JOSIP JURAJ STROSSMAYER UNIVERSITY OF OSIJEK

FACULTY OF ELECTRICAL ENGINEERING, COMPUTER SCIENCE AND INFORMATION TECHNOLOGY OSIJEK

Proposal for Amendments

to the Graduate University Study Programme in Computer Engineering

CLASS: 602-01/23-12/00001
REG.NO: 2158-80-05-23-00001
Osijek, 12 December 2023

Osijek, 2023

Table of Contents

1. BASIC INFORMATION ABOUT THE STUDY PROGRAMME

The graduate university study programme in Computer Engineering at the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek (formerly known as the Faculty of Electrical Engineering in Osijek) has been carried out since the 2008/2009 academic year. Taking into account the interests and the needs of the job market, the broader social community, student interests, as well as scientific promotion of staff who could participate in the teaching process, and considering the results of the Croatian Qualifications Framework (CQF) project "Dig IT - Development of occupational standards and qualification standards", we have decided to propose amendments to the study programme.

Pursuant to the amendments to the graduate university study programme in Computer Engineering, students will be able to enrol on one of the four modules, and a separate admission quota will be defined for each module:

- DRA – Computer Engineering
- DRB – Artificial Intelligence and Robotics
- DRC – Software Engineering
- DRD – Data Science

1.1. Name of the authorised body of the University constituent unit that approved the amended study programme

The Faculty Council of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek, Josip Juraj Strossmayer University of Osijek, adopted the document entitled "Proposal for Amendments to the Graduate University Study Programme in Computer Engineering" at its 308th session held on 12 December 2023 (the Faculty Council decision is attached in Appendix 7.1).

1.2. List of individuals involved in the document preparation process

At its 272nd session held on 8 March 2022, the Faculty Council of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek appointed a working group on the procedure and implementation of major changes to the graduate university study programme in Computer Engineering. The working group members who also participated in the preparation of this document are as follows:

- Tomislav Matić, PhD, Full Professor – Technical Sciences/Electrical Engineering, Dean
- Danijel Topić, PhD, Associate Professor - Technical Sciences/Electrical Engineering, Vice-Dean for Education and Student Affairs
- Goran Martinović, PhD, Full Professor with Tenure - Technical Sciences/Computer Engineering
- Dražen Slišković, PhD, Full Professor - Technical Sciences/Fundamental Technical Sciences
- Tomislav Rudec, PhD, Assistant Professor – Natural Sciences/Mathematics, Head of the Department of Core Courses
- Irena Galić, PhD, Full Professor - Technical Sciences/Computer Engineering, Head of the Department of Software Engineering
- Ivan Vidović, PhD, Assistant Professor - Technical Sciences/Computer Engineering
- Zdravko Krpić, PhD, Associate Professor - Technical Sciences/Computer Engineering
- Josip Job, PhD, Full Professor - Technical Sciences/Computer Engineering
- Damir Blažević, PhD, Full Professor - Technical Sciences/Computer Engineering
- Damir Filko, PhD, Associate Professor - Technical Sciences/Computer Engineering

In addition to the working group members,

- Tomislav Keser, PhD, Associate Professor - Technical Sciences/Computer Engineering, Head of the Department of Computer Engineering and Automation,

also participated in the preparation of this document.

1.3. Brief description of the study programme coordinator and basic information (*history, organisational structure or schematic representation, mission and vision, name and address, email address, website address*)

History

Higher education in the field of electrical engineering was initiated in 1978 at the Electromechanical Engineering Study Programme in Osijek, which was the third study programme in electrical engineering in the Republic of Croatia. In 1981, it evolved into an independent higher education institution. In 1988, the Electromechanical Engineering Study Programme changed its name to the Electrical Engineering Study Programme and, within the framework of the new curriculum, it expanded the existing electromechanical engineering study programme to encompass the education of electrical and electronics engineers.

In 1989, a joint Electrical Engineering Study Programme curriculum was adopted in the Republic of Croatia, with branches in Electrical Engineering and Electronics, which was applied in Osijek by the 2002/2003 academic year. At the same time, due to the high demand for professionals in the field of electrical engineering in this part of Croatia, with the support of the University and the regional economy, preparations were initiated to transform the Electrical Engineering Study Programme into a faculty offering a four-year degree programme.

In the 1990/1991 academic year, the Electrical Engineering Study Programme in Osijek evolved into the Faculty of Electrical Engineering Osijek. As of the 2005/2006 academic year, the Faculty has conducted undergraduate study programmes, and as of the 2008/2009 academic year, graduate study programmes as well, which were aligned with the Bologna Declaration. Postgraduate specialist study programmes, harmonised with the Bologna Declaration, have been conducted since the 2006/2007 academic year. Postgraduate doctoral study programmes have been conducted at the Faculty of Electrical Engineering, Computer Science and Information Technology since the 2000/2001 academic year. The postgraduate university (doctoral) study programme in Electrical Engineering, with branches in Power Engineering and Communications and Informatics, aligned with the Bologna Declaration, has been conducted since the 2006/2007 academic year. In the 2018/2019 academic year, the study programme changed its name to the postgraduate university study programme in Electrical Engineering and Computer Science, with the following three modules: Power Engineering, Communications and Informatics, and Computer Science.

In order to present educational, scientific and professional activities of the Faculty more clearly and distinctly, and in accordance with the qualifications awarded to students upon successful completion of the study programmes, the Faculty of Electrical Engineering Osijek changed its name in 2016 to the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek (hereinafter referred to as: FERIT).

Organisational structure

FERIT is organised into six departments with corresponding chairs and laboratories, along with additional laboratories designed for teaching and research, as well as a scientific and research centre:

- Department of Core Courses:
  - Chair of Mathematics, Physics and Mechanical Engineering
  - Chair of Humanities and Social Sciences
- Department of Software Engineering:
  - Chair of Programming Languages and Systems
  - Chair of Visual Computing
- Department of Computer Engineering and Automation:
  - Chair of Computer Engineering
  - Chair of Automation and Robotics
- Department of Electromechanical Engineering:
  - Chair of Electric Machines and Power Electronics
  - Chair of Fundamentals of Electrical Engineering and Measurement
- Department of Power Engineering:
  - Chair of Power Systems and Substations
  - Chair of Power Plants and Energy Processes
  - Electromagnetic Compatibility Laboratory (an internationally accredited laboratory)
- Department of Electromechanical Engineering:
  - Chair of Electronics and Microelectronics
  - Chair of Radiocommunications and Telecommunications
  - Chair of Multimedia Systems and Digital Television
  - Laboratory for High Frequency Measurements (an internationally accredited laboratory)
- Scientific Research Centre for Electrical Engineering and Computer Science

Mission and vision

The mission of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek is to:

- educate professionals who will create added value and contribute to the development of the Republic of Croatia with knowledge acquired and competencies adopted in the fields of electrical engineering, computer science and IT,
- deliver education based on knowledge gained through the implementation of competitive research projects and projects in collaboration with companies, and
- foster economic development through innovation and technology transfer, contributing in this way to the advancement of society.

The vision of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek is to be an institution with developed educational and research capacities, ensuring competitiveness with European and global higher education institutions, scientific excellence, and international recognition in the fields of electrical engineering, computer science and information technology, as well as effective transfer of knowledge and new technologies to the economy.

Name of the higher education institution:

Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

Address:
Kneza Trpimira 2b
31000 Osijek

Cara Hadrijana 10b
31000 Osijek

Phone:
+385 31 224 600

Email address:
ferit@ferit.hr

Web address:
http://www.ferit.unios.hr

1.4. Name of the study programme

Graduate university study programme in Computer Engineering

1.5. Type of the study programme (*university or professional*)

University study programme

1.6. Study programme level according to the CROQF (*qualification and full name*)

7.1.sv - Graduate university study programme

1.7. Duration of the study programme

2 years

1.8. Number of ECTS credits

120 TS credits

1.9. Provider of the study programme

Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

1.10. Study programme delivery location

Osijek, Republic of Croatia (at the seat of the study programme provider)

1.11. Academic year in which the amended study programme is planned to be implemented

The amended graduate university study programme in Computer Engineering is planned to be implemented in the 2024/2025 academic year.

1.12. Academic/professional title awarded upon completion of the study programme

Upon completion of the Graduate university study programme in Computer Engineering, a student is awarded the academic title of **Master of Computer Engineering**, with the module (**Computer Engineering, Artificial Intelligence and Robotics, Software Engineering,** or **Data Science**) specified.

1.13. Access to a regulated profession by acquiring the qualification

Acquiring the qualification does not grant access to a regulated profession.

1.14. Optimal number of students specified in the study programme implementation document

The total anticipated number of students enrolling in this study programme has slightly increased compared to previous years, i.e., 112 students are planned to enrol in the first year of study.
Given that these amendments to the study programme introduce four modules, an admission quota is planned for each module as follows:

Module: Computer Engineering - 16 admissions
Module: Artificial Intelligence and Robotics - 16 admissions
Module: Software Engineering - 48 admissions
Module: Data Science - 32 admissions

2. INSTITUTIONAL PREREQUISITES

2.1 To initiate the graduate university study programme, provide documentation detailing an accredited undergraduate study programme in the same scientific or artistic field.

The document detailing the accredited undergraduate university study programme in Computer Engineering is given in Appendix 7.9. Licence for carrying out the graduate university study programme in Computer Engineering to which these amendments are proposed is also attached in the appendix.

2.2. Information on the occupational title from the national standard classification of occupations equivalent to the academic title obtained.

2.25.251.2519 Software Development Engineers and Software Support Analysts, n.e.c.

2.3. Provide the rationale for the study programme *(social needs, economic needs, scientific and cultural needs, lack of academic profiles in the job market, etc.).*

Over the past decade, the IT sector has been rapidly developing in the area of the city of Osijek and Osijek-Baranja County (as well as throughout the Republic of Croatia), requiring ICT experts. The graduate university study programme in Computer Engineering has been conducted since the 2008/2009 academic year, educating professionals in the field of computer science to meet the demands of the job market. As the IT sector (to which computer science belongs) evolves and changes rapidly, it is necessary to regularly align study programmes with the needs of the job market. These study programme changes are primarily aimed at synchronising the study programme with the needs of the job market.

The rationale for implementing this study programme is also confirmed by the *Recommendations of the Croatian Employment Service for enrolment policy and scholarship policies for 2023,*[1] which identify university study programmes in computer engineering as educational programmes in which the admission quota and the number of students who receive a scholarship should be increased. Furthermore, in 2022, FERIT carried out a study into the needs of the job market in the fields of electrical engineering, computer science, and information technology for the period 2022-2026 in the Osijek area, which is entitled *FERIT and OSIJEK 2026.*[2] This study highlights a significant demand expressed by the real sector for professionals in the field of computer science and the need to raise the admission quota, which in turn justifies the implementation of the graduate university study programme in Computer Engineering and the proposed amendments.

2.4. Assessment of relevance in relation to labour market needs in the public and private sectors.

Taking into account the interests and needs of the job market, the broader social community, student interests, as well as scientific promotion of the staff who could participate in the teaching process, and considering the results of the Croatian Qualifications Framework (CQF) project "Dig IT - Development of occupational standards and qualification standards", we have decided to propose amendments to the study programme. Additionally, the undergraduate university study programme in Computer Engineering was modified in the 2020/2021 academic year based on the expressed needs of employers in the public

[1] *Recommendations of the Croatian Employment Service for enrolment policy and scholarship policies for 2023* are available at: https://www.hzz.hr/app/uploads/2023/07/preporuke_22kor-1.pdf

[2] *FERIT and OSIJEK 2026* study is available at: https://www.ferit.unios.hr/fakultet/dokumenti-za-fakultet#dokument-ferit-osijek-2026pdf

and private sectors. These amendments are being proposed to ensure that upon completion of the undergraduate and graduate study programmes, students acquire the competencies and skills demanded by the job market.

Furthermore, occupational standards for a Master's degree in Software Engineering and a Master's degree in Computer Engineering[3] were developed within the Croatian Qualifications Framework (CQF) project entitled "Dig IT - Development of occupational standards and qualification standards", based on surveys of employers throughout Croatia. Proposals for qualification standards (currently under evaluation) were developed within the project in line with these occupational standards, and the current amendments are aligned with the proposed qualification standards, which justifies the amendments to the graduate university study programme in Computer Engineering with respect to the needs of the labour market in the public and private sectors

2.5. Alignment of the proposed study programme with the mission and strategy of the University and strategic documents of the network of higher education institutions

The Strategy of Josip Juraj Strossmayer University of Osijek for the period 2021-2030 was adopted at the 2nd session of the Senate of Josip Juraj Strossmayer University of Osijek in the 2021/2022 academic year that was held on 24 November 2021. The Development Strategy of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek for the period 2021-2025 was adopted at the 272nd regular session of the Faculty Council held on 8 March 2022, and it is aligned with the Strategy of Josip Juraj Strossmayer University of Osijek for the period 2021-2030.

The Development Strategy, *inter alia*, consolidates individual action plans for the teaching process, scientific research, professional development activities, a quality assurance system, and resource development, as well as detailed road maps to all activities. It is continuously monitored and analysed whether the tasks aimed at achieving the set strategic goals are fulfilled, and the Faculty Council ensures the implementation of the Strategy, among other things, by adopting the plan and report submitted by the Commission for Quality Enhancement and Assurance in Higher Education, and the annual Report of the Dean on the work and operation of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek.

Alignment of the study programme, especially the proposed amendments to the study programme, with the Strategy and the mission of Josip Juraj Strossmayer University of Osijek can be recognised through several strategic goals outlined in the Strategy.[4] Alignment of the study programme with the University Strategy is described through compliance with relevant strategic objectives in the field of study programmes, the teaching process and student support as follows:

STRATEGIC GOAL 1.1. Align general goals of all study programmes with the mission and strategic goals of the University and with social and economic needs.

- One of the tasks to achieve this goal is “to monitor the labour market situation and adjust admission quotas in accordance with the recommendations of professional associations and the Croatian Employment Service (CES)”. These amendments to the study programme have resulted from monitoring the labour market situation, aligning study programmes with market needs, and following the CES recommendations.

[3] The given occupational standards are available in the CROQF register.

[4] The Strategy of Josip Juraj Strossmayer University of Osijek for the period 2021-2030 is available at: http://www.unios.hr/wp-content/uploads/2022/02/strategija-sveucilista_20211124.pdf

STRATEGIC GOAL 1.2. Align study programme learning outcomes with the level and profile of qualifications based on them.

- One of the tasks to achieve this goal is that “learning outcomes clearly reflect competencies needed for employment, further education, or other individual and societal needs". The proposed amendments to the study programme are primarily based on the Croatian Qualifications Framework (CROQF) project described in the previous chapter, i.e. learning outcomes are based on competencies and key tasks identified by employers through surveys as necessary for the job market.

STRATEGIC GOAL 1.6. Ensure the quality of student practical training in all study programmes.

- According to these amendments to the study programme, students are planned to have only practical training and a Master's thesis in the 4th semester. This will enable students to collaborate with companies in developing their Master's theses, facilitate mobility, and enhance the quality of practical training.

STRATEGIC GOAL 1.7. Increase flexibility within study programmes to enhance internal mobility.

- One of the tasks to achieve this goal is to “increase the number of elective courses/modules". The proposed amendments to the study programme enable students to choose one of the following four different modules: Computer Engineering, Artificial Intelligence and Robotics, Software Engineering, and Data Science. Within each module, students also have the option to choose at least 1 elective course in the 3rd semester.

STRATEGIC GOAL 2.1. Clear requirements for enrolment or continuation of studies.

- Enrolment requirements are defined in Chapter 3.6, while continuation of studies is described in Chapter 4.8.

STRATEGIC GOAL 2.2. Student-centred teaching and learning and continuous work aimed at improving student achievement.

- The proposed amendments to the study programme contribute to student-centred teaching and learning and continuous monitoring of student work and achievement. The Commission for Quality Enhancement and Assurance regularly conducts student surveys. Furthermore, these changes contribute to the use of advanced technologies with the aim of modernising the teaching process.

STRATEGIC GOAL 2.3. Student support.

- One of the tasks is “to provide study and career counselling for students". Within the curriculum, students are regularly provided with guest lectures aimed at career counselling. Furthermore, FERIT organises an Open Doors and Career Day (DOVIK) every year, and participates in the University Career Week. These two events enable students to come into direct contact with employers and provide information about potential development of their careers.

STRATEGIC GOAL 2.4. Provide students with opportunities to gain international experience.

- Workshops on the possibilities of international student mobility are organised for students every year. A procedure for the recognition of ECTS credits obtained at other institutions has been established. Pursuant to the amendments proposed for the 4th semester, students are planned

to have only practical training and a Master's thesis, which in turn makes it easier for them to take part in international mobility programmes and gain international experience.

STRATEGIC GOAL 2.6. Increase the number of students in STEM fields.

- The proposed study programme, i.e. amendments to the study programme, directly contribute to an increasing number of students in STEM fields.

2.6. Alignment of the proposed study programme with the strategic goals of the constituent unit.

Alignment of the proposed study programme, i.e. its amendments, with the Development Strategy of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek 2021-2025[5] is reflected in the contribution to the following strategic goals in the field of teaching activities.

STRATEGIC GOAL 1. To ensure high-quality education for students and lifelong education and training in the fields of electrical engineering, computer science and information and communication technology through the integration of teaching, scientific research, and collaboration with the industry.

This study programme, i.e. its amendments, contribute to addressing the following issues set within the aforementioned goal:

- Revision and modernisation of study programmes in line with the requirements of the Croatian Qualifications Framework, taking into account employer recommendations.
- Increasing the visibility of FERIT as a higher education institution that offers high-quality education in the fields of electrical engineering, computer science and information and communication technology.
- Involving students in scientific research and programmes with industry collaboration through student projects and practical training, as well as final papers and Master's theses.
- Informatisation and digitalisation of processes and procedures related to teaching, including the promotion of further development of e-learning.
- Encouraging academic excellence.
- Strengthening the publishing activities of the Faculty.
- Increasing incoming and outgoing mobility of students and teachers.

As previously mentioned, the proposed amendments to the study programme are largely based on the Croatian Qualifications Framework (CROQF) project results, which also include recommendations from employers referring to alignment with the labour market needs. This amended and updated study programme undoubtedly enhances visibility and recognition of FERIT as a higher education institution. Practical training is conducted in private and public sector companies, and every year, Master's thesis topics are assigned in collaboration with industry partners, which encourages academic-industry collaboration and scientific research. Furthermore, every year, during the celebration of the Day of the Faculty, the academic excellence awards are conferred on the highest-achieving students, including students enrolled on this study programme, and the awards are provided by industry partners. These amendments to the study programme also introduce new courses that follow the latest technology trends in the IT sector, which would result in the development of new educational materials and strengthening the publishing activities of the Faculty. The shift of practical training and Master's theses to the 4th (final) semester facilitates student mobility.

[5] Development Strategy of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek 2021-2025 is available at: https://www.ferit.unios.hr/fakultet/dokumenti-za-fakultet#dokument-strategija-razvoja-ferit-a-2021---2025pdf

2.7. Alignment with the Recommendations of the Croatian Employment Service for education policy.

Based on the Regulation of the Government of the Republic of Croatia on the monitoring, analysis and forecasting of the labour market needs for particular qualifications and on the preparation and taking into account the recommendations for education policy (Official Gazette 93/10), regional offices and Croatian Employment Service offices, under the coordination of the Central Office, carried out an analysis, forecast market needs for specific occupations and developed recommendations for education enrolment policy.

Pursuant to the Recommendations for Enrolment Policy and Scholarship Policies issued by the Croatian Employment Service (CES) for 2023[6], university study programmes recommended to be studied and financially supported, in almost all counties, are the ones in the field of computer engineering.

2.8. Compliance of the study programme with the study programmes accredited in Croatia and other European Union countries (the study programme is based on the latest research and related skills and is harmonised with professional standards and contemporary achievements in the field).

The proposed graduate university study programme in Computer Engineering is greatly based on the current graduate university study programme thus preserving the initial comparison with the quality of related accredited study programmes in the Republic of Croatia and the European Union countries. One of the main reasons for initiating the proposed changes to the graduate university study programme in Computer Engineering is the implementation of the project "Dig IT - Development of occupational standards and qualification standards in computer science" engaging all higher education institutions that carry out university study programmes in the field of Computer Engineering in the Republic of Croatia. Within the project, four proposals for occupational and qualification standards were prepared; one for the undergraduate university level and three for the graduate university level. All partner institutions participating in the project have undertaken to align their graduate university study programmes in the field of Computer Engineering with the proposed qualification standards "Master's degree in Computer Engineering, Computing", "Master's degree in Computer Engineering, Software Engineering" and Master's degree in Computer Engineering, Data Science". The aforementioned harmonisation of study programmes at all partner institutions (Faculty of Electrical Engineering, Computer Science and Information Technology Osijek, Faculty of Electrical Engineering and Computing Zagreb, Faculty of Engineering Rijeka, Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture Split and the University of Dubrovnik) ensured that all study programmes include all compulsory courses learning outcomes of the proposed qualification standards, which ensures a complete comparability of the graduate university study programme in Computer Engineering with related accredited studies in Croatia.

The study programme is comparable in terms of the content and qualifications to the graduate study programmes of Croatian universities:

- graduate university study programme in Computing at the Faculty of Electrical Engineering and Computing, University of Zagreb (https://www.fer.unizg.hr/studiji/dipl/rac). To illustrate, the learning

[6] Recommendations of the Croatian Employment Service for enrolment policy and scholarship policies for 2023 are available at: https://www.hzz.hr/app/uploads/2023/07/preporuke_22kor-1.pdf

outcomes of the proposed study programme are greatly comparable to the learning outcomes of the following courses:

- Bioinformatics 1
- Deep Learning 1
- Internet of Things
- Software Testing
- Advanced Databases
- Advanced Algorithms and Data Structures
- Advanced Development of Software Support for the Web
- Neural Networks
- Numerical Methods in Computer Science
- Design Patterns in Programming
- Natural Language Processing
- Basics of Robotics
- Parallel Programming
- Embedded Systems Design
- Computer Vision
- Cloud Computing
- Distributed Ledgers and Cryptocurrencies
- Distributed Systems
- Pattern Recognition
- Machine Learning 1
- Real-Time Computer Systems
- Ubiquitous Computing
- Embedded Computer Systems
- Project Management
- Introduction to Data Science
- Data Visualisation

• graduate university study programmes at the Faculty of Organisation and Informatics, University of Zagreb (https://www.fer.unizg.hr/studiji/dipl/ikt). To illustrate, the learning outcomes of the proposed study programme are greatly comparable to the learning outcomes of the following courses:

- Automation of Plants and Processes
- Autonomous Mobile Robots
- Deep Learning 1
- Basics of Robotics
- Robot Programming and Simulation
- Computer Systems Management
- Sensors, Perception and Actuation in Robotics
- Three-Dimensional Computer Vision
- Embedded Computer Systems

• graduate university study programme in Computing at the Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture, University of Split (https://www.fesb.unist.hr/studiji/diplomski-racunarstvo/). To illustrate, the learning outcomes of the proposed study programme are greatly comparable to the learning outcomes of the following courses:

- Digital Image Processing and Analysis
- Optimisation Methods
- Advanced Web Technologies
- Advanced Algorithms

- Neural Networks and Genetic Algorithms
- Parallel Programming
- Computer Games Development
- Designing and Using Computer Networks
- Data Warehouses
- Embedded Computer Systems
- Artificial Intelligence
- Project Management

• graduate university study programme in Electronics and Computer Engineering at the Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture, University of Split (https://www.fesb.unist.hr/studiji/diplomski-studij/elektronika-i-racunalno-inzenjerstvo/). To illustrate, the learning outcomes of the proposed study programme are greatly comparable to the learning outcomes of the following courses:

- Algorithms and Data Structures
- Programming of Mobile Robots and Aircraft
- Computer Network Design and Use
- Embedded Computer Systems
- Artificial Intelligence

• graduate university study programme in Computer Engineering at the Faculty of Engineering Rijeka (http://www.riteh.uniri.hr/obrazovanje/sveucilisni-diplomski-studij/racunarstvo/). To illustrate, the learning outcomes of the proposed study programme are greatly comparable to the learning outcomes of the following courses:

- Digital Image Processing
- Mobile Robotics
- Advanced Computer Networks
- Advanced Algorithms and Data Structures
- Embedded Systems Programming
- Project Management
- Mobile Applications Development
- Machine Learning

Additionally, the study programme is comparable to study programmes carried out at European universities (see Chapter 3.21 for a detailed comparison):

• Technical University of Kaiserslautern, study programme in Computer Science: https://rptu.de/studienangebot/22777/Informatik-Master-Computer_Science
• Technical University of Wien, study programme in Data Science: https://tiss.tuwien.ac.at/curriculum/public/curriculum.xhtml?dswid=3430&dsrid=934&key=67853
• Technical University of Wien, study programme in Computer Engineering: https://tiss.tuwien.ac.at/curriculum/public/curriculum.xhtml?dswid=6205&dsrid=342&key=56543
• Technical University of Bremen, study programme in Informatik: https://www.uni-bremen.de/en/studies/orientation-application/study-programs/dbs/study/26?cHash=fd4ee5e1b64a3e2c90d2bc185e2a89c6
• Technical University of Munich, study programme in Computational Science and Engineering https://www.tum.de/en/studies/degree-programs/detail/computational-science-and-engineering-cse-master-of-science-msc

- Technical University of Munich, study programme in Robotics, Cognition, Intelligence: https://www.tum.de/studium/studienangebot/detail/robotics-cognition-intelligence-master-of-science-msc

The study programmes are generally comparable because they last for two years, students acquire the same number of ECTS credits (120) and the academic title of the Master of Computer Engineering is fully comparable in the Republic of Croatia and other European Union countries. The evidence of comparability is successful incoming and outgoing student mobility within Erasmus mobility programmes (p. 3.24). Mobility will be continued because the basic compliance assumptions with the Bologna process will not be altered.

The quality of the teaching process is ensured by the teachers and associates affiliated to the following departments. They will teach the majority of the courses.

- **Department of Software Engineering**, which consists of the Chair of Programming Languages and Systems and the Chair of Visual Computing;
- **Department of Computer Engineering and Automation**, which consists of the Chair of Computer Engineering and Chair of Automation and Robotics;
- **Department of Core Courses**, which consists of the Chair of Mathematics, Physics and Mechanical Engineering and the Chair of Humanities and Social Sciences;
- **Department of Communications**, which consists of the Chair of Radiocommunications and Telecommunications, Chair of Electronics and Microelectronics, Laboratory of Multimedia Systems and Digital Television and the Laboratory for High Frequency Measurements.

It should be emphasised that the Committee for the Quality Enhancement and Assurance in Higher Education at the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek continuously evaluates the students, studies their surveys on the teaching process and teachers as well as carries out other activities on enhancing the quality of studies.

The comparison of the proposed graduate university study programme in Computer Engineering leads to a conclusion on a high comparability level of this study programme with related study programmes, thus potentially resulting in better student mobility between the University of Osijek and other Croatian and the majority of European universities.

2.9. Openness of the study programme to student mobility (national and international student mobility)

Figure 1 shows the vertical scheme of the study programmes carried out at the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek. Holders of a university Master degree in Computing Engineering, who complete the graduate university study programme in Computer Engineering, are eligible to enrol in university postgraduate specialist study programmes or doctoral study programmes at the Faculty (see Figure 1), but also at related faculties of other Croatian and foreign universities meeting to the requirements given by the respective institutions.

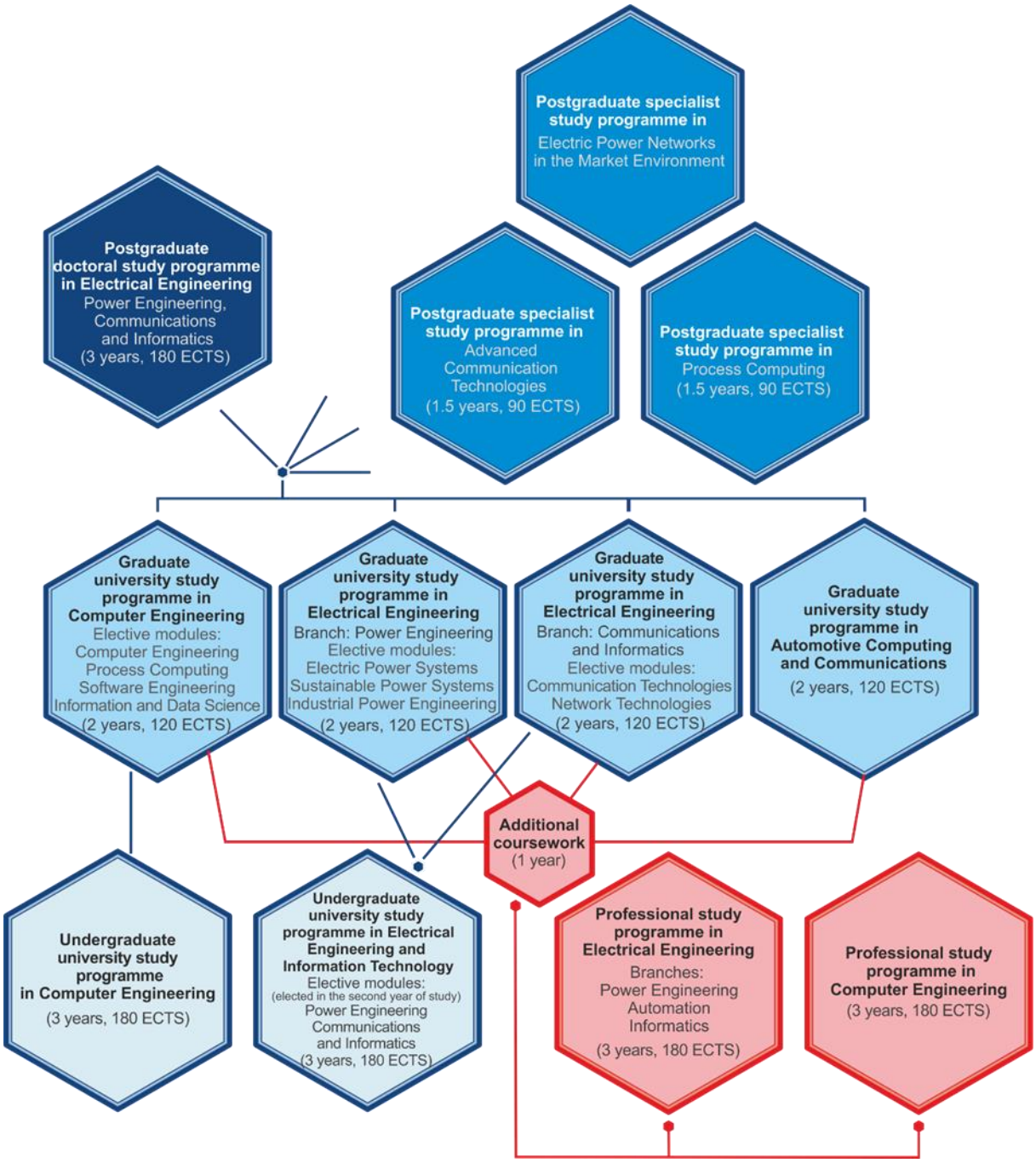

Figure 1. Vertical studying scheme at the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

The implementation of the project "Dig IT - Development of occupational standards and qualification standards in computer science" ensured the harmonisation of study programmes in the field of computer engineering with the proposed qualification standard, thereby enabling easier vertical student mobility at the national level.

These amendments to the study programme contribute to the openness of student mobility. In the current version of the study programme, in the last (4th) semester, in addition to the Master's thesis, students

had three compulsory and one elective course, while practical work was done in the 3rd semester. With these changes, in the last (4th) semester, students will only have to do their practical work and Master's thesis, which will enable them to engage in mobility programmes more easily. Also, it will be much easier for them to do a student internship and Master's thesis at a foreign institution because they will not have to take care of courses but fully devote themselves to internships and Master's thesis.

The outgoing and incoming student international mobility can be carried out during their studies through the Erasmus+ mobility programme between the Faculty and about 70 foreign higher education institutions. Also, mobility takes place through IAESTE and other mobility programmes. In the last 5 years, there were 52 and 62 outgoing and incoming student mobilities, and 83 and 81 teaching and non-teaching staff mobilities, respectively.

2.10. The University standards and regulations for the verification of learning outcomes within the study programme

The Ordinance on Studies and Studying at Josip Juraj Strossmayer University of Osijek regulates the verification of learning outcomes, i.e. examination procedures (written, oral, practical part of the exam, prerequisites, deadlines, number of exams, etc.), appeals to a grade, procedure for retaking exams, content, form and method of keeping exam-related documents, providing the public at exams, the right to study one's exam and other issues. The Ordinance on Studies and Studying at Josip Juraj Strossmayer University of Osijek is published on the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek website and is therefore available to the public, especially students and aspiring students.

Based on the Criteria for evaluating and assessing students studying according to the Bologna process from 20 November2007, the Evaluation criteria were created and published on the Faculty's website. The latest version was adopted at the 244th session of the Faculty Council held on 2 June 2020. According to that version, each teacher is required to provide evaluation criteria for each course according to the method prescribed in the document available at https://www.ferit.unios.hr/2021/upisi-i-studiji/dokumenti-za-upise-i-studije#dokument-okviri-kriterija-ocenjenja-2020pdf.

The examination criteria for each course are clearly stated on the individual course page on Merlin e-learning platform (https://merlin.srce.hr/) and the Faculty's website https://www.ferit.unios.hr/2021/upisi-i-studiji/diploma-university-study.

In addition to the learning outcomes, the thresholds for successful passing and the percentage in the final grade are defined for each activity. As stated in 2.4, one of the main reasons for the changes in the study programme is the implementation of the project "Dig IT - Development of occupational standards and qualification standards in computer science". Within the project, workshops on learning outcomes for teachers were held. Also, compulsory and optional sets of learning outcomes were defined for individual qualification standard proposals. For each set of learning outcomes in the proposed qualification standards, the procedures and evaluation examples of each set of learning outcomes are specified. The mentioned activities contribute to the verification of the learning outcomes acquisition.

An important part of the quality assurance is the University student survey, which is conducted according to J. J. Strossmayer University of Osijek guidelines. The survey is completed by all full-time students. The survey is usually conducted at the end of the academic year. Students evaluate classes, criteria for assessing students' knowledge and work, teachers' availability and attitude towards students. The survey results are analysed by the administration and, if necessary, the necessary measures are taken.

Cumulative results are presented at the Faculty Council. The Committee for the Quality Enhancement and Assurance in Higher Education prepares and sends individual results to each teacher, and the Committee's Head, together with the dean, in case of frequent or important student complaints, talks to specific teachers whose behaviour needs to change. The survey results are used for the evaluation of teaching and professional activities in the selection process for scientific and teaching positions required by the Rector's Board.

Additionally, the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek conducts a Faculty survey on learning outcomes and ECTS credits, thus examining the students' workload, i.e. the number of working hours spent mastering individual activities in the course and passing the exam. The survey also includes the questions on if the teachers taught contents provided for in the study programme and how much the individual forms of teaching contributed to the successful acquisition of the learning outcomes. The information is compared with the students' grades in all courses. Furthermore, teachers' attendance and course requirements (with a special emphasis on revision exams and exams) presented at the beginning of the semester are also surveyed.

Additional Faculty surveys are as follows:

- a survey on the postgraduate study programme in which the students evaluate their supervisor, Vice Dean for Science and Postgraduate Studies, Student Administration Office, the quality of information about the study programme as well as the procedures they encounter in it;
- a survey for alumni who evaluate their former teachers, course delivery and how much the Faculty has helped them get the desired employment and the status within the company;
- a survey for employers who evaluate the students employed. They also suggest which direction the Faculty should develop in, i.e. the study programmes carried out by the Faculty.

If necessary, other surveys are conducted. The following surveys were conducted - a survey on the graduate university study programme in Automotive Computing and Communications with students evaluating the importance and applicability of the acquired competencies on the labour market. A survey on Open Doors and Career Day (DOVIK), a survey on teaching materials available on the course's pages and a survey on student mobility were conducted. Based on the survey results, measures to improve the quality of higher education are taken.

Furthermore, the Faculty Council of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek, at its 234th regular session, held on November 5, 2019, adopted the Handbook for Quality Enhancement and Assurance in Higher Education, which describes the basic procedures and forms required to systematically monitor activities' compliance with the document "Standards and Guidelines for Quality Assurance in the European Higher Education Area". The Handbook describes, among other things, the system for quality enhancement and assurance, activities to be done, quality indicators that are an integral part of the internal evaluation adopted by the Faculty Council at the end of the academic year, among which are:

- Number of candidates applying for the study programme/Enrolment quota;
- Number of freshmen/Number of graduates;
- Total number of students/Number of students retaking a study year;
- Total number of university students/Number of teachers elected in scientific-teaching positions;
- Total number of professional studies students/Number of teachers elected in teaching positions;
- Number of students enrolled in a higher year/Number of freshmen;
- Number of students enrolled in the postgraduate study programme;
- Number of foreign students enrolled in the first year of the postgraduate study programme/Total number of students enrolled in the first year of the postgraduate study programme;
- Number of defended doctoral theses;

- Number of research papers published in journals indexed in the Web of Science database/Number of teachers elected in scientific-teaching positions;
- Citation of papers indexed in the Web of Science database;
- Total impact factor of papers/Number of research papers published in journals indexed in the Web of Science database;
- Number of research papers in the five-year period published in journals that are, according to the IF level, among 25% of journals with the highest IF within the corresponding category (Q1)/Number of research papers published in journals indexed in the Web of Science database in the five-year period;
- Number of competitive research projects approved for financing/Number of competitive research project applications (HRZZ, UKF, FP7, Horizon2020);
- Contracted funds for competitive research projects;
- Number of other research projects approved for financing/Number of applications for other research projects (IPA; PoC, contracts with partners from the industry, etc.);
- Contracted funds for other research projects;
- Number of researchers who spent at least two weeks at foreign institutions/Total number of teachers elected in scientific positions, assistants, senior assistants and research assistants;
- Outgoing mobility of teachers/Number of teachers elected in teaching and scientific-teaching positions;
- Incoming mobility of teachers/Number of teachers elected in teaching and scientific-teaching positions.

A new Regulations on the Organisation of the Quality Assurance System in Higher Education at the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek was unanimously adopted at the 290th session of the Faculty Council held on 17 January 2023. A new Handbook for Quality Enhancement and Assurance in Higher Education was unanimously adopted at the 294th session of the Faculty Council held on 14 March 2023.

2.11. Ensuring student participation in all processes related to assuring the quality of a higher education institution (e.g. *Faculty committees and University boards*)

The Faculty Council of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek consists of student representatives suggested by the Student Union. Student representatives must make up at least 10% of the members of the Faculty Council. Pursuant to Article 33 of the Statute of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek, student representatives can exercise the right to a suspensive veto on issues, which are of special interest to students, such as changing the study system, quality assurance, proposing study programmes, teaching plans and their execution and student standards. Student representatives can use a suspensive veto when requested by a majority of all student representatives in the Faculty Council. After the suspensive veto, the Faculty Council discusses the above issue again within 8 days at the earliest. In repeated decision-making, the decision is made by a majority of votes of the total number of the Faculty Council members who have the right to vote, without the right to use a suspensive veto.

A student representative is a member of the Ethics Committee, the Committee for Awarding Students, the Committee for Quality Enhancement and Assurance and the Disciplinary Court. The student representative is also appointed to the Working Group for Learning Outcomes and participates in the work of the aforementioned bodies, especially when it comes to revising or creating new quality-related documents.

In addition to their representatives, students can directly participate in the processes related to the quality assurance at the higher education institution, primarily by completing the University survey and the survey on learning outcomes and ECTS credits.

Based on survey results, if negative, the Faculty management introduces a set of measures to enhance the quality in depicted cases. The University survey results are used for the evaluation of teaching and professional activities in the selection process to scientific and teaching positions required by the Rector's Board.

2.12. Ensuring adequate support for future students (*counselling and career opportunities*)

For several years, FERIT has been organising LABUS[7] (Laboratory for High School and Primary School Students) workshops aimed at high school and primary school students. LABUS is a portal developed by FERIT for students, high school and primary school teachers, and other participants in the high school and primary school educational process in the STEM fields, particularly in mathematics, physics, electrical engineering, computer science, and information technology. LABUS functions as a "bridge," connecting primary and high school education in natural sciences, mathematics, and technical subjects with higher education engineering courses. Through LABUS, students are expected to learn how to integrate fundamental physics and mathematical knowledge and skills while acquiring engineering knowledge and skills, thereby providing support for future students in developing their careers.

Furthermore, every year FERIT visits high schools in the Slavonia and Baranja region to present study programmes and showcase career development opportunities in the fields of computer science, electrical engineering, and information technology.

Before the start of each new academic year, FERIT regularly organises "Pre-sessional courses for freshmen". Over a two-week period, FERIT teachers conduct free preparatory lectures and exercises in mathematics, physics, programming, and electrical engineering for freshmen, offering additional support for the development of their careers.

Throughout the academic year, within various courses, FERIT organises various guest lectures delivered by industry companies to acquaint students with various career development opportunities. Additionally, Open Doors and Career Day (DOVIK) is organised each year, where various companies come to present job opportunities, internships, and scholarships, directly supporting the career development of students.

2.13. Connection with the local community

The connection of the study programme with the needs of the local community is partially described under 2.14, which outlines the involvement of representatives from the labour market in the development of the University. Furthermore, the connection with the local community is reflected in the implementation of the FERIT and OSIJEK 2026 study (described in Chapter 2.3), in which local employers participated in market research and expressed the need for professionals whom the Faculty will educate.
Another piece of evidence of collaboration with the local community is the city-business scholarship programme "Best for the Best"[8], initiated by the city of Osijek in collaboration with various companies. Through this programme, students from five constituent units of J. J. Juraj Strossmayer University of Osijek are awarded scholarships in the areas where there is a significant need for a workforce. FERIT is one of these five units.
Furthermore, experts from FERIT regularly participate in local events in the fields of computer science, electrical engineering, IT, research, and education, such as conferences, forums, lectures, and the like, fulfilling their responsibility to the local community.

[7] More information about Labus can be found at: https://labus.ferit.hr/
[8] More information about the scholarship can be found at: https://najzanaj.osijek.hr/

2.14. Ensuring the involvement of labour market representatives in the development of the University (external stakeholders participating in the work of specific bodies at the University constituent units and the University level)

The management of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek has expressed its interest in strengthening the connection between the profession and the activities of the Faculty. Increasing collaboration with the surrounding economy is one of the goals outlined in the draft of the Development Strategy of the Faculty of Electrical Engineering, Computer Science and Information Technology in Osijek 2021–2025.

To achieve this, scientific and professional conferences are organised, and interested experts are always invited to participate. Examples include international scientific conferences such as the International Conference on Smart Systems and Technologies (https://sst-conference.org), the International Conference on Organisation and Technology of Maintenance (https://oto2021.panon.eu), and the Cyber Security Conference (https://csc.ferit.hr). Additionally, occasional presentations by economic and other entities are organised, involving discussions with students and faculty through panel discussions and guest lectures.

The Faculty of Electrical Engineering, Computer Science and Information Technology Osijek makes efforts to establish a connection between employers and students through the "Open Doors and Career Day" (DOVIK), which is organised every spring. Efforts are made to present high school and university students with:

- everything the Faculty offers, primarily through presentations in laboratories,

- everything that companies offer, i.e., where they can find employment after completing their studies, presented through 15-minute presentations or interesting demonstrations of activities that companies engage in.

A portal for FERIT students and employers, called STUP (https://stup.ferit.hr/), has been established. STUP, connecting our students and employers, was launched on 1 May 2016. Through this portal, companies can directly inform students about opportunities for:

- Employment
- Scholarships
- Completion of their final papers/Master's theses in the company
- Practical training
- As well as publish all other non-commercial content and non-commercial activities of interest to our students.

Additionally, companies have access to data on students interested in collaboration, and the Faculty informs companies about academic and extracurricular activities they can engage in. The Faculty of Electrical Engineering, Computer Science and Information Technology Osijek also organises guest lectures held by companies, which employers find highly beneficial and willingly participate in. As of January 2023, the statistics on STUP were as follows:

- 500 companies using the portal, of which
- 69 companies offered practical training
- 268 practical training positions were available, and 210 students completed their practical training in 64 companies.

Companies can publish all other non-commercial content and non-commercial activities that are of interest to our students. One of the most significant collaborations between the Faculty and companies is established through students' final papers and Master's theses. Companies can propose topics for Master's theses and final papers or join as mentors to a topic defined by our faculty mentors.

The connection between the Faculty and representatives of the labour market is also evident in the recognition of the best students on the occasion of the Faculty Day. For example, in 2023, companies such as H&MV, Infineon, AKD, Atron, CROZ-INTHEOS, Danieli-Systec d.o.o., Diverto d.o.o., Ekonerg d.o.o., FINA, HGK, HKIE, HOPS d.d., Hrvatska poštanska banka d.d., Hrvatski telekom d.d., Končar, MONTELEKTRO, Odašiljači i veze d.o.o., OG Consultancy Services d.o.o., SPAN, TEO-Belišće d.o.o., TTTech Auto d.o.o., Visage Technologies d.o.o., Yazaki, rewarded our students.

Faculty teachers are actively engaging in other scientific and professional projects involving collaboration with the industry. During the 2022/2023 academic year, agreements on collaboration were signed with the following companies: Atron, Montelektro, INTIS Engineering, and HEP-NOC.

Additionally, five representatives from the labour market, including the president of the FERIT Alumni Club are members of the Committee for Quality Enhancement and Assurance in Higher Education which periodically provides recommendations for study programmes with regards to trends and requirements of the labour market.

In terms of collaboration with the industry, accredited laboratories for testing low-frequency and high-frequency electromagnetic fields play a significant role. The Electromagnetic Compatibility Laboratory and the Laboratory for High Frequency Measurements have successfully undergone reaccreditation this year and have carried out a certain number of professional tasks. The specificities of these laboratories are highlighted on the Faculty's website under "Collaboration with the Industry" and in separate promotional flyers for each laboratory: https://www.ferit.unios.hr/znanost-i-suradnja/suradnja-s-gospodarstvom.

2.15. Information system for collection, management, processing, and reporting of statistical data related to the organisation and implementation of study programmes and those needed for quality assurance

The Higher Education Institutions Information System (ISVU) enables, among other things:

- Generation of ad-hoc reports on student success, exam pass rates;
- Review of entered data based on specific criteria (e.g., top 10% of students, average grades in exams);
- Drawing up reports (periodically required by the Ministry of Science and Education, the University, etc.).

ISVU is a solution for coordinated informatisation of all higher education institutions in the Republic of Croatia. It is primarily an application for the computerisation of student-related activities at a higher education institution, allowing the management of databases related to students, teachers, courses, curricula, enrolments, and exams. Additionally, the application supports standard activities of any higher education institution, such as student enrolments, exam registrations, grade entries, issuance of certificates and documents, and automatically generates summary reports.

Study programmes are also defined through MOZVAG (a module for higher education institutions of the Agency for Science and Higher Education). The MOZVAG system is an independent web application that facilitates the preparation and assessment of teaching staff and material conditions for the implementation of study programmes.

In addition, the Faculty has a system called “Mrkve” where teachers input reports on conducted classes, which are automatically compared with the syllabus implementation plan. The “Mrkve” system is also linked to the schedule of classes and exams, which is digitally accessible on the Faculty's website: https://www.ferit.unios.hr/studenti/raspored-nastave-i-ispita#predodabir. This schedule includes information on the division of students into smaller groups for various forms of teaching, such as laboratory exercises, etc.

Furthermore, the Faculty utilises the "Mak" system, which facilitates the process of conducting final and master's exams (including theses) from assigning topics to the automated generation of final reports and forms, as well as grading of students. Through "Mak", it is possible to monitor student progress through the final/master's exam process, document submission and retrieval, file submissions, as well as evaluation by mentors, the Commission for Master's Exams, and the Committee for Master’s Theses and Final Papers.The system also contains a repository of necessary documents and forms, serves as an information hub for students and teachers, and more.

All teaching materials for individual courses can be found on the e-learning system called Merlin (https://moodle.srce.hr).This platform also serves for communication between teachers and students, and all news related to specific courses is published there, with students receiving notifications via email.

Within the scope of the project "Application of Croatian Qualification Framework for University Study Programmes in the Field of Electrical Engineering - HKO-ELE", the existing system for creating the implementation plan and monitoring the implementation of study programmes has been improved and adapted. The functionalities have been expanded with the aim of enhancing the monitoring of quality indicators at FERIT. The mentioned improvements and system upgrades enable tracking the following data:

- Data on FERIT teachers:
  - Total number of employees;
  - Total number of employees for each academic title;
  - Detailed information by name and surname (name and surname, type of position, employment percentage, and salary coefficient).
- Pass rate data for all exam sessions:
  - For each study programme, each year of that programme, and each course in that programme:
  - Semester of implementation, number of students who took the exam, number of students who passed the exam, percentage, average grade, number of newly enrolled students, total number of enrolled students.
- Data on study success (for each study programme and each year of study, with the option to choose the academic year):
  - Number of enrolled students;
  - Number of first-time students in the study year compared to the number of first-time students in the previous study year / number of first-time students in the previous study year;
  - Number of repeaters/total number of enrolled in the same year of study in the previous academic year;
  - Average total number of earned ECTS credits per year of study (in parentheses is the number of students);
  - Individual student data.

2.16. Defining and publishing standards and regulations of the University regarding the periodic review of study programmes involving external experts

External experts are involved in the process of re-accreditation of the University every five years. The reaccreditation procedure at the Faculty of Electrical Engineering, Computer Science, and Information Technology Osijek was conducted in May 2018. The Accreditation Council of the Agency appointed an expert panel that included external experts. The reaccreditation was carried out based on the prepared self-evaluation, the Ordinance on the Content of Licence and Conditions for Issuing Licence for Performing Higher Education Activity, Carrying Out a Study Programme and Re-accreditation of Higher Education Institutions, as well as the Ordinance on Conditions for Issuing Licence for Scientific Activity, Conditions for Re-accreditation of Scientific Organisations and Content of Licence. The process was also based on Criteria for the Assessment of Quality of Higher Education Institutions within Universities of the Agency for Science and Higher Education.

Based on the re-accreditation evaluation (June 2018) and improvement recommendations, a Commission for the Development of a Quality Improvement Action Plan was established at the Faculty. The initial version of the Action Plan, created by the aforementioned Commission, was thoroughly analysed at a Quality Commission meeting. The final Action Plan was adopted at the 229th regular session of the Faculty Council held on 11 June 2019. In May 2022, the current version of the implementation of the Action Plan was submitted to the Agency for Science and Higher Education. The final version of the Action Plan implementation report was unanimously adopted at the 300th session of the Faculty Council held on 11 July 2023. Prior to that, the same version of the report was adopted at the Quality Commission meeting on 24 April 2023.

2.17. Defining and publishing standards and regulations for protecting students' rights with respect to informing students, receiving and resolving student complaints, and procedures for protecting rights; the way designated individuals handle students' rights issues.

Standards and regulations for the protection of students' rights, especially with respect to informing students, receiving and resolving student complaints, and procedures for protecting rights, are defined by the Ordinance on Studies and Studying at Josip Juraj Strossmayer University of Osijek.

At the Faculty, there is an organised Student Union. The Student Union is a student-elected representative body that protects the interests of students, participates in decision-making in the Faculty Council, and represents students in the higher education system. The Faculty's Student Union has a Statute, adopted on the proposal of the Student Union by the Faculty Council. The Statute of the Student Union determines the working methods, bodies, composition, method of election and jurisdiction of each body of the Student Union, the method of appointing the Student ombudsman, the method of electing student representatives to the Faculty bodies, the responsibility of the bodies and members of the Student Union for not fulfilling the tasks entrusted to them related to the work of the Student Union, as well as other matters important for the functioning of the Student Union.

The Student Union elects representatives of students to the Faculty Council. Student representatives constitute 10% of the members of the Faculty Council. During decision-making in the Faculty Council, student representatives have the right to a suspensive veto when deciding on matters such as changes in study conditions, amendments to study programmes and the implementation plan of studies, ensuring the quality of studies and issues related to student standards. Student representatives can use the suspensive veto when requested by a majority of all student representatives in the Faculty Council. After the suspensive veto, the Faculty Council re-discusses the issue, at the earliest within 8 days. In the repeated decision-making, the decision is made by a simple majority from the total number of voting members of the Faculty Council, without the right to use a suspensive veto.

The Student ombudsman is appointed by the Assembly of the Student Union upon the proposal of the President of the Student Union. The Student ombudsman receives complaints from students related to

their rights, discusses them with the relevant bodies of the Faculty, advises students on how to exercise their rights, and may participate in disciplinary proceedings against students to protect their rights.

The Vice-Dean for Education and Student Affairs is appointed by the Faculty Council upon the proposal of the Dean.

The Disciplinary Board for students consists of a president and two members, one of whom is a student. The Faculty Council appoints and dismisses the president and one member, along with their deputies from the ranks of the teaching staff. Additionally, the Student Union appoints and dismisses one member and their deputy from the ranks of students.

According to the Ordinance on Studies and Studying at Josip Juraj Strossmayer University of Osijek, Article 59, a student who is not satisfied with the received grade can, within 48 hours after the oral exam/oral part of the exam, or after the announcement of the results of a written exam, submit a request for retaking the exam before the Examination Committee.

2.18. Alignment with the requirements of professional associations (If the proposed study programme grants access to a regulated profession, it is necessary to align with national and European regulations, as well as recommendations from national and international professional associations)

The study programme provides access to a regulated profession.

2.19. Identify possible partners outside the higher education system

The proposed amendments to the graduate university study programme in Computer Engineering at the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek are largely based on the results of the project "Dig IT - Development of occupational standards and qualification standards in computer science" and on improved connections with the economy and monitoring of general technological developments. Through participation of external associates in field work, practical training, and writing of Master's theses, numerous partners, whose activities are in the field of computer engineering, would be involved in the proposed programme. Collaboration with these partners has already been established through the Stup portal[9].

[9] More information about the STUP portal can be found at: https://stup.ferit.hr/

3. GENERAL INFORMATION ABOUT THE STUDY PROGRAMME

3.1. Scientific/artistic area/field which the study programme falls into or interdisciplinary area of science/art with corresponding scientific/artistic fields

Scientific area: Technical Sciences, Scientific field: Computer Engineering

3.2. Duration of the study programme

The graduate university study programme in Computer Engineering lasts for two years (four semesters).

3.3. Minimum number of ECTS credits required for programme completion

To complete the graduate university study programme in Computer Engineering, a minimum of 120 ECTS credits prescribed by the study programme must be earned.

3.4. Possibility of further education (*defined procedures for recognising domestic and foreign higher education qualifications, study periods, and prior learning in case of continuing education*)

Upon completing the graduate university programme in Computer Engineering, students can continue their education at postgraduate university study programmes. At the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek, they can pursue further education in the postgraduate doctoral study programmes in Electrical Engineering and Computer Engineering. The vertical study scheme at FERIT is illustrated in Figure 1.

3.5. Language of instruction

The graduate university study programme in Computer Engineering is carried out in the Croatian language.

3.6. Admission requirements for the programme (*clearly defined admission criteria – evaluation of high school performance, State Matura levels, elective exams, additional assessments of knowledge and skills; defined decision-making procedures in relation to the admission criteria, compliance of admission requirements with the quality standards in the Croatian Qualifications Framework Register*)

Admission to the study programme is carried out based on a vacancy announcement.

Candidates eligible for enrolment in the graduate university study programme in Computer Engineering must have graduated from FERIT and have earned the following titles:

- University Bachelors of Computer Engineering;
- Bachelors of Computer Engineering who have enrolled and passed all differential exams at FERIT required for enrolment in the graduate university study programme in Computer Engineering

Furthermore, candidates eligible for enrolment in the graduate university study programme in Computer Engineering include:

- University Bachelors of Computer Engineering who graduated from other higher education institutions;

- University Bachelors with a specialisation in Technical or Natural Sciences.

In such cases, the Committee for Education and Student Affairs will determine differential exams to be passed.

The admission procedure consists of compiling a rank list of candidates based on the overall GPA, defined as the arithmetic mean of exam grades during the undergraduate studies (rounded to two decimal places). Admission quotas for individual modules are determined each academic year.

For graduates of other technical and related faculties, the Admissions Committee for Graduate Studies will define the possibility of admission and/or any required additional exams, depending on the curriculum of their undergraduate studies. If admission is approved, the candidate takes the additional exams in the first year of the graduate programme along with other exams specified in the curriculum and must pass them before enrolling in the second year of the programme.

For all Bachelor's degree holders from the Faculty of Electrical Engineering, Computer Science, and Information Technology Osijek or any related technical and professional studies and polytechnics, the condition for admission to the graduate university programme is the completion of differential obligations at the Faculty of Electrical Engineering, Computer Science, and Information Technology Osijek, as determined by the decision of the Faculty Council on the structure of differential obligations.

Candidates entering the graduate programme following the completion of differential exams at FERIT will have their GPA determined by averaging the grades obtained in the professional undergraduate programme and the grades earned in the differential exams (rounded to two decimal places).

3.7. Anticipated number of students in the study programme in the first year of study

The total anticipated number of students enrolling in this study programme has slightly increased compared to previous years, i.e. 112 students are planned to enrol in the first year of study. Given that these programme amendments introduce four modules, an admission quota is planned for each module as follows:

- Module: Computer Engineering -16 admissions
- Module: Artificial Intelligence and Robotics - 16 admissions
- Module: Software Engineering - 48 admissions
- Module: Data Science - 32 admissions

3.8. The learning outcomes at the programme level are determined in accordance with legal regulations and recommendations from professional associations (if applicable). They are aligned with the mission and strategic goals of the constituent unit, as well as the general objectives of the study programme. Furthermore, these outcomes are compatible with the quality standards in the Croatian Qualifications Framework Register and are designed to align with the competencies expected of students upon programme completion.

Below are the learning outcomes at the level of the study programme that are achieved upon completion of the graduate university study programme in Computer Engineering. Given that the study programme is divided into four modules, the learning outcomes will also be defined according to individual modules.

Module: Computer Engineering

1. **Design, develop, and implement technical solutions in the field of computer engineering to be applied in economic, industrial, and other sectors.**

2. **Independently and in a team, develop creative and systemic solutions to complex engineering problems in the field of computer engineering.**
3. **Identify scientific and technical challenges in the field of computer engineering, formulate and elaborate hypotheses in accordance with recommendations for writing professional and scientific publications.**
4. **Manage and lead engineering and development teams of experts in the implementation of projects in the field of computer engineering.**
5. Determine and apply current scientific methods and knowledge, as well as develop new approaches to the design and realisation of purpose-specific technical solutions in the fields of computer engineering and artificial intelligence, applicable to diversified technical systems and environments.
6. By means of real-time computer systems, design and implement purpose-specific solutions in embedded, distributed, and ubiquitous computer-controlled environments based on technologies and methods for universal connectivity and management in technical processes.
7. Design hardware and corresponding drivers for dedicated computer systems and related technical solutions.
8. Design, apply, and evaluate diverse computer systems in technical processes, with increased reliability, based on hardware and software solutions for functional redundancy and fault detection.
9. Design, implement, and evaluate specialised computer systems based on embedded computer architectures, microcontrollers, and digital signal processors tailored to specific application requirements.
10. Design, apply, and evaluate dedicated computer systems for digital signal processing based on highly specialised hardware and software solutions, and current scientific knowledge and methods in the field of digital signal processing.
11. Design, implement, and evaluate fundamental communication network structures and devices for operation and service in complex computer environments.
12. Design, implement, and evaluate client-server application service network solutions based on current diverse internet technologies, as well as other specialised software solutions in universal connectivity and ubiquitous computing environments.

Module: Artificial Intelligence and Robotics

1. **Design, develop and implement technical solutions in the field of computer engineering to be applied in economic, industrial and other sectors.**
2. **Individually and as a team, provide creative and systematic solutions to complex engineering problems in the field of computer engineering.**
3. **Determine scientific and technical challenges in the field of computer engineering, form and elaborate on hypotheses according to the recommendations for writing professional and research papers.**
4. **Manage and lead engineering and development teams of experts in the implementation of projects in the field of computer engineering.**
5. Choose appropriate optimisation methods and adapt them to solve a specific optimisation problem.
6. Design and create a computer vision system for image and data processing obtained by 3D sensors to be applied in robotics, automation and autonomous vehicles.
7. Choose a suitable artificial intelligence method for solving technical problems in the field of signal processing and process management.
8. Choose and apply the appropriate machine learning technique in solving problems in the field of supervised, unsupervised and supported learning.
9. Create control software for robot manipulators and mobile robots.

10. Integrate signal processing, control and artificial intelligence algorithms as well as software and hardware components into robotic systems, autonomous vehicles and industrial production processes.
11. Design and build an automatic control system.
12. Design and build an embedded computer system.
13. Computerise and automate complex technical systems.
14. Build models of objects and complex processes for the purposes of simulation and solving technical problems.

Module: Software Computing

1. Design, develop and implement technical solutions in the field of computer engineering to be applied in economic, industrial and other sectors.
2. Individually and as a team, provide creative and systematic solutions to complex engineering problems in the field of computer engineering.
3. Determine scientific and technical challenges in the field of computer engineering, form and elaborate on hypotheses according to the recommendations for writing professional and research papers.
4. Manage and lead engineering and development teams of experts in the implementation of projects in the field of computer engineering.
5. Develop complex applications using advanced algorithms and data structures in procedural and object-oriented programming languages.
6. Recommend the approach, models and techniques of software engineering and apply them in complex software development.
7. Design and develop comprehensive software solutions, such as web and mobile applications, corresponding advanced databases, and software and computer systems such as those in biomedicine and healthcare, real-time systems, information systems, image and text processing systems .
8. Determine appropriate approaches, methods and technologies for the development of software support for different types of distributed computer systems, including energy-efficient, heterogeneous, self-sustaining, self-adaptive and high-performance systems.
9. Develop software for data analysis characterised with methods and technologies of artificial intelligence and develop effective methods of presenting them.
10. Lead the software development process and independently take responsibility for strategic decision-making and successful task implementation.
11. Evaluate software and ensure its quality and the quality of the development procedure.
12. Adapt software solutions for working in advanced computer systems such as cloud computing, virtual and contained environments using methods of analysing large data sets and data warehouses.

Module: Data Sciences

1. Design, develop and implement technical solutions in the field of computer engineering to be applied in economic, industrial and other sectors.
2. Individually and as a team, provide creative and systematic solutions to complex engineering problems in the field of computer engineering.
3. Determine scientific and technical challenges in the field of computer engineering, form and elaborate on hypotheses according to the recommendations for writing professional and research papers.
4. Manage and lead engineering and development teams of experts in the implementation of projects in the field of computer engineering.

5. Recommend adequate methods and data analysis tools in systematic solving of complex engineering problems of software engineering and evaluate the analysis results.
6. Assess the suitability of different methods in data sciences, apply appropriate methods in solving problems and evaluate the efficiency of a particular algorithm.
7. Plan and manage the implementation of development activities, make decisions in the field of computer engineering and evaluate the successful task implementation.
8. Propose effective methods and design for the presentation of data and research results as a means of support in decision-making, thesis argumentation, prediction and comparison.
9. Connect acquired knowledge and apply methods for signal processing and interpret results.
10. Propose and implement quantum computing algorithms and apply methods for correcting quantum errors.
11. Design, project, create and evaluate software solutions in web programming, mobile applications, databases, Internet of things, 3D computer graphics, computer games development, natural language processing, robotics, blockchain, cryptocurrencies and bioinformatics.

3.9. Employability assessment upon graduation, involving the opinions of three organisations in the labour market

According to official data from the Croatian Employment Service (CES) at the beginning of January 2023, it is noticeable that the number of unemployed individuals has been very low in recent years. The following tables present data on the number of unemployed individuals at the beginning of the year (2023), the number of newly registered and employed individuals from the records during the year (all aged up to 39) in the Regional Centre Osijek.

Table 1. The number of newly enrolled students in the graduate university programme in Computer Engineering

The number of newly enrolled students in the graduate university programme in Computer Engineering					
2018	**2019**	**2020**	**2021**	**2022**	**Total**
60	**73**	**85**	**85**	**95**	**398**

Table 2. The number of newly registered and employed individuals from the records during the year (all aged up to 39) in the Regional Centre Osijek

Newly registered with the Croatian Employment Service aged up to 39 years						**Newly registered with the Croatian Employment Service aged up to 39 years and employed within 6 months**					
2018	**2019**	**2020**	**2021**	**2022**	**Total**	**2018**	**2019**	**2020**	**2021**	**2023**	**Total**
n/a	**6**	**16**	**14**	**16**	**52**	**n/a**	**5**	**13**	**15**	**17**	**50**

Based on the previously mentioned data, it can be observed that students who complete the graduate study programme in Computer Engineering tend to find employment within 6 months, indicating a high employability rate for graduates of this study programme. Furthermore, the majority of students who finish this study programme do not even register with the Croatian Employment Service; instead, they immediately find employment upon completing their studies.

Positive opinions from at least three organisations related to the labour market are attached in Appendix 7.3.

3.10. The study/studies of a lower level offered by the proposer or other institutions of the Republic of Croatia that allow enrolment in the proposed study *(e.g., undergraduate studies from*

***which enrolment in the graduate university study is possible if the proposed study programme is suggested*).**

FERIT conducts a lower-level study, the undergraduate university programme in Computer Engineering, the completion of which enables enrolment in the proposed study programme.

The document regarding the accredited undergraduate university programme in Computer Engineering is included in Appendix 7.9.

4. DESCRIPTION OF THE STUDY PROGRAMME

4.1. Provide a list of compulsory and elective courses with the number of contact hours required for their implementation and the number of ECTS credits.

Table 3. List of compulsory and elective courses for the Computer Engineering module

Module: Computer Engineering							
Year of study: 1							
Semester: 1							
COURSE	**LEAD INSTRUCTOR(S)**	**ECTS**	**L**	**AE**	**LE**	**DE**	**STATUS**
Statistical Data Analysis Methods	Anita Katić, PhD, Assistant Professor	**6**	**30**	**0**	**20**	**10**	**C**
Computer Systems Design	Ivan Vidović, PhD, Assistant Professor	**6**	**45**	**0**	**30**	**0**	**C**
DSP Algorithms and Architecture	Tomislav Matić (Jr.), PhD, Associate Professor Ivan Aleksi, PhD, Associate Professor	**5**	**30**	**15**	**15**	**0**	**C**
Software Quality Assurance	Damir Blažević, PhD, Full Professor	**7**	**30**	**15**	**15**	**0**	**C**
Cloud Computing and Data Analysis	Goran Martinović, PhD, Full Professor	**6**	**30**	**0**	**30**	**0**	**C**
Semester: 2							
COURSE	**LEAD INSTRUCTOR(S)**	**ECTS**	**L**	**AE**	**LE**	**DE**	**STATUS**
Real-Time Computer Systems	Goran Martinović, PhD, Full Professor	**7**	**45**	**0**	**30**	**0**	**C**
Embedded Computer Systems	Tomislav Keser, PhD, Associate Professor	**6**	**30**	**0**	**30**	**0**	**C**
Introduction to Artificial Intelligence	Damir Blažević, PhD, Full Professor	**6**	**30**	**15**	**30**	**0**	**C**
Ubiquitous Computing	Ivan Aleksi, PhD, Associate Professor Tomislav Matić (Jr.), PhD, Associate Professor	**5**	**30**	**0**	**30**	**0**	**C**
Integration of Digital Systems	Tomislav Keser, PhD, Associate Professor	**6**	**30**	**0**	**15**	**30**	**C**
Year of study: 2							

Semester: 3							
COURSE	**LEAD INSTRUCTOR(S)**	**ECTS**	**L**	**AE**	**LE**	**DE**	**STATUS**
Computer System Reliability and Diagnostics	Tomislav Matić (Jr.), PhD, Associate Professor	**6**	**45**	**15**	**15**	**0**	**C**
Computer Networks Design	Damir Blažević, PhD, Full Professor	**5**	**30**	**0**	**30**	**0**	**C**
Discrete Mathematics	Tomislav Rudec, PhD, Assistant Professor	**5**	**30**	**30**	**0**	**0**	**C**
Advanced Web Programming	Ivica Lukić, PhD, Associate Professor	**5**	**30**	**0**	**30**	**0**	**C**
Project Management	Dominika Crnjac-Milić, PhD, Full Professor	**4**	**30**	**15**	**0**	**0**	**C**
Elective course 1	-	**Min. 5**	**-**	**-**	**-**	**-**	**E**
Potential elective course:							
Software Support Testing Methods and Techniques	Goran Martinović, PhD, Full Professor Ivan Vidović, PhD, Assistant Professor	**6**	**30**	**0**	**30**	**0**	**E**
Robot Programming and Simulation	Damir Filko, PhD, Associate Professor	**6**	**30**	**0**	**30**	**0**	**E**
Basics of Robotics	Robert Cupec, PhD, Full Professor	**6**	**30**	**0**	**15**	**15**	**E**
Image Processing and Computer Vision	Irena Galić, PhD, Full Professor	**6**	**45**	**0**	**30**	**0**	**E**
Advanced Algorithms and Data Structures	Alfonzo Baumgartner, PhD, Associate Professor Tomislav Galba, PhD, Assistant Professor	**6**	**30**	**15**	**15**	**0**	**E**
Optimisation Methods	Emmanuel Karlo Nyarko, PhD, Associate Professor	**6**	**30**	**0**	**30**	**0**	**E**
Computer Control of Systems	Robert Cupec, PhD, Full Professor	**6**	**45**	**0**	**15**	**15**	**E**
Semester: 4							
COURSE	**LEAD INSTRUCTOR(S)**	**ECTS**	**L**	**AE**	**LE**	**DE**	**STATUS**
Practical Training		**10**					**C**
Master's Thesis		**20**					**C**

Table 4. List of compulsory and elective courses for the Artificial Intelligence and Robotics module

Module: Artificial Intelligence and Robotics							
Year of study: 1							
Semester: 1							
COURSE	**LEAD INSTRUCTOR(S)**	**ECTS**	**L**	**AE**	**LE**	**DE**	**STATUS**
Computer Control of Systems	Robert Cupec, PhD, Full Professor	**6**	**45**	**0**	**15**	**15**	**C**
Basics of Robotics	Robert Cupec, PhD, Full Professor	**6**	**30**	**0**	**15**	**15**	**C**
Image Processing and Computer Vision	Irena Galić, PhD, Full Professor	**6**	**45**	**0**	**30**	**0**	**C**
Robot Programming and Simulation	Damir Filko, PhD, Associate Professor	**6**	**30**	**0**	**30**	**0**	**C**
Optimisation Methods	Emmanuel Karlo Nyarko, PhD, Associate Professor	**6**	**30**	**0**	**30**	**0**	**C**
Semester: 2							
COURSE	**LEAD INSTRUCTOR(S)**	**ECTS**	**L**	**AE**	**LE**	**DE**	**C**
Introduction to Artificial Intelligence	Damir Blažević, PhD, Full Professor	**6**	**30**	**15**	**30**	**0**	**C**
Embedded Computer Systems	Tomislav Keser, PhD, Associate Professor	**6**	**30**	**0**	**30**	**0**	**C**
Robot Vision	Robert Cupec, PhD, Full Professor	**6**	**30**	**0**	**15**	**15**	**C**
Data-Based Modelling	Dražen Slišković, PhD, Full Professor	**6**	**30**	**0**	**30**	**0**	**C**
Machine Learning	Ratko Grbić, PhD, Associate Professor Emmanuel Karlo Nyarko, PhD, Associate Professor	**6**	**45**	**0**	**30**	**0**	**C**
Year of study: 2							
Semester: 3							
COURSE	**LEAD INSTRUCTOR(S)**	**ECTS**	**L**	**AE**	**LE**	**DE**	**C**
Autonomous Mobile Robots	Robert Cupec, PhD, Full Professor	**5**	**30**	**0**	**30**	**15**	**C**

	Damir Filko, PhD, Associate Professor						
Advanced Robot Systems	Robert Cupec, PhD, Full Professor Damir Filko, PhD, Associate Professor	**5**	**30**	**0**	**30**	**15**	**C**
Industrial Informatics	Dražen Slišković, PhD, Full Professor	**5**	**30**	**0**	**30**	**0**	**C**
Deep Learning	Ratko Grbić, PhD, Associate Professor Emmanuel Karlo Nyarko, PhD, Associate Professor	**6**	**45**	**0**	**30**	**0**	**C**
Project Management	Dominika Crnjac-Milić, PhD, Full Professor	**4**	**30**	**15**	**0**	**0**	**C**
Elective course 1	-	**Min. 5**	**-**	**-**	**-**	**-**	**E**
Potential elective course:							
Basics of Bioinformatics	Emmanuel Karlo Nyarko, PhD, Associate Professor Mario Lovrić, PhD, Assistant Professor	**6**	**30**	**0**	**30**	**0**	**E**
Natural Language Processing	Josip Job, PhD, Full Professor Petra Pejić, PhD, Assistant Professor	**6**	**30**	**0**	**30**	**0**	**E**
Computer Systems Design	Ivan Vidović, PhD, Assistant Professor	**6**	**45**	**0**	**30**	**0**	**E**
Cloud Computing and Data Analysis	Goran Martinović, PhD, Full Professor	**6**	**30**	**0**	**30**	**0**	**E**
Blockchain Technology and Cryptocurrencies	Mirko Köhler, PhD, Associate Professor	**5**	**30**	**0**	**30**	**0**	**E**
Internet of Things	Josip Job, PhD, Full Professor Ratko Grbić, PhD, Associate Professor	**5**	**30**	**0**	**15**	**15**	**E**
Statistical Data Analysis Methods	Anita Katić, PhD, Assistant Professor	**6**	**30**	**0**	**20**	**10**	**E**
Discrete Mathematics	Tomislav Rudec, PhD, Assistant Professor	**5**	**30**	**30**	**0**	**0**	**E**
Advanced Web Programming	Ivica Lukić, PhD, Associate Professor	**5**	**30**	**0**	**30**	**0**	**E**
System Programming	Alfonzo Baumgartner, PhD, Associate Professor	**6**	**45**	**0**	**15**	**0**	**E**
Distributed Computer Systems	Goran Martinović, PhD, Full Professor	**6**	**45**	**0**	**15**	**0**	**E**
NoSQL Databases	Ivica Lukić, PhD, Associate Professor (Krešimir Romić, PhD, Assistant Professor)	**5**	**30**	**0**	**30**	**0**	**E**
Advanced Algorithms and Data Structures	Alfonzo Baumgartner, PhD, Associate Professor Tomislav Galba, PhD, Assistant Professor	**6**	**30**	**15**	**15**	**0**	**E**

Semester: 4							
COURSE	**LEAD INSTRUCTOR(S)**	**ECTS**	**L**	**AE**	**LE**	**DE**	**STATUS**
Practical Training		**10**					**C**
Master's Thesis		**20**					**C**

Table 5. List of compulsory and elective courses for the Software Engineering module

Module: Software Engineering							
Year of study: 1							
Semester: 1							
COURSE	**LEAD INSTRUCTOR(S)**	**ECTS**	**L**	**AE**	**LE**	**DE**	**STATUS**
System Programming	Alfonzo Baumgartner, PhD, Associate Professor	**6**	**45**	**0**	**15**	**0**	**C**
Software Support Testing Methods and Techniques	Goran Martinović, PhD, Full Professor Ivan Vidović, PhD, Assistant Professor	**6**	**30**	**0**	**30**	**0**	**C**
Modelling and Design of Software Systems	Zdravko Krpić, PhD, Associate Professor	**6**	**30**	**15**	**15**	**0**	**C**
Cloud Computing and Data Analysis	Goran Martinović, PhD, Full Professor	**6**	**30**	**0**	**30**	**0**	**C**
Advanced Data Structures and Algorithms	Alfonzo Baumgartner, PhD, Associate Professor Tomislav Galba, PhD, Assistant Professor	**6**	**30**	**15**	**15**	**0**	**C**
Semester: 2							
COURSE	**LEAD INSTRUCTOR(S)**	**ECTS**	**L**	**AE**	**LE**	**DE**	**STATUS**
Real-Time Computer Systems	Goran Martinović, PhD, Full Professor	**7**	**45**	**0**	**30**	**0**	**C**
Mobile Applications Development	Josip Balen, PhD, Associate Professor	**5**	**30**	**0**	**30**	**15**	**C**
Data Visualisation	Josip Job, PhD, Full Professor	**5**	**30**	**0**	**15**	**15**	**C**
Web Programming	Krešimir Nenadić, PhD, Full Professor	**7**	**45**	**15**	**15**	**0**	**C**

Parallel Programming	Zdravko Krpić, PhD, Associate Professor Tomislav Matić (Jr.), PhD, Associate Professor	**6**	**30**	**0**	**30**	**0**	**C**
Year of study: 2							
Semester: 3							
COURSE	**LEAD INSTRUCTOR(S)**	**ECTS**	**L**	**AE**	**LE**	**DE**	**STATUS**
Distributed Computer Systems	Goran Martinović, PhD, Full Professor	**6**	**45**	**0**	**15**	**0**	**C**
Advanced Web Programming	Ivica Lukić, PhD, Associate Professor	**5**	**30**	**0**	**30**	**0**	**C**
NoSQL Databases	Ivica Lukić, PhD, Associate Professor (Krešimir Romić, PhD, Assistant Professor)	**5**	**30**	**0**	**30**	**0**	**C**
Project Management	Dominika Crnjac-Milić, PhD, Full Professor	**4**	**30**	**15**	**0**	**0**	**C**
Elective course 1	-	**Min. 10**	**-**	**-**	**-**	**-**	**E**
Elective course 2	-		**-**	**-**	**-**	**-**	**E**
Potential elective course:							
Development of Start-Up Technologies	Josip Balen, PhD, Associate Professor	**5**	**30**	**0**	**15**	**15**	**E**
Intelligent Transportation Systems	Josip Balen, PhD, Associate Professor	**5**	**30**	**0**	**30**	**0**	**E**
iOS Programming	Krešimir Nenadić, PhD, Full Professor	**5**	**30**	**0**	**30**	**0**	**E**
Blockchain Technology and Cryptocurrencies	Mirko Köhler, PhD, Associate Professor	**5**	**30**	**0**	**30**	**0**	**E**
Computer System Reliability and Diagnostics	Tomislav Matić, PhD, Associate Professor	**6**	**45**	**15**	**15**	**0**	**E**
Software Quality Assurance	Damir Blažević, PhD, Full Professor	**7**	**30**	**15**	**15**	**0**	**E**
Natural Language Processing	Josip Job, PhD, Full Professor Petra Pejić, PhD, Assistant Professor	**6**	**30**	**0**	**30**	**0**	**E**
Computer Systems Design	Ivan Vidović, PhD, Assistant Professor	**6**	**45**	**0**	**30**	**0**	**E**

Discrete Mathematics	Tomislav Rudec, PhD, Assistant Professor	**5**	**30**	**30**	**0**	**0**	**E**
Image Processing and Computer Vision	Irena Galić, PhD, Full Professor	**6**	**45**	**0**	**30**	**0**	**E**
Optimisation Methods	Emmanuel Karlo Nyarko, PhD, Associate Professor	**6**	**30**	**0**	**30**	**0**	**E**
Basics of Robotics	Robert Cupec, PhD, Full Professor	**6**	**30**	**0**	**15**	**15**	**E**
Semester: 4							
COURSE	**LEAD INSTRUCTOR(S)**	**ECTS**	**L**	**AE**	**LE**	**DE**	**STATUS**
Practical Training		**10**					**C**
Master's Thesis		**20**					**C**

Table 6. List of compulsory and elective courses for the Data Science module

Module: Data Science							
Year of study: 1							
Semester: 1							
COURSE	**LEAD INSTRUCTOR(S)**	**ECTS**	**L**	**AE**	**LE**	**DE**	**STATUS**
Advanced Data Structures and Algorithms	Alfonzo Baumgartner, PhD, Associate Professor Tomislav Galba, PhD, Assistant Professor	**6**	**30**	**15**	**15**	**0**	**C**
Statistical Data Analysis Methods	Anita Katić, PhD, Assistant Professor	**6**	**30**	**0**	**20**	**10**	**C**
Software Support Testing Methods and Techniques	Goran Martinović, PhD, Full Professor Ivan Vidović, PhD, Assistant Professor	**6**	**30**	**0**	**30**	**0**	**C**
Cloud Computing and Data Analysis	Goran Martinović, PhD, Full Professor	**6**	**30**	**0**	**30**	**0**	**C**
Image Processing and Computer Vision	Irena Galić, PhD, Full Professor	**6**	**45**	**0**	**30**	**0**	**C**
Semester: 2							
COURSE	**LEAD INSTRUCTOR(S)**	**ECTS**	**L**	**AE**	**LE**	**DE**	**STATUS**
Web Programming	Krešimir Nenadić, PhD, Full Professor	**7**	**45**	**15**	**15**	**0**	**C**

Quantum Computing	Irena Galić, PhD, Full Professor	**6**	**30**	**15**	**15**	**0**	**C**
Data Visualisation	Josip Job, PhD, Full Professor	**5**	**30**	**0**	**15**	**15**	**C**
Machine Learning	Ratko Grbić, PhD, Associate Professor Emmanuel Karlo Nyarko, PhD, Associate Professor	**6**	**45**	**0**	**30**	**0**	**C**
Computer Games Development	Časlav Livada, PhD, Associate Professor	**6**	**30**	**0**	**30**	**0**	**C**
Year of study: 2							
Semester: 3							
COURSE	**LEAD INSTRUCTOR(S)**	**ECTS**	**L**	**AE**	**LE**	**DE**	**STATUS**
Natural Language Processing	Josip Job, PhD, Full Professor Petra Pejić, PhD, Assistant Professor	**6**	**30**	**0**	**30**	**0**	**C**
Blockchain Technology and Cryptocurrencies	Mirko Köhler, PhD, Associate Professor	**5**	**30**	**0**	**30**	**0**	**C**
Internet of Things	Josip Job, PhD, Full Professor Ratko Grbić, PhD, Associate Professor	**5**	**30**	**0**	**15**	**15**	**C**
Research in Data Science	Irena Galić, PhD, Full Professor	**5**	**30**	**0**	**15**	**0**	**C**
Project Management	Dominika Crnjac-Milić, PhD, Full Professor	**4**	**30**	**15**	**0**	**0**	**C**
Elective course 1	-	**Min. 5**	**-**	**-**	**-**	**-**	**E**
Potential elective course:							
Deep Learning	Ratko Grbić, PhD, Associate Professor Emmanuel Karlo Nyarko, PhD, Associate Professor	**6**	**45**	**0**	**30**	**0**	**E**
Basics of Bioinformatics	Emmanuel Karlo Nyarko, PhD, Associate Professor Mario Lovrić, PhD, Assistant Professor	**6**	**30**	**0**	**30**	**0**	**E**
Advanced Web Programming	Ivica Lukić, PhD, Associate Professor	**5**	**30**	**0**	**30**	**0**	**E**

NoSQL Databases	Ivica Lukić, PhD, Associate Professor Krešimir Romić, PhD, Assistant Professor	**5**	**30**	**0**	**30**	**0**	**E**
3D Computer Graphics	Alfonzo Baumgartner, PhD, Associate Professor Irena Galić, PhD, Full Professor	**5**	**30**	**0**	**30**	**0**	**E**
Numerical Methods in Computer Science	Anita Katić, PhD, Assistant Professor	**5**	**30**	**0**	**15**	**15**	**E**
Optimisation Methods	Emmanuel Karlo Nyarko, PhD, Associate Professor	**6**	**30**	**0**	**30**	**0**	**E**
Development of Start-Up Technologies	Josip Balen, PhD, Associate Professor	**5**	**30**	**0**	**15**	**15**	**E**
iOS Programming	Krešimir Nenadić, PhD, Full Professor	**5**	**30**	**0**	**30**	**0**	**E**
Software Quality Assurance	Damir Blažević, PhD, Full Professor	**7**	**30**	**15**	**15**	**0**	**E**
Computer Systems Design	Ivan Vidović, PhD, Assistant Professor	**6**	**45**	**0**	**30**	**0**	**E**
Discrete Mathematics	Tomislav Rudec, PhD, Assistant Professor	**5**	**30**	**30**	**0**	**0**	**E**
Basics of Robotics	Robert Cupec, PhD, Full Professor	**6**	**30**	**0**	**15**	**15**	**E**
Semester: 4							
COURSE	**LEAD INSTRUCTOR(S)**	**ECTS**	**L**	**AE**	**LE**	**DE**	**STATUS**
Practical Training		**10**					**C**
Master's Thesis		**20**					**C**

4.2. Description of all courses

Detailed syllabi are available in Appendix 7.4.

4.3. Structure of the study programme

The graduate university study programme in Computer Engineering is structured in four semesters, i.e. two years. When enrolling in the aforementioned study programme, students have so far opted for one of four modules.

Pursuant to the proposed amendments to the graduate university study programme in Computer Engineering, students can choose one of the four different modules as follows:

- Computer Engineering;
- Artificial Intelligence and Robotics;
- Software Engineering;
- Data Science.

According to the selected module, the student enrols in module-specific courses as described below. Some courses, depending on the targeted competencies, can be offered at multiple modules. On the one hand, the structuring of elective courses in the form of modules enables the student to improve his/her skills according to his/her interests. On the other hand, by choosing elective courses, the student can specialise in a narrow field.

The study programme structure with a list of courses per semesters and modules is as follows.

Module Computer Engineering

Semester 1:

Statistical Data Analysis Methods
Computer Systems Design
DSP Algorithms and Architecture
Software Quality Assurance
Cloud Computing and Data Analysis

Semester 2:

Real-Time Computer Systems
Embedded Computer Systems
Introduction to Artificial Intelligence
Ubiquitous Computing
Integration of Digital Systems

Semester 3:

Computer System Reliability and Diagnostics
Computer Networks Design
Discrete Mathematics

Advanced Web Programming
Project Management
Elective courses*
Software Support Testing Methods and Techniques
Robot Programming and Simulation
Basics of Robotics
Image Processing and Computer Vision
Advanced Algorithms and Data Structures
Optimisation Methods
Computer Control of Systems

* Students opt for at least one elective course

Semester 4:

Practical Training
Master's Thesis

Module Artificial Intelligence and Robotics

Semester 1:

Computer Control of Systems
Basics of Robotics
Image Processing and Computer Vision
Robot Programming and Simulation
Optimisation Methods

Semester 2:

Introduction to Artificial Intelligence
Embedded Computer Systems
Robot Vision
Data-Based Modelling
Machine Learning

Semester 3:

Autonomous Mobile Robots
Advanced Robot Systems
Industrial Informatics
Deep Learning
Project Management
Elective courses*
Basics of Bioinformatics
Natural Language Processing

Computer Systems Design
Cloud Computing and Data Analysis
Blockchain Technology and Cryptocurrencies
Internet of Things
Statistical Data Analysis Methods
Discrete Mathematics
Advanced Web Programming
System Programming
Distributed Computer Systems
NoSQL Databases
Advanced Algorithms and Data Structures

* Students opt for at least one elective course

Semester 4:

Practical Training
Master's Thesis

Module Software Engineering

Semester 1:

System Programming
Software Support Testing Methods and Techniques
Computer Systems Modelling and Design
Cloud Computing and Data Analysis
Advanced Algorithms and Data Structures

Semester 2:

Real-Time Computer Systems
Mobile Applications Development
Data Visualisation
Web Programming
Parallel Programming

Semester 3:

Distributed Computer Systems
Advanced Web Programming
NoSQL Databases
Project Management
Elective courses**
Development of Start-Up Technologies
Intelligent Transportation Systems
iOS Programming

Blockchain Technology and Cryptocurrencies
Computer System Reliability and Diagnostics
Software Quality Assurance
Natural Language Processing
Computer Systems Design
Discrete Mathematics
Image Processing and Computer Vision
Optimisation Methods
Basics of Robotics

** Students opt for at least two elective courses

Semester 4:

Practical Training
Master's Thesis

Module Data Science

Semester 1:

Advanced Algorithms and Data Structures
Statistical Data Analysis Methods
Software Support Testing Methods and Techniques
Cloud Computing and Data Analysis
Image Processing and Computer Vision

Semester 2:

Web Programming
Quantum Computing
Data Visualisation
Machine Learning
Computer Games Development

Semester 3:

Natural Language Processing
Blockchain Technology and Cryptocurrencies
Internet of Things
Research in Data Science
Project Management
Elective course*
Deep Learning
Basics of Bioinformatics
Advanced Web Programming
NoSQL Databases

3D Computer Graphics
Numerical Methods in Computer Science
Optimisation Methods
Development of Start-Up Technologies
iOS Programming
Software Quality Assurance
Computer Systems Design
Discrete Mathematics
Basics of Robotics

* Students opt for at least one elective course

Semester 4:

Practical Training
Master's Thesis

Note:
• During the final semester, students enrolled in all modules only have practical training and a Master's Thesis. This allows students easier mobility and the opportunity to complete their training or thesis at another institution.
• In semesters 1, 2 and 3, students enrolled in any module can enrol in an elective course.

4.4. Enrolment requirements in the successive academic year and the mode of studying (study pace and student obligations, criteria for progressing through the programme, enrolment in the successive semester or academic year, and prerequisites for enrolling in specific courses or a group of courses).

The enrolment requirements in the successive academic year has been determined by the Ordinance on Studies and Studying of J.J. Strossmayer University of Osijek as well as the University Senate's Decision on Requirements for Enrolment in the Successive Year, and they refer to the following:

- regular fulfilment of obligations determined in the study programme
- the number of ECTS credits earned by passing the exams.

The graduate university study programme in Computer Engineering can be enrolled as a full-time study.

4.5. List of courses and/or modules students can enrol in other study programmes

Every academic year, students have the option to choose elective courses offered at Josip Juraj Strossmayer University of Osijek. For example, in the academic year 2022/2023, a total of 51 courses from 17 University constituent units were offered. The list is available at the following link: http://www.unios.hr/wp-content/uploads/2022/07/izborni-kolegiji-22-23.pdf

4.6. List of courses and/or modules which can be taught in a foreign language

All courses can be taught in the English language.

4.7. Criteria and conditions for the transfer of ECTS credits

The Faculty organises and carries out the Erasmus International Mobility Programme. The Erasmus International Mobility Programme enables students to spend one part of their studies at a foreign higher

education institution and/or undergo practical training, which significantly contributes to their independence, cultural enrichment, proficiency in foreign languages and ability to work in multicultural environments. Implementation and basic principles of incoming and outgoing student mobility, students' rights and obligations, rights and obligations of the University Committee for the Erasmus International Mobility Programme and the institutional Erasmus coordinator, as well as other questions relevant for the implementation of the mobility programme have been specified in the Regulations on the Erasmus Mobility Programme. On the recommendation of the Erasmus coordinator, the Committee for Education and Student Affairs lays down the criteria and conditions for ECTS recognition for students participating in the Mobility Programme.

4.8. Requirements for resuming interrupted studies

Students who have interrupted their studies can apply for a resumption of their studies. The decision regarding the approval of resuming studies lies with the Committee for Education and Student Affairs at FERIT, which takes into consideration any changes to the study programme compared to the one the student had interrupted.
The conditions under which students who have lost their full-time status can continue their studies as part-time students are prescribed by the Ordinance on Studies and Studying:

- Article 39: A student who has lost his/her full-time status cannot re-enrol in the same study programme or continue studying in the same study programme, however; he/she may choose to re-enrol or continue his/her studies as a part-time student.
- Article 46: A student who has lost his/her full-time status due to repeating a study year can continue his/her studies as a part-time student with a time restriction, i.e., the obligation to complete the studies within twice the regular duration of the programme.

4.9. Completion of the study programme

The graduate university study programme in Computer Engineering is completed when the student passes all exams and does other study-related requirements as well as defends a Master's thesis. Doing the Master's thesis, the student must prove that he/she is capable of applying the knowledge acquired during his/her studies and show that he/she can successfully solve tasks at the graduate level.

The details related to the preparation of the Master's thesis are regulated by the Faculty's Regulations on Final Papers and Master's Theses.

5. MONITORING THE QUALITY AND EFFECTIVENESS OF THE STUDY PROGRAMME IMPLEMENTATION

5.1. The quality assurance plan of the study programme must be developed in accordance with the standards and guidelines for quality assurance in the European Higher Education Area.

By accepting the Bologna Declaration, each organisational unit in the higher education system has committed to fostering European collaboration to ensure quality by establishing a quality assurance system. The Faculty of Electrical Engineering, Computer Science and Information Technology Osijek has implemented a system that undergoes constant monitoring and necessary modifications throughout the educational process to attain qualitative enhancements. The Faculty conducts continuous internal assessments along with periodic external audits.

Internal control is ensured through the Committee for Quality Enhancement and Assurance in Higher Education, whose findings are imperative for the Faculty's management for qualitative changes.

Internal control is also based on student surveys, allowing students' direct participation in defining and raising quality standards.

External quality control is carried out periodically (according to legal regulations) or as urgently needed by independent accredited agencies for monitoring the quality of higher education in the Republic of Croatia.

5.2.The quality assurance plan of the study programme includes an overview of quality assurance on which the quality system of the constituent unit is based. It is evident from this overview that the proposer of the study programme has ensured procedures for evaluating the work of teachers and associates, monitoring student evaluation, evaluating the availability of resources for the processes of learning and teaching, assessing student support, monitoring student pass rates, monitoring satisfaction of external and internal stakeholders participating in the study programme, and informing them about the study programme.

Through regular communication among employees and professional services, the quality of work provided by professional services is analysed and improved, and necessary measures are taken when required.

An important aspect of informing all employees is certainly the Intranet system where all relevant decisions, minutes and documents are published, making it easier for employees to perform their assigned tasks more efficiently.

Moreover, a survey is conducted to evaluate the performance of all professional services within the higher education institution.

6. APPENDICES

6.1. Decision of the Faculty Council on the amendments to the study programme

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK
JOSIP JURAJ STROSSMAYER UNIVERSITY OF OSIJEK
FACULTY OF ELECTRICAL ENGINEERING, COMPUTER SCIENCE AND INFORMATION TECHNOLOGY OSIJEK
FERIT

KLASA: 602-01/23-08/00022
URBROJ: 2158-80-17-23-00031
Osijek, 12. prosinca 2023. godine

Na temelju članka 30. Statuta Fakulteta elektrotehnike, računarstva i informacijskih tehnologija Osijek, Fakultetsko vijeće Fakulteta elektrotehnike, računarstva i informacijskih tehnologija Osijek, na 309. sjednici (5. sjednici Fakultetskog vijeća u akademskoj 2023./2024. godini) održanoj 12. prosinca 2023. godine, pod točkom 12. dnevnog reda, donijelo je sljedeću

ODLUKU

1. Prihvaća se prijedlog izmjena i dopuna studijskog programa sveučilišnog diplomskog studija Računarstvo na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija Osijek.
2. Izmjene i dopune studijskog programa sveučilišnog diplomskog studija Računarstvo dostavljaju se Povjerenstvu za prijediplomske, diplomske i stručne studije Sveučilišta Josipa Jurja Strossmayera u Osijeku na daljnji postupak.
3. Ova Odluka stupa na snagu danom donošenja.

Dekan
Prof. dr. sc. Tomislav Matić

Dostaviti:
1. Povjerenstvo za prijediplomske, diplomske i stručne studije
2. Pismohrana Fakultetskog vijeća
3. Pismohrana Fakulteta

6.2. List of compulsory and elective courses with the number of contact hours required for their implementation and the number of ECTS credits.

Module: Computer Engineering							
Year of study: 1							
Semester: 1							
COURSE	**LEAD INSTRUCTOR(S)**	**ECTS**	**L**	**AE**	**LE**	**DE**	**STATUS**
Statistical Data Analysis Methods	Anita Katić, PhD, Assistant Professor	**6**	**30**	**0**	**20**	**10**	**C**
Computer Systems Design	Ivan Vidović, PhD, Assistant Professor	**6**	**45**	**0**	**30**	**0**	**C**
DSP Algorithms and Architecture	Tomislav Matić (Jr.), PhD, Associate Professor Ivan Aleksi, PhD, Associate Professor	**5**	**30**	**15**	**15**	**0**	**C**
Software Quality Assurance	Damir Blažević, PhD, Full Professor	**7**	**30**	**15**	**15**	**0**	**C**
Cloud Computing and Data Analysis	Goran Martinović, PhD, Full Professor	**6**	**30**	**0**	**30**	**0**	**C**
Semester: 2							
COURSE	**LEAD INSTRUCTOR(S)**	**ECTS**	**L**	**AE**	**LE**	**DE**	**STATUS**
Real-Time Computer Systems	Goran Martinović, PhD, Full Professor	**7**	**45**	**0**	**30**	**0**	**C**
Embedded Computer Systems	Tomislav Keser, PhD, Associate Professor	**6**	**30**	**0**	**30**	**0**	**C**
Introduction to Artificial Intelligence	Damir Blažević, PhD, Full Professor	**6**	**30**	**15**	**30**	**0**	**C**
Ubiquitous Computing	Ivan Aleksi, PhD, Associate Professor Tomislav Matić (Jr.), PhD, Associate Professor	**5**	**30**	**0**	**30**	**0**	**C**
Integration of Digital Systems	Tomislav Keser, PhD, Associate Professor	**6**	**30**	**0**	**15**	**30**	**C**
Year of study: 2							
Semester: 3							
COURSE	**LEAD INSTRUCTOR(S)**	**ECTS**	**L**	**AE**	**LE**	**DE**	**STATUS**
Computer System Reliability and Diagnostics	Tomislav Matić (Jr.), PhD, Associate Professor	**6**	**45**	**15**	**15**	**0**	**C**

Computer Networks Design	Damir Blažević, PhD, Full Professor	**5**	**30**	**0**	**30**	**0**	**C**
Discrete Mathematics	Tomislav Rudec, PhD, Assistant Professor	**5.**	**30**	**30**	**0**	**0**	**C**
Advanced Web Programming	Ivica Lukić, PhD, Associate Professor	**5**	**30**	**0**	**30**	**0**	**C**
Project Management	Dominika Crnjac-Milić, PhD, Full Professor	**4**	**30**	**15**	**0**	**0**	**C**
Elective course 1	-	**Min. 5**	**-**	**-**	**-**	**-**	**E**
Potential elective course:							
Software Support Testing Methods and Techniques	Goran Martinović, PhD, Full Professor Ivan Vidović, PhD, Assistant Professor	**6**	**30**	**0**	**30**	**0**	**E**
Robot Programming and Simulation	Damir Filko, PhD, Associate Professor	**6**	**30**	**0**	**30**	**0**	**E**
Basics of Robotics	Robert Cupec, PhD, Full Professor	**6**	**30**	**0**	**15**	**15**	**E**
Image Processing and Computer Vision	Irena Galić, PhD, Full Professor	**6**	**45**	**0**	**30**	**0**	**E**
Advanced Algorithms and Data Structures	Alfonzo Baumgartner, PhD, Associate Professor Tomislav Galba, PhD, Assistant Professor	**6**	**30**	**15**	**15**	**0**	**E**
Optimisation Methods	Emmanuel Karlo Nyarko, PhD, Associate Professor	**6**	**30**	**0**	**30**	**0**	**E**
Computer Control of Systems	Robert Cupec, PhD, Full Professor	**6**	**45**	**0**	**15**	**15**	**E**
Semestar: 4							
COURSE	**LEAD INSTRUCTOR(S)**	**ECTS**	**L**	**AE**	**LE**	**DE**	**STATUS**
Practical Training		**10**					**C**
Master's Thesis		**20**					**C**

Module: Artificial Intelligence and Robotics							
Year of study: 1							
Semester: 1							
COURSE	**LEAD INSTRUCTOR(S)**	**ECTS**	**L**	**AE**	**LE**	**DE**	**STATUS**
Computer Control of Systems	Robert Cupec, PhD, Full Professor	**6**	**45**	**0**	**15**	**15**	**C**
Basics of Robotics	Robert Cupec, PhD, Full Professor	**6**	**30**	**0**	**15**	**15**	**C**
Image Processing and Computer Vision	Irena Galić, PhD, Full Professor	**6**	**45**	**0**	**30**	**0**	**C**
Robot Programming and Simulation	Damir Filko, PhD, Associate Professor	**6**	**30**	**0**	**30**	**0**	**C**
Optimisation Methods	Emmanuel Karlo Nyarko, PhD, Associate Professor	**6**	**30**	**0**	**30**	**0**	**C**
Semester: 2							
COURSE	**LEAD INSTRUCTOR(S)**	**ECTS**	**L**	**AE**	**LE**	**DE**	**STATUS**
Introduction to Artificial Intelligence	Damir Blažević, PhD, Full Professor	**6**	**30**	**15**	**30**	**0**	**C**
Embedded Computer Systems	Tomislav Keser, PhD, Associate Professor	**6**	**30**	**0**	**30**	**0**	**C**
Robot Vision	Robert Cupec, PhD, Full Professor	**6**	**30**	**0**	**15**	**15**	**C**
Data-Based Modelling	Dražen Slišković, PhD, Full Professor	**6**	**30**	**0**	**30**	**0**	**C**
Machine Learning	Ratko Grbić, PhD, Associate Professor Emmanuel Karlo Nyarko, PhD, Associate Professor	**6**	**45**	**0**	**30**	**0**	**C**

Year of study: 2							
Semester: 3							
COURSE	**LEAD INSTRUCTOR(S)**	**ECTS**	**L**	**AE**	**LE**	**DE**	**STATUS**
Autonomous Mobile Robots	Robert Cupec, PhD, Full Professor Damir Filko, PhD, Associate Professor	**5**	**30**	**0**	**30**	**15**	**C**
Advanced Robot Systems	Robert Cupec, PhD, Full Professor Damir Filko, PhD, Associate Professor	**5**	**30**	**0**	**30**	**15**	**C**
Industrial Informatics	Dražen Slišković, PhD, Full Professor	**5**	**30**	**0**	**30**	**0**	**C**
Deep Learning	Ratko Grbić, PhD, Associate Professor Emmanuel Karlo Nyarko, PhD, Associate Professor	**6**	**45**	**0**	**30**	**0**	**C**
Project Management	Dominika Crnjac-Milić, PhD, Full Professor	**4**	**30**	**15**	**0**	**0**	**C**
Elective course 1	-	**Min. 5**	**-**	**-**	**-**	**-**	**E**
Potential elective course:							
Basics of Bioinformatics	Emmanuel Karlo Nyarko, PhD, Associate Professor Mario Lovrić, PhD, Assistant Professor	**6**	**30**	**0**	**30**	**0**	**E**
Natural Language Processing	Josip Job, PhD, Full Professor Petra Pejić, PhD, Assistant Professor	**6**	**30**	**0**	**30**	**0**	**E**
Computer Systems Design	Ivan Vidović, PhD, Assistant Professor	**6**	**45**	**0**	**30**	**0**	**E**
Cloud Computing and Data Analysis	Goran Martinović, PhD, Full Professor	**6**	**30**	**0**	**30**	**0**	**E**
Blockchain Technology and Cryptocurrencies	Mirko Köhler, PhD, Associate Professor	**5**	**30**	**0**	**30**	**0**	**E**
Internet of Things	Josip Job, PhD, Full Professor Ratko Grbić, PhD, Associate Professor	**5**	**30**	**0**	**15**	**15**	**E**
Statistical Data Analysis Methods	Anita Katić, PhD, Assistant Professor	**6**	**30**	**0**	**20**	**10**	**E**
Discrete Mathematics	Tomislav Rudec, PhD, Assistant Professor	**5**	**30**	**30**	**0**	**0**	**E**

Advanced Web Programming	Ivica Lukić, PhD, Associate Professor	**5**	**30**	**0**	**30**	**0**	**E**
System Programming	Alfonzo Baumgartner, PhD, Associate Professor	**6**	**45**	**0**	**15**	**0**	**E**
Distributed Computer Systems	Goran Martinović, PhD, Full Professor	**6**	**45**	**0**	**15**	**0**	**E**
NoSQL Databases	Ivica Lukić, PhD, Associate Professor (Krešimir Romić, PhD, Assistant Professor)	**5**	**30**	**0**	**30**	**0**	**E**
Advanced Algorithms and Data Structures	Alfonzo Baumgartner, PhD, Associate Professor Tomislav Galba, PhD, Assistant Professor	**6**	**30**	**15**	**15**	**0**	**E**
Semester: 4							
COURSE	**LEAD INSTRUCTOR(S)**	**ECTS**	**L**	**AE**	**LE**	**DE**	**STATUS**
Practical Training		**10**					**C**
Master's Thesis		**20**					**C**

Module: Software Engineering							
Year of study: 1							
Semester: 1							
COURSE	**LEAD INSTRUCTOR(S)**	**ECTS**	**L**	**AE**	**LE**	**DE**	**STATUS**
System Programming	Alfonzo Baumgartner, PhD, Associate Professor	**6**	**45**	**0**	**15**	**0**	**C**
Software Support Testing Methods and Techniques	Goran Martinović, PhD, Full Professor Ivan Vidović, PhD, Assistant Professor	**6**	**30**	**0**	**30**	**0**	**C**
Modelling and Design of Software Systems	Zdravko Krpić, PhD, Associate Professor	**6**	**30**	**15**	**15**	**0**	**C**
Cloud Computing and Data Analysis	Goran Martinović, PhD, Full Professor	**6**	**30**	**0**	**30**	**0**	**C**

Advanced Data Structures and Algorithms	Alfonzo Baumgartner, PhD, Associate Professor Tomislav Galba, PhD, Assistant Professor	**6**	**30**	**15**	**15**	**0**	**C**
Semester: 2							
COURSE	**LEAD INSTRUCTOR(S)**	**ECTS**	**L**	**AE**	**LE**	**DE**	**STATUS**
Real-Time Computer Systems	Goran Martinović, PhD, Full Professor	**7**	**45**	**0**	**30**	**0**	**C**
Mobile Applications Development	Josip Balen, PhD, Associate Professor	**5**	**30**	**0**	**30**	**15**	**C**
Data Visualisation	Josip Job, PhD, Full Professor	**5**	**30**	**0**	**15**	**15**	**C**
Web Programming	Krešimir Nenadić, PhD, Full Professor	**7**	**45**	**15**	**15**	**0**	**C**
Parallel Programming	Zdravko Krpić, PhD, Associate Professor Tomislav Matić (Jr.), PhD, Associate Professor	**6**	**30**	**0**	**30**	**0**	**C**
Year of study: 2							
Semester: 3							
COURSE	**LEAD INSTRUCTOR(S)**	**ECTS**	**L**	**AE**	**LE**	**DE**	**STATUS**
Distributed Computer Systems	Goran Martinović, PhD, Full Professor	**6**	**45**	**0**	**15**	**0**	**C**
Advanced Web Programming	Ivica Lukić, PhD, Associate Professor	**5**	**30**	**0**	**30**	**0**	**C**
NoSQL Databases	Ivica Lukić, PhD, Associate Professor (Krešimir Romić, PhD, Assistant Professor)	**5**	**30**	**0**	**30**	**0**	**C**
Project Management	Dominika Crnjac-Milić, PhD, Full Professor	**4**	**30**	**15**	**0**	**0**	**C**
Elective course 1	-	**Min. 10**	**-**	**-**	**-**	**-**	**E**
Elective course 2	-		**-**	**-**	**-**	**-**	**E**

Potential elective course:							
Development of Start-Up Technologies	Josip Balen, PhD, Associate Professor	**5**	**30**	**0**	**15**	**15**	**E**
Intelligent Transportation Systems	Josip Balen, PhD, Associate Professor	**5**	**30**	**0**	**30**	**0**	**E**
iOS Programming	Krešimir Nenadić, PhD, Full Professor	**5**	**30**	**0**	**30**	**0**	**E**
Blockchain Technology and Cryptocurrencies	Mirko Köhler, PhD, Associate Professor	**5**	**30**	**0**	**30**	**0**	**E**
Computer System Reliability and Diagnostics	Tomislav Matić (Jr.), PhD, Associate Professor	**6**	**45**	**15**	**15**	**0**	**E**
Software Quality Assurance	Damir Blažević, PhD, Full Professor	**7**	**30**	**15**	**15**	**0**	**E**
Natural Language Processing	Josip Job, PhD, Full Professor Petra Pejić, PhD, Assistant Professor	**6**	**30**	**0**	**30**	**0**	**E**
Computer Systems Design	Ivan Vidović, PhD, Assistant Professor	**6**	**45**	**0**	**30**	**0**	**E**
Discrete Mathematics	Tomislav Rudec, PhD, Assistant Professor	**5**	**30**	**30**	**0**	**0**	**E**
Image Processing and Computer Vision	Irena Galić, PhD, Full Professor	**6**	**45**	**0**	**30**	**0**	**E**
Optimisation Methods	Emmanuel Karlo Nyarko, PhD, Associate Professor	**6**	**30**	**0**	**30**	**0**	**E**
Basics of Robotics	Robert Cupec, PhD, Full Professor	**6**	**30**	**0**	**15**	**15**	**E**
Semester: 4							
COURSE	**LEAD INSTRUCTOR(S)**	**ECTS**	**L**	**AE**	**LE**	**DE**	**STATUS**
Practical Training		**10**					**C**
Master's Thesis		**20**					**C**

Module: Data Science							
Year of study: 1							
Semester: 1							
COURSE	**LEAD INSTRUCTOR(S)**	**ECTS**	**L**	**AE**	**LE**	**DE**	**STATUS**
Advanced Data Structures and Algorithms	Alfonzo Baumgartner, PhD, Associate Professor Tomislav Galba, PhD, Assistant Professor	**6**	**30**	**15**	**15**	**0**	**C**
Statistical Data Analysis Methods	Anita Katić, PhD, Assistant Professor	**6**	**30**	**0**	**20**	**10**	**C**
Software Support Testing Methods and Techniques	Goran Martinović, PhD, Full Professor Ivan Vidović, PhD, Assistant Professor	**6**	**30**	**0**	**30**	**0**	**C**
Cloud Computing and Data Analysis	Goran Martinović, PhD, Full Professor	**6**	**30**	**0**	**30**	**0**	**C**
Image Processing and Computer Vision	Irena Galić, PhD, Full Professor	**6**	**45**	**0**	**30**	**0**	**C**
Semester: 2							
COURSE	**LEAD INSTRUCTOR(S)**	**ECTS**	**L**	**AE**	**LE**	**DE**	**STATUS**
Web Programming	Krešimir Nenadić, PhD, Full Professor	**7**	**45**	**15**	**15**	**0**	**C**
Quantum Computing	Irena Galić, PhD, Full Professor	**6**	**30**	**15**	**15**	**0**	**C**
Data Visualisation	Josip Job, PhD, Full Professor	**5**	**30**	**0**	**15**	**15**	**C**
Machine Learning	Ratko Grbić, PhD, Associate Professor Emmanuel Karlo Nyarko, PhD, Associate Professor	**6**	**45**	**0**	**30**	**0**	**C**
Computer Games Development	Časlav Livada, PhD, Associate Professor	**6**	**30**	**0**	**30**	**0**	**C**
Year of study: 2							
Semester: 3							
COURSE	**LEAD INSTRUCTOR(S)**	**ECTS**	**L**	**AE**	**LE**	**DE**	**STATUS**

Natural Language Processing	Josip Job, PhD, Full Professor Petra Pejić, PhD, Assistant Professor	**6**	**30**	**0**	**30**	**0**	**C**
Blockchain Technology and Cryptocurrencies	Mirko Köhler, PhD, Associate Professor	**5**	**30**	**0**	**30**	**0**	**C**
Internet of Things	Josip Job, PhD, Full Professor Ratko Grbić, PhD, Associate Professor	**5**	**30**	**0**	**15**	**15**	**C**
Research in Data Science	Irena Galić, PhD, Full Professor	**5**	**30**	**0**	**15**	**0**	**C**
Project Management	Dominika Crnjac-Milić, PhD, Full Professor	**4**	**30**	**15**	**0**	**0**	**C**
Elective course 1	-	**Min. 5**	**-**	**-**	**-**	**-**	**E**
Potential elective course:							
Deep Learning	Ratko Grbić, PhD, Associate Professor Emmanuel Karlo Nyarko, PhD, Associate Professor	**6**	**45**	**0**	**30**	**0**	**E**
Basics of Bioinformatics	Emmanuel Karlo Nyarko, PhD, Associate Professor Mario Lovrić, PhD, Assistant Professor	**6**	**30**	**0**	**30**	**0**	**E**
Advanced Web Programming	Ivica Lukić, PhD, Associate Professor	**5**	**30**	**0**	**30**	**0**	**E**
NoSQL Databases	Ivica Lukić, PhD, Associate Professor (Krešimir Romić, PhD, Assistant Professor)	**5**	**30**	**0**	**30**	**0**	**E**
3D Computer Graphics	Alfonzo Baumgartner, PhD, Associate Professor, Irena Galić, PhD, Full Professor	**5**	**30**	**0**	**30**	**0**	**E**
Numerical Mathematics	Anita Katić, PhD, Assistant Professor	**5**	**30**	**0**	**15**	**15**	**E**
Optimisation Methods	Emmanuel Karlo Nyarko, PhD, Associate Professor	**6**	**30**	**0**	**30**	**0**	**E**
Development of Start-Up Technologies	Josip Balen, PhD, Associate Professor	**5**	**30**	**0**	**15**	**15**	**E**
iOS Programming	Krešimir Nenadić, PhD, Full Professor	**5**	**30**	**0**	**30**	**0**	**E**
Software Quality Assurance	Damir Blažević, PhD, Full Professor	**7**	**30**	**15**	**15**	**0**	**E**
Computer Systems Design	Ivan Vidović, PhD, Assistant Professor	**6**	**45**	**0**	**30**	**0**	**E**
Discrete Mathematics	Tomislav Rudec, PhD, Assistant Professor	**5**	**30**	**30**	**0**	**0**	**E**
Basics of Robotics	Robert Cupec, PhD, Full Professor	**6**	**30**	**0**	**15**	**15**	**E**

Semester: 4							
COURSE	**LEAD INSTRUCTOR(S)**	**ECTS**	**L**	**AE**	**LE**	**DE**	**STATUS**
Practical Training		**10**					**C**
Master's Thesis		**20**					**C**

6.3. Description and general information about each course

General information		
Lead instructor(s)	**Alfonzo Baumgartner, PhD, Associate Professor** **Irena Galić, PhD, Full Professor**	
Course title	3D Computer Graphics	
Study programme	Graduate university study programme in Computer Engineering	
Course status	Elective on module: Data Science, Artificial Intelligence and Robotics, Software Engineering	
Year of study	2.	
ECTS credits and teaching methods	ECTS credits	5
	Number of classes (lectures + auditory exercises + laboratory exercises + design exercises + seminars)	30+(0+30+0)+0

1. COURSE DESCRIPTION

1.1. Course objectives

The applicant gains theoretical and practical knowledge regarding the application of principles in geometric modelling, 3D graphics, and computer animation. Understanding and implementation of matrix representation of geometric transformations and projections in 3D. Application of the programming interfaces OpenGL and BMRT (virtual scene, coordinate systems, camera model, z-buffer, rendering, shading). Implementation and understanding of basic lighting models, transparency, and textures.

1.2. Course enrolment requirements

Requirements for enrolment in the second year of study fulfilled.

1.3. Expected learning outcomes

1. Define and describe the concepts of geometric modelling, 3D graphics, and computer animation.
2. Interpret methods for modelling 3D objects and creating realistic representations.
3. Explain lighting, transparency, texture, and shading models.
4. Interpret basic principles of interpolation and hierarchical structures required for implementing virtual representation processes.
5. Apply mathematical foundations and physics knowledge to computer graphics problems.
6. Connect acquired knowledge and develop an algorithm in the field of computer graphics.

1.4. Course content

The student will become acquainted with the theoretical and practical foundations of applying principles in geometric modelling, 3D graphics, and computer animation. Concepts and techniques for representing three-dimensional objects and achieving realistic renderings will be elaborated. Understanding the basic principles of interpolation, hierarchical structures necessary for implementing rendering processes, lighting, and shading. Practical programming skills in computer graphics will be developed.

1.5. Types of classes	☒ lectures ☐ seminars and workshops ☐ auditory exercises ☒ distance learning ☐ field work	☐ individual exercises ☐ multimedia and network ☒ laboratory exercises ☐ design exercises ☐ working with a supervisor ☐ other _____

1.6. Comments	

1.7. Student obligations

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.8. Monitoring and assessment of student work

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.9. Assessment and evaluation of student work during classes and in the final exam

STUDENT ACTIVITY	ECTS	LEARNING OUTCOME	TEACHING METHOD	ASSESSMENT METHOD	POINTS	
					Min	Max
Attendance at lectures (L), auditory exercises (AE), laboratory exercises (LE), design exercises (DE)	2	1	Lectures (L), laboratory exercises (LE), design exercises (CE)	Attendance tracking. Minimum attendance percentage: 70%.	0	10
Problem-solving exercises	1	5,6	Revision exams (written exam)	Evaluation	10	20
Preparation for laboratory exercises (LE), results analysis, report writing	2	1,2,3,5	Laboratory exercises (LE)	Preparation for LE, LE supervision, LE report assessment	10	20
Preparing for an oral exam and oral exam	1	1,2,3,4,5,6	Oral exam	Evaluation	25	50

1.10. Obligatory literature (at the time of submitting a study programme proposal)

1. I.S. Pandžić: Virtualna okruženja

1.11. Recommended additional literature (at the time of submitting a study programme proposal)

1. J. Neider, T. Davis, M. Woo: OpenGL Programming Guide
2. A. S. Glassner: Principles of Digital Image Synthesis
3. A. S. Glassner: An Introduction to Ray-Tracing
4. A. H. Watt: 3D Computer Graphics
5. P. Shirley, M. Ashikhmin, S. Marschner: Fundamentals of Computer Graphics
6. J. D. Foley, J. F. Huges, A. van Dam, M. McGuire, D. F. Sklar, S. K. Feiner, K. Akeley: Computer Graphics: Principles and Practice

1.12. Number of obligatory literature copies in relation to the number of students currently taking the course

Title	*Number of copies*	*Number of students*
I.S. Pandžić: Virtualna okruženja	4	15

1.13. Quality assurance methods ensuring the acquisition of knowledge, skills and competences

Conducting university questionnaires about teachers (student-teacher relationship, transparency of assessment criteria, motivation for teaching, teaching clarity, etc.). Conducting Faculty surveys of courses

(upon passing the exam, student self-assessment of the adopted learning outcomes and student workload in relation to the number of ECTS credits allocated to activities and courses as a whole).

General information

Lead instructor(s)	**Tomislav Matić, PhD, Associate Professor** **Ivan Aleksi, PhD, Associate Professor**	
Course title	DSP Algorithms and Architecture	
Study programme	Graduate university study programme in Computer Engineering	
Course status	Compulsory on module: Computer Engineering	
Year of study	1.	
ECTS credits and teaching methods	ECTS credits	5
	Number of classes (lectures + auditory exercises + laboratory exercises + design exercises + seminars)	30+(0+30+0)+0

1. COURSE DESCRIPTION		
1.1. Course objectives		
To present students theoretical, simulation and practical knowledge for digital signal processors (DSP), architecture, algorithms and programming.		
1.2. Course enrolment requirements		
Requirements for enrolment in the study program fulfilled.		
1.3. Expected learning outcomes		
1. describe processor requirements digital signal processing 2. compare the properties of different processors for digital signal processing 3. select and apply processor functional units for digital signal processing 4. evaluate the programme execution effectiveness on the processor for digital signal processing 5. apply software tools for simulation and development of software for digital signal processing 6. apply and test the developed software solution on the DSP development system.		
1.4. Course content		
Introduction. Requirements on the processor for digital signal processing algorithms: IIR, FIR, FFT. DSP processor architecture: RISC, DSP, data path. MAC unit, ALU unit, shift circuit, memory organisation, bus architectures, arbitration, addressing methods. Instruction set, data formats, number representation; basic operations, complex arithmetic, convolution, vector arithmetic, parallel data processing. Programming languages C, assembler, algorithms, development tools and DSP programming, real-time operation. DSP applications: sound processing, image processing, computer vision, video encoding and decoding.		
1.5. Types of classes	☒ lectures ☐ seminars and workshops ☐ auditory exercises ☒ distance learning ☐ field work	☒ individual exercises ☐ multimedia and network ☒ laboratory exercises ☐ design exercises ☐ working with a supervisor ☐ other _____
1.6. Comments		
1.7. Student obligations		

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.8. Monitoring and assessment of student work

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.9. Assessment and evaluation of student work during classes and in the final exam

STUDENT ACTIVITY	ECTS	LEARNING OUTCOME	TEACHING METHOD	ASSESSMENT METHOD	POINTS	
					Min	Max
Attendance at lectures (L), auditory exercises (AE), laboratory exercises (LE), design exercises (DE)	1	1, 2, 3, 4	Lectures (L), laboratory exercises (LE), design exercises (CE)	Attendance tracking. Minimum attendance percentage: 70%.	7	10
Problem-solving exercises	2,5	5, 6	Revision exams (written exam)	Evaluation	10	20
Preparation for laboratory exercises (LE), results analysis, report writing	1,5	3, 4, 5, 6	Laboratory exercises (LE)	Preparation for LE, LE supervision, LE report assessment	20	40
Preparing for an oral exam and oral exam	2	1, 2, 3, 4	Oral exam	Evaluation	15	30

1.10. Obligatory literature (at the time of submitting a study programme proposal)

1. Chassaing, R., Reay, D., Digital Signal Processing and Applications with the TMS320C6713 and TMS320C6416 DSK
2. Lapsley, P., Bier, J., Shoham, A., Lee, E. A., DSP Processor Fundamentals, Architectures and Features

1.11. Recommended additional literature (at the time of submitting a study programme proposal)

1. Pirsch, P. Architectures for Digital Signal Processing

2. Mayer-Lindenberg, F., Dedicated Digital Processors, Methods in Hardware/Software System Design; 1st Edition

3. Markovic, D., Brodersen, R. W., DSP Architecture Design Essentials (Electrical Engineering Essentials)

1.12. Number of obligatory literature copies in relation to the number of students currently taking the course

Title	*Number of copies*	*Number of students*
Chassaing, R., Reay, D., Digital Signal Processing and Applications with the TMS320C6713 and TMS320C6416 DSK	1	30
Lapsley, P., Bier, J., Shoham, A., Lee, E. A., DSP Processor Fundamentals, Architectures and Features	1	30

1.13. Quality assurance methods ensuring the acquisition of knowledge, skills and competences

Conducting university questionnaires about teachers (student-teacher relationship, transparency of assessment criteria, motivation for teaching, teaching clarity, etc.). Conducting Faculty surveys of courses (upon passing the exam, student self-assessment of the adopted learning outcomes and student workload in relation to the number of ECTS credits allocated to activities and courses as a whole).

General information		
Lead instructor(s)	**Robert Cupec, PhD, Full Professor with Tenure** **Damir Filko, PhD, Associate Professor**	
Course title	Autonomous Mobile Robots	
Study programme	Graduate university study programme in Computer Engineering	
Course status	Compulsory on module: Artificial Intelligence and Robotics	
Year of study	2.	
ECTS credits and teaching methods	ECTS credits	5
	Number of classes (lectures + auditory exercises + laboratory exercises + design exercises + seminars)	30+(0+30+15)+0

1. COURSE DESCRIPTION		
1.1. Course objectives		
To introduce students to the kinematic model of a mobile robot and how to control the locomotion of a mobile robot. To teach students different types of spatial maps and how to create them. To explain to students how to apply the SLAM algorithm to create an environment map. To introduce students to mobile robot localization methods and teach the students how to apply these methods in practice. To teach basic path planning methods for mobile robots. Introducing students to the methods of controlling unmanned aerial vehicles. Informing students about the basic principles of locomotion of walking and underwater robots and their applications.		
1.2. Course enrolment requirements		
Requirements for enrolment in the second year of study fulfilled.		
1.3. Expected learning outcomes		
1. Create a program for motion control of a mobile robot. 2. Enumerate the types of environment maps and explain how such a map can be created. 3. List methods for mobile robot localisation, explain their operating principle and apply them in practice. 4. Explain the SLAM algorithm and apply it to create an environment map. 5. Create a program for the path planning of a mobile robot. 6. Explain the basics of the control of unmanned aerial vehicles and the locomotion of walking and underwater robots.		
1.4. Course content		
Mobile robot kinematic model. Mobile robot motion control. Building an environment map. Mobile robot localisation. Localisation based on the Kalman filter. Localisation based on the particle filter. Simultaneous Localisation and Map Building (SLAM). Mobile robot path planning. Control of unmanned aerial vehicles. Walking robots. Underwater robot.		
1.5. Types of classes	☒ lectures ☐ seminars and workshops ☐ auditory exercises ☒ distance learning ☐ field work	☐ individual exercises ☐ multimedia and network ☒ laboratory exercises ☒ design exercises ☐ working with a supervisor ☐ other _____
1.6. Comments		

1.7. Student obligations

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.8. Monitoring and assessment of student work

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.9. Assessment and evaluation of student work during classes and in the final exam

STUDENT ACTIVITY	ECTS	LEARNING OUTCOME	TEACHING METHOD	ASSESSMENT METHOD	POINTS	
					Min	Max
Attendance at lectures (L), laboratory exercises (LE), design exercises (DE)	2	1, 2, 3, 4, 5, 6	Lectures (L), laboratory exercises (LE), design exercises (DE)	Attendance tracking. Minimum attendance percentage: 70%.	7	10
Problem-solving exercises set during design exercises (DE)	1	1, 2, 3, 4, 5, 6	Design exercises (DE)	Evaluation	8	30
Preparation for laboratory exercises (LE), results analysis, report writing	1	1,2, 3, 4, 5, 6	Laboratory exercises (LE)	Preparation for LE, LE supervision, LE report assessment	10	30
Preparing for an oral exam and oral exam	1	2, 3, 4, 6	Oral exam	Evaluation	10	30

1.10. Obligatory literature (at the time of submitting a study programme proposal)

1. R. Siegwart, I. Nourbakhsh and D. Scaramuzza: Autonomous Mobile Robots

1.11. Recommended additional literature (at the time of submitting a study programme proposal)

1. J. J. Craig Introduction to Robotics: Mechanics and Control
2. J. C. Latombe: Robot Motion Planning
3. S. Thrun, W. Burgard, D. Fox: Probabilistic Robotics

1.12. Number of obligatory literature copies in relation to the number of students currently taking the course

Title	*Number of copies*	*Number of students*
R. Siegwart, I. Nourbakhsh and D. Scaramuzza: Autonomous Mobile Robots	1	16

1.13. Quality assurance methods ensuring the acquisition of knowledge, skills and competences

Conducting university questionnaires about teachers (student-teacher relationship, transparency of assessment criteria, motivation for teaching, teaching clarity, etc.). Conducting Faculty surveys of courses (upon passing the exam, student self-assessment of the adopted learning outcomes and student workload in relation to the number of ECTS credits allocated to activities and courses as a whole).

General information

Lead instructor(s)	**Mirko Köhler, PhD, Associate Professor**	
Course title	Blockchain Technology and Cryptocurrencies	
Study programme	Graduate university study programme in Computer Engineering	
Course status	Compulsory on module: Data Science Elective on modules: Artificial Intelligence and Robotics and Software Engineering	
Year of study	2.	
ECTS credits and teaching methods	ECTS credits	5
	Number of classes (lectures + auditory exercises + laboratory exercises + design exercises + seminars)	30+(0+30+0)+0

1. COURSE DESCRIPTION

1.1. Course objectives

Provide students with insight into blockchain technology and the principles on which it is based. Introduce them to the benefits of the P2P network and distributed ledger. Explain the different approaches in creating blockchains and show the advantages and disadvantages of their most famous approaches. The hash function will be reviewed. Present to students the differences between blockchain and cryptocurrencies, the differences between public and private chains, and the differences between blockchain technology and directed acyclic graphs. Teach students the basics of various consensus algorithms. Give participants an insight into how the application of blockchain technology will affect the future of the private and public sectors.

1.2. Course enrolment requirements

Requirements for enrolment in the second year of study fulfilled.

1.3. Expected learning outcomes

1. Explain the motives of using blockchain technology.
2. Explain the functionality and the application of the new technology.
3. Analyse existing applications of technology and understand their advantages and disadvantages.
4. Analyse and use existing blockchain records.
5. Create your own blockchain record.
6. Create new software solutions for certain problems by applying the acquired knowledge.

1.4. Course content

1.5. Types of classes	☒ lectures ☐ seminars and workshops ☐ auditory exercises ☒ distance learning ☐ field work	☒ individual exercises ☐ multimedia and network ☒ laboratory exercises ☐ design exercises ☐ working with a supervisor ☒ other group work
1.6. Comments		

1.7. Student obligations

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.8. Monitoring and assessment of student work

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.9. Assessment and evaluation of student work during classes and in the final exam

STUDENT ACTIVITY	ECTS	LEARNING OUTCOME	TEACHING METHOD	ASSESSMENT METHOD	POINTS	
					Min	Max
Attendance at lectures (L), auditory exercises (AE), laboratory exercises (LE), design exercises (DE)	1.5	1,2,3,4,5	Lectures (L), laboratory exercises (LE), design exercises (CE)	Attendance tracking. Minimum attendance percentage: 70%.	0	0
Problem-solving exercises	1.5	1,2,3,4,5,6	Revision exams (written exam)	Evaluation	20	40
Preparation for laboratory exercises (LE), results analysis, report writing	1	4,5,6	Laboratory exercises (LE)	Preparation for LE, LE supervision, LE report assessment	10	20
Preparing for an oral exam and oral exam	1	1,2,3,5,6	Oral exam	Evaluation	20	40

1.10. Obligatory literature (at the time of submitting a study programme proposal)

1. Satoshi Nakamoto; Bitcoin: A Peer-to-Peer Electronic Cash System - white paper https://bitcoin.org/bitcoin.pdf
2. A. M. Antonopoulos; Mastering Bitcoin: Programming the Open Blockchain https://github.com/bitcoinbook/bitcoinbook
3. A. Narayanan, J. Bonneau, E. Felten, A. Miller, S. Goldfeder; Bitcoin and Cryptocurrency Technologies; Princeton University; textbook; 2016. https://d28rh4a8wq0iu5.cloudfront.net/bitcointech/readings/princeton_bitcoin_book.pdf

1.11. Recommended additional literature (at the time of submitting a study programme proposal)

1. Developer Documentation - https://bitcoin.org/en/developer-guide
2. Developer Documentation - https://ethereum.org/en/developers/docs/
3. M. Swan; Blockchain: Blueprint for a New Economy; O'Reilly Media; January 2015

1.12. Number of obligatory literature copies in relation to the number of students currently taking the course

Title	*Number of copies*	*Number of students*
A. M. Antonopoulos; Mastering Bitcoin: Programming the Open Blockchain	1	

1.13. Quality assurance methods ensuring the acquisition of knowledge, skills and competences

Conducting university questionnaires about teachers (student-teacher relationship, transparency of assessment criteria, motivation for teaching, teaching clarity, etc.). Conducting Faculty surveys of courses (upon passing the exam, student self-assessment of the adopted learning outcomes and student workload in relation to the number of ECTS credits allocated to activities and courses as a whole).

General information		
Lead instructor(s)		
Course title	Master's Thesis	
Study programme	Graduate university study programme in Computer Engineering	
Course status	Compulsory	
Year of study	2.	
ECTS credits and teaching methods	ECTS credits	20
	Number of classes (lectures + auditory exercises + laboratory exercises + design exercises + seminars)	-

1. COURSE DESCRIPTION		
1.1. Course objectives		
Assign a task at the appropriate scientific and professional level for a student to demonstrate his/her engineering skills when working on a specific practical problem (measurement, calculation, developing, designing hardware and software, etc.). Supervising the student in doing the task.		
1.2. Course enrolment requirements		
Requirements for enrolment in the second year of study fulfilled.		
1.3. Expected learning outcomes		
1. Analyse a given complex project task in the field of computer engineering and conceptually model its solution. 2. Independently acquire additional knowledge and skills necessary for successfully solving the given complex project task in the field of computer engineering. 3. Plan activities and resources to solve the given complex project task. 4. Identify suitable research or professional methods, techniques and tools for solving the given complex project task relevant to the field of the thesis. 5. Implement one's own solution to the given complex project task. 6. Evaluate the solution of the given complex project task, compare it with known solutions and propose procedures for future work and improvements.		
1.4. Course content		
It depends on the topic of the Master's thesis.		
1.5. Types of classes	☐ lectures ☐ seminars and workshops ☐ auditory exercises ☒ distance learning ☐ field work	☒ individual exercises ☐ multimedia and network ☐ laboratory exercises ☐ design exercises ☒ working with a supervisor ☐ other _____
1.6. Comments		
1.7. Student obligations		

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.8. Monitoring and assessment of student work

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.9. Assessment and evaluation of student work during classes and in the final exam

STUDENT ACTIVITY	ECTS	LEARNING OUTCOME	TEACHING METHOD	ASSESSMENT METHOD	POINTS	
					Min	Max
Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek	-	-	-	-	-	-

1.10. Obligatory literature (at the time of submitting a study programme proposal)

It depends on the topic of the Master's thesis.

1.11. Recommended additional literature (at the time of submitting a study programme proposal)

It depends on the topic of the Master's thesis.

1.12. Number of obligatory literature copies in relation to the number of students currently taking the course

Title	*Number of copies*	*Number of students*
It depends on the topic of the Master's thesis.	-	-

1.13. Quality assurance methods ensuring the acquisition of knowledge, skills and competences

According to the Rulebook on Final and Graduation Exams:
- the topic is approved by the Committee for Final Papers and Master's Theses,
- the graduation exam is conducted before the Graduation Exam Committee.

General information		
Lead instructor(s)	**Tomislav Rudec, PhD, Assistant Professor**	
Course title	Discrete Mathematics	
Study programme	Graduate university study programme in Computer Engineering	
Course status	Compulsory on module: Computer Engineering	
Year of study	2.	
ECTS credits and teaching methods	ECTS credits	5
	Number of classes (lectures + auditory exercises + laboratory exercises + design exercises + seminars)	30+(30+0+0)+0

3. COURSE DESCRIPTION		
1.1. Course objectives		
To teach students the concepts and application of the results of mathematical logic, set theory and number theory. To prepare students for lifelong learning and the use of mathematical structures, relations and operations as tools in application.		
1.2. Course enrolment requirements		
Requirements for enrolment in the second year of study fulfilled.		
1.3. Expected learning outcomes		
1. Create a proof for a given formula in the system of natural deduction. 2. Analyse a given relation from the field of set theory. 3. Analyse large number properties using congruence properties and Euler's and Fermat's little theorem. 4. Construct a mathematical procedure for solving nonlinear and linear Diophantine equations. 5. Create and analyse a computer algorithm for solving NP-hard problems in the field of game theory and discrete mathematics.		
1.4. Course content		
Mathematical logic. Introduction to logic. Propositional logic. The alphabet of propositional logic. Semantics and syntax. Operators. Propositional logic equations. Natural deduction. Basics of set theory. Set operations. Venn diagrams. Binary relations. Equivalence relations. Partition of the set. Order relations. Basics of number theory. Whole numbers. Divisibility and prime numbers. Congruences. Euler's function. Euler's theorem and Fermat's little theorem. Diophantine equations.		
1.5. Types of classes	☒ lectures ☐ seminars and workshops ☒ auditory exercises ☒ distance learning ☐ field work	☒ individual exercises ☐ multimedia and network ☐ laboratory exercises ☐ design exercises ☐ working with a supervisor ☐ other _____
1.6. Comments		
1.7. Student obligations		
Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.		

1.8. Monitoring and assessment of student work

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.9. Assessment and evaluation of student work during classes and in the final exam

STUDENT ACTIVITY	ECTS	LEARNING OUTCOME	TEACHING METHOD	ASSESSMENT METHOD	POINTS	
					Min	Max
Attendance at lectures (L), auditory exercises (AE), laboratory exercises (LE), design exercises (DE)	2	1, 2, 3, 4, 5	Lectures (L), laboratory exercises (LE), design exercises (CE)	Attendance tracking. Minimum attendance percentage: 70%.	0	0
Problem-solving exercises	1	1, 2, 3, 4, 5	Revision exams (written exam)	Evaluation	20	50
Preparing for an oral exam and oral exam	2	2, 3, 5	Oral exam	Evaluation	20	50

1.10. Obligatory literature (at the time of submitting a study programme proposal)

1. T. Rudec, A. Šteko. 30+30 Diskretna matematika, Sveučilište Josip Juraj Strossmayer u Osijeku, Fakultet Elektrotehnike, računarstva i informacijskih tehnologija, Osijek, https://www.ferit.unios.hr/fakultet/knjiznica-i-izdavacka-djelatnost
2. A. Dujella. Number theory, Školska knjiga, Zagreb 2021.

1.11. Recommended additional literature (at the time of submitting a study programme proposal)

1. D. Žubrinić. Diskretna matematika Zagreb:Element, 2002
2. I. Anderson. A First Course in Discrete Mathematics Springer Verlag, 2001.

1.12. Number of obligatory literature copies in relation to the number of students currently taking the course

Title	*Number of copies*	*Number of students*
T. Rudec, A. Šteko. 30+30 Diskretna matematika, Sveučilište Josip Juraj Strossmayer u Osijeku, Fakultet Elektrotehnike, računarstva i informacijskih tehnologija, Osijek, https://www.ferit.unios.hr/fakultet/knjiznica-i-izdavacka-djelatnost	The book is available online and it is free of charge	30

1.13. Quality assurance methods ensuring the acquisition of knowledge, skills and competences

Conducting university questionnaires about teachers (student-teacher relationship, transparency of assessment criteria, motivation for teaching, teaching clarity, etc.). Conducting Faculty surveys of courses (upon passing the exam, student self-assessment of the adopted learning outcomes and student workload in relation to the number of ECTS credits allocated to activities and courses as a whole).

General information

Lead instructor(s)	**Ivan Vidović, PhD, Assistant Professor**	
Course title	Computer Systems Design	
Study programme	Graduate university study programme in Computer Engineering	
Course status	Compulsory on module: Computer Engineering Elective on modules: Artificial Intelligence and Robotics, Software Engineering and Data Science	
Year of study	1. on module Computer Engineering 2. on modules Artificial Intelligence and Robotics, Software Engineering, Data Science	
ECTS credits and teaching methods	ECTS credits	6
	Number of classes (lectures + auditory exercises + laboratory exercises + design exercises + seminars)	45+(0+30+0)+0

1. COURSE DESCRIPTION

1.1. Course objectives

Present students with theoretical and practical knowledge in the field of computer design, microprocessors, and computer and microprocessor systems. Teach students to recognise specific problems in the field of design of computer and microprocessor systems. Acquire skills in the application of tools for circuit design, design of computer systems, and development of software support, as well as for simulation and testing of the operation of the designed system. Implement designed computer systems on FPGA integrated circuits and develop appropriate software support.

1.2. Course enrolment requirements

Requirements for enrolment in the first year of study fulfilled (module: Computer Engineering).
Requirements for enrolment in the second year of study fulfilled (modules: Artificial Intelligence and Robotics, Software Engineering, Data Science).

1.3. Expected learning outcomes

1. Specify the functionality of computer systems.
2. Critically evaluate parts of the computer system.
3. Test the operation of various simple and complex computer systems and critically judge the obtained results.
4. Specify and design simple processor systems with peripheral units.
5. Apply and test simple processor systems on technical processes.
6. Evaluate and assess the developed and applied processor systems.

1.4. Course content

Computer organisation. Processor and microprocessor. RISC and CISC processor architectures. Multiprocessor systems. Types and purposes of buses. Single-bus and multi-bus systems. Types and purposes of memories. Memory access methods. Languages for hardware description (VHDL/Verilog). Design of a simple arithmetic logic unit. Design of various peripheral units. Microprocessor design. Microprocessor design. Computer systems on FPGA integrated circuit. Microprocessors on FPGA integrated circuit. Input-output units. Timers and counters. Interrupt controllers. Development of software support for computer systems on FPGA integrated circuit. Testing, debugging, validation and verification of the designed computer system and program support. Implementation of advanced computer algorithms on FPGA integrated circuit.

1.5. Types of classes	☒ lectures ☐ seminars and workshops ☐ auditory exercises	☐ individual exercises ☐ multimedia and network ☒ laboratory exercises ☐ design exercises

	☒ distance learning ☐ field work	☐ working with a supervisor ☐ other _____
1.6. Comments		

1.7. Student obligations

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.8. Monitoring and assessment of student work

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.9. Assessment and evaluation of student work during classes and in the final exam

STUDENT ACTIVITY	ECTS	LEARNING OUTCOME	TEACHING METHOD	ASSESSMENT METHOD	POINTS	
					Min	Max
Attendance at lectures (L), auditory exercises (AE), laboratory exercises (LE), design exercises (DE)	1	1,2,3,4,5,6	Lectures (L), laboratory exercises (LE), design exercises (CE)	Attendance tracking. Minimum attendance percentage: 70%.	1	2
Problem-solving exercises	2.5	1,2,3,4,5,6	Revision exams (written exam)	Evaluation	25	50
Preparation for laboratory exercises (LE), results analysis, report writing	1.5	1,2,3,4,5,6	Laboratory exercises (LE)	Preparation for LE, LE supervision, LE report assessment	0	20
Preparing for an oral exam and oral exam	1	1, 2, 3, 4, 5, 6	Oral exam	Evaluation	14	28

1.10. Obligatory literature (at the time of submitting a study programme proposal)

1. J. Ledin: Modern Computer Architecture and Organization: Learn x86, ARM, and RISC-V architectures and the design of smartphones, PCs, and cloud servers, 2nd edition, Packt Publishing, 2022.

1.11. Recommended additional literature (at the time of submitting a study programme proposal)

1. N. Nisam and S. Schocken: The Elements of Computing Systems, second edition: Building a Modern Computer from First Principles, 2nd Edition, The MIT Press, 2021

2. J. Ledin: Architecting High-Performance Embedded Systems: Design and build high-performance real-time digital systems based on FPGAs and custom circuits, Packt Publishing, 2021

3. U. Meyer-Baese: Embedded Microprocessor System Design using FPGAs, 1st ed., Springer, 2022.

4. Volnei A. Pedroni, Circuit Design and Simulation with VHDL, Second Edition London, 2010.

1.12. Number of obligatory literature copies in relation to the number of students currently taking the course

Title	*Number of copies*	*Number of students*

J. Ledin: Modern Computer Architecture and Organization: Learn x86, ARM, and RISC-V architectures and the design of smartphones, PCs, and cloud servers, 2nd edition, Packt Publishing, 2022.		30
1.13. Quality assurance methods ensuring the acquisition of knowledge, skills and competences		
Conducting university questionnaires about teachers (student-teacher relationship, transparency of assessment criteria, motivation for teaching, teaching clarity, etc.). Conducting Faculty surveys of courses (upon passing the exam, student self-assessment of the adopted learning outcomes and student workload in relation to the number of ECTS credits allocated to activities and courses as a whole).		

General information		
Lead instructor(s)	**Emmanuel Karlo Nyarko, PhD, Associate Professor** **Ratko Grbić, PhD, Associate Professor**	
Course title	Deep Learning	
Study programme	Graduate university study programme in Computer Engineering	
Course status	Compulsory on module: Artificial Intelligence and Robotics Elective on module: Data Science	
Year of study	2.	
ECTS credits and teaching methods	ECTS credits	6
	Number of classes (lectures + auditory exercises + laboratory exercises + design exercises + seminars)	(45+0+30+0+0)

1. COURSE DESCRIPTION		
1.1. Course objectives		
Introduction to the basics of deep learning, popular architectures of deep neural networks and ways of training deep models. Application of deep models in solving problems from the field of supervised, unsupervised, and reinforcement learning in signal and information processing: image processing, computer vision, natural language processing and software agent training. Acquisition of relevant skills using modern libraries and application frameworks for deep learning .		
1.2. Course enrolment requirements		
Requirements for enrolment in the second year of study fulfilled.		
1.3. Expected learning outcomes		
1. Explain the advantages of deep learning compared to other machine learning approaches. 2. Evaluate the suitability of a deep learning model for a given problem. 3. Analyse and evaluate the performance of a given deep learning model. 4. Use modern application frameworks for the development of deep learning models. 5. Apply deep learning techniques to solve problems in the fields of supervised and unsupervised learning. 6. Apply deep learning techniques to solve problems in the field of reinforcement learning.		
1.4. Course content		
Basic terminology and introduction to deep learning. Convolutional neural networks. Siamese neural networks. Autoencoders. Generative adversarial networks. Transformers. Recurrent neural networks. Self-supervised learning. Deep reinforcement learning. Deep learning model selection and evaluation. Various deep learning applications and examples.		
1.5. Types of classes	☒ lectures ☐ seminars and workshops ☐ auditory exercises ☒ distance learning ☐ field work	☒ individual exercises ☐ multimedia and network ☒ laboratory exercises ☐ design exercises ☐ working with a supervisor ☐ other _____
1.6. Comments		
1.7. Student obligations		

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.8. Monitoring and assessment of student work

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.9. Assessment and evaluation of student work during classes and in the final exam

STUDENT ACTIVITY	ECTS	LEARNING OUTCOME	TEACHING METHOD	ASSESSMENT METHOD	POINTS	
					Min	Max
Attendance at lectures (L), laboratory exercises (LE)	2	1, 2, 3, 4, 5, 6	Lectures (L), laboratory exercises (LE)	Attendance tracking. Minimum attendance percentage: 70%.	7	10
Preparation for laboratory exercises (LE), results analysis, report writing	1.5	3, 4, 5, 6	Laboratory exercises (LE)	Preparation for LE, LE supervision, LE report assessment	15	30
Solving project task	1	3, 4, 5, 6	Individual exercises	Verification of the solution of the project task, presentation of the solution	0	25
Preparing for oral exam and oral exam	1.5	1, 2, 3, 4, 5, 6	Oral exam	Evaluation	18	35

1.10. Obligatory literature (at the time of submitting a study programme proposal)

1. S. Raschka, Y. (Hayden) Liu, V. Mirjalili, Machine Learning with PyTorch and Scikit-Learn: Develop machine learning and deep learning models with Python, 1st Edition, Packt Publishing, 2022.

1.11. Recommended additional literature (at the time of submitting a study programme proposal)

1. R. Atienza, Advanced Deep Learning with TensorFlow 2 and Keras: Apply DL, GANs, VAEs, deep RL, unsupervised learning, object detection and segmentation, and more, 2nd Edition, Packt Publishing, 2020.

1.12. Number of obligatory literature copies in relation to the number of students currently taking the course

Title	*Number of copies*	*Number of students*
S. Raschka, Y. (Hayden) Liu, V. Mirjalili, Machine Learning with PyTorch and Scikit-Learn: Develop machine learning and deep learning models with Python, 1st Edition, Packt Publishing, 2022.	0	
R. Atienza, Advanced Deep Learning with TensorFlow 2 and Keras: Apply DL, GANs, VAEs, deep RL, unsupervised learning, object detection and segmentation, and more, 2nd Edition, Packt Publishing, 2020.	0	

1.13. Quality assurance methods ensuring the acquisition of knowledge, skills and competences

Conducting university questionnaires about teachers (student-teacher relationship, transparency of assessment criteria, motivation for teaching, teaching clarity, etc.). Conducting Faculty surveys of courses (upon passing the exam, student self-assessment of the adopted learning outcomes and student workload in relation to the number of ECTS credits allocated to activities and courses as a whole).

General information

Lecturer(s)	**Dražen Slišković, PhD, Full professor**	
Course title	Industrial Informatics	
Study programme	Graduate university study programme in Computer Engineering	
Course status	Compulsory on module: Artificial Intelligence and Robotics	
Year of study	2.	
ECTS credits and teaching methods	ECTS credits	5
	Number of classes (lectures + auditory exercises + laboratory exercises + design exercises + seminars)	30+0+30+0+0

1. COURSE DESCRIPTION

1.1. Course objectives

To acquaint students with the tasks of managing and control a complex production process, and how to realize a highly computerized system for automation, from the level of connection with the technical process, through the control system and the process monitoring system, to the levels of production and business management. To acquaint students with the basics of the application of PLC, SCADA system and industrial communication system, which is the basis for the practical implementation of systems for automatic control of various industrial processes. To acquaint students with informatization/computerization in these systems and the most important software systems and tools.

1.2. Course enrolment requirements

Requirements for enrolment in the second year of study fulfilled.

1.3. Expected learning outcomes

1. describe the ways of managing a complex technical (production) process and explain what informatization and automation are in process management and control
2. describe the structure and mode of operation of a process computer and its implementation in the form of a programmable logic controller
3. choose a PLC configuration and write a control/user program for simpler and more complex tasks
4. explain the advantages and disadvantages of (de)centralization in the realization of a system for automatic process control
5. describe the role and structure of SCADA software support, and its main interfaces
6. define the requirements for the communication system at individual levels of managing and control and choose appropriate communication for a specific purpose
7. establish communication, with several communication standards, using Simatic equipment

1.4. Course content

Production system and industrial plant. Process management and control and stratification of related tasks. Informatization and automation of the production system. The basic structure of the system for automatic process control. Examples from practice. Digital realization of the control system. Process computer and programmable logic controller (PLC). Process peripherals - sensors and actuators and connection to the process computer. Control unit - the central unit of the system for automatic process management and control. Control unit structures: centralised and decentralized, hierarchical and distributed. Supervisory unit - subsystem for operator-production system interfacing and process database. Structures of the supervisory unit and ways of managing the modern automated system. Supervisory control. Equipment for the implementation of the control unit and supervisory unit. SCADA system. Computerization of production and business management - MES and ERP systems. Industry 4.0. Computer/digital communication systems for application in industry. General purpose communication technologies/standards. Bus and network communication. Industrial communication standards. Communication at the field level and at higher

(management) levels. ProfiBus, MPI, CAN, ASI, Industrial Ethernet, ProfiNet. Software support in automation systems and user software tools. Examples of complete systems; for the control and automation of production systems and for monitoring the automated production system.

1.5. Types of classes	☒ lectures ☐ seminars and workshops ☐ auditory exercises ☒ distance learning ☐ field work	☐ individual exercises ☐ multimedia and network ☒ laboratory exercises ☐ design exercises ☐ working with a supervisor ☐ other _____
1.6. Comments		

1.7. Student obligations

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.8. Monitoring and assessment of student work

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.9. Assessment and evaluation of student work during classes and in the final exam

STUDENT ACTIVITY	ECTS	LEARNING OUTCOME	TEACHING METHOD	ASSESSMENT METHOD	POINTS	
					Min	Max
Attendance at lectures (L), laboratory exercises (LE)	2	1, 2, 3, 4, 5, 6,7	Lectures (L), laboratory exercises (LE)	Attendance tracking. Minimum attendance percentage: 70%.	2	5
Preparation for laboratory exercises (LE), results analysis, report writing	2	3, 6, 7	Laboratory exercises (LE)	Checking of preparation for LE, LE supervision, LE report assessment	20	40
Preparing for an oral exam and oral exam	1	1, 2, 4, 5, 6	Oral exam	Evaluation	28	55

1.10. Obligatory literature (at the time of submitting a study programme proposal)

1. Slišković, D. Procesna automatizacija – predavanja, zavodska skripta

1.11. Recommended additional literature (at the time of submitting a study programme proposal)

1. Perić, N. Automatizacija postrojenja i procesa - predavanja
2. Crispin, A. J. Programmable Logic Controllers and their Engineering Applications
3. Jović, F. Kompjutersko vođenje procesa

1.12. Number of obligatory literature copies in relation to the number of students currently taking the course

Title	*Number of copies*	*Number of students*

1.13. Quality assurance methods ensuring the acquisition of knowledge, skills and competences		
Conducting university questionnaires about teachers (student-teacher relationship, transparency of assessment criteria, motivation for teaching, teaching clarity, etc.). Conducting Faculty surveys of courses (upon passing the exam, student self-assessment of the adopted learning outcomes and student workload in relation to the number of ECTS credits allocated to activities and courses as a whole).		

General information		
Lead instructor(s)	**Tomislav Keser, PhD, Associate Professor**	
Course title	Integration of Digital Systems	
Study programme	Graduate university study programme in Computer Engineering	
Course status	Compulsory on module: Computer Engineering	
Year of study	1.	
ECTS credits and teaching methods	ECTS credits	6
	Number of classes (lectures + auditory exercises + laboratory exercises + design exercises + seminars)	30+0+15+30+0

1. COURSE DESCRIPTION		
1.1. Course objectives		
Introduce and train students in the procedures and tasks of designing, building, integrating and testing the functionality of complex digital and computer systems. Demonstrate and become familiar with the basic principles of designing complex digital systems through the use of development and simulation tools for integration. Become familiar with the design principles of printed circuit board designs and procedures for creating them. Familiarize themselves with digital system design standards. Demonstrate and learn the principles of good practice in the design of printed circuits, taking into account electromagnetic compatibility rules and circuit layout. Apply recommendations and standards in the design and integration of digital systems.		
1.2. Course enrolment requirements		
Requirements for enrolment in the second year of study fulfilled.		
1.3. Expected learning outcomes		
1. Explain the basic concepts and recognize the development stages in the integration of complex digital systems. 2. Distinguish needs and design specifics between flat and 3D structured circuits. 3. Recognise and model the requirements for the correct modelling of planar structured circuits. 4. Design flat-structured, more complex circuits in accordance with EMC instructions and rules. 5. Apply design tools and knowledge of EMF challenges in correctly structuring planar circuits. 6. Create and integrate a more complex, flat dimensioned circuit into a functional unit. 7. Apply recommendations and standards in the design and integration of digital systems.		
1.4. Course content		
Basic concepts and tasks of the design and development of complex digital systems. Basic concepts and tasks of integration and functional testing of digital and computer systems. Design and functional simulation of structural schemes using development-oriented CAD design, simulation and PCB tools. Design of flat dimensioned complex circuits. Fundamentals of flat-structured design. Elements of electromagnetic compatibility in the design of flat structures. Good practice rules when using design and simulation tools to integrate complex digital and computer systems.		
1.5. Types of classes	☒ lectures ☐ seminars and workshops ☐ auditory exercises ☒ distance learning ☐ field work	☐ individual exercises ☐ multimedia and network ☒ laboratory exercises ☒ design exercises ☐ working with a supervisor

		☐ other _____
1.6. Comments		

1.7. Student obligations

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.8. Monitoring and assessment of student work

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.9. Assessment and evaluation of student work during classes and in the final exam

STUDENT ACTIVITY	ECTS	LEARNING OUTCOME	TEACHING METHOD	ASSESSMENT METHOD	POINTS	
					Min	Max
Attendance at lectures (L), auditory exercises (AE), laboratory exercises (LE), design exercises (DE)	2	1-6	Lectures (L), laboratory exercises (LE), design exercises (CE)	Attendance tracking. Minimum attendance percentage: 70%.	7	15
Problem-solving exercises	0	-	Revision exams (written exam)	Evaluation	0	0
Preparation for laboratory exercises (LE), results analysis, report writing	1	2-4	Laboratory exercises (LE)	Preparation for LE, LE supervision, LE report assessment	8	20
Problem-solving in design exercises	2	3-6	Design exercises (DE)	DE problem solutions assessment	25	40
Preparing for an oral exam and oral exam	1	1-7	Oral exam	Evaluation	10	25

1.10. Obligatory literature (at the time of submitting a study programme proposal)

1. Christopher T. Robertson, Printed Circuit Board Designer's Reference: Basics, Prentice Hall Professional, 2004.
2. Eric Bogatin, Signal and Power Integrity, Simplified, Prentice Hall, 2018.
3. David A. Weston, Electromagnetic Compatibility: Methods, Analysis, Circuits, and Measurement, CRC Press 2016.

1.11. Recommended additional literature (at the time of submitting a study programme proposal)

1. Bruce R. Archambeault, James Drewniak, PCB Design for Real-World EMI Control, Springer Science & Business Media, 2002.
2. Douglas Brooks, Signal Integrity Issues and Printed Circuit Board Design, Prentice Hall Professional, 2003.
3. Leonard Marks, James Caterina, Printed Circuit Assembly Design, McGraw Hill Professional, 2000.
4. Hanqiao Zhang, Steven Krooswyk, Jeffrey Ou, High Speed Digital Design: Design of High-Speed Interconnects and Signaling, Elsevier Science, 2015.

1.12. Number of obligatory literature copies in relation to the number of students currently taking the course

Title	*Number of copies*	*Number of students*
Christopher T. Robertson, Printed Circuit Board Designer's Reference: Basics, Prentice Hall Professional, 2004.	0	15
Eric Bogatin, Signal and Power Integrity, Simplified, Prentice Hall, 2018.	0	15
David A. Weston, Electromagnetic Compatibility: Methods, Analysis, Circuits, and Measurement, CRC Press 2016.	0	15
1.13. Quality assurance methods ensuring the acquisition of knowledge, skills and competences		
Conducting university questionnaires about teachers (student-teacher relationship, transparency of assessment criteria, motivation for teaching, teaching clarity, etc.). Conducting Faculty surveys of courses (upon passing the exam, student self-assessment of the adopted learning outcomes and student workload in relation to the number of ECTS credits allocated to activities and courses as a whole).		

General information		
Lead instructor(s)	Josip Balen, PhD, Associate Professor	
Course title	Intelligent Transportation Systems	
Study programme	Graduate university study programme in Computer Engineering	
Course status	Elective on module: Software Engineering	
Year of study	2.	
ECTS credits and teaching methods	ECTS credits	5
	Number of classes (lectures + auditory exercises + laboratory exercises + design exercises + seminars)	30+0+30+0+0

1. COURSE DESCRIPTION

1.1. Course objectives

Introduce the wider field of intelligent transport systems such as cooperative, connected and automated mobility in road, rail, air and sea transport, as well as autonomous vehicles and robots in indoor spaces. Introduce the usage of modern wireless, electronic and automated technologies and devices, data processing, control and communication to improve safety, mobility and traffic efficiency. Testing applications for processing and sharing information in intelligent transport systems.

1.2. Course enrolment requirements

Requirements for enrolment in the first year of study fulfilled.

1.3. Expected learning outcomes

1. Explain the basic concepts, methods and challenges in intelligent transport systems
2. Analyse new technologies and devices built into vehicles and robots
3. Create a prototype and perform validation of a complex software product with an emphasis on the field of intelligent transportation systems
4. Select and evaluate a suitable algorithm for a known (already solved) problem
5. Design your own algorithm for an unknown problem and compare it with other algorithms
6. Document your own solution to a given complex project task through the creation of a technical report, scientific paper or presentation materials

1.4. Course content

Introduction to basic principles and challenges in intelligent transport systems. Intelligent roads, industrial spaces and transport infrastructure. Overview of new technologies embedded in vehicles and robots (architecture, embedded systems, operating systems, communication devices). Autonomous vehicles without drivers in outdoor and indoor spaces. Efficient information sharing in wireless vehicles networks (applications and concepts). Security of communication, vehicles and people. Algorithms and protocols for efficient information dissemination between vehicles. Simulations of traffic and communication between vehicles and infrastructure. Testing applications for processing and sharing information in intelligent transport systems. Processing of obtained results and evaluation of performance.

1.5. Types of classes	☒ lectures ☐ seminars and workshops ☐ auditory exercises	☒ individual exercises ☐ multimedia and network ☒ laboratory exercises ☐ design exercises

	☒ distance learning ☐ field work	☒ working with a supervisor ☐ other _____
1.6. Comments		
1.7. Student obligations		
Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.		
1.8. Monitoring and assessment of student work		
Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.		
1.9. Assessment and evaluation of student work during classes and in the final exam		

STUDENT ACTIVITY	ECTS	LEARNING OUTCOME	TEACHING METHOD	ASSESSMENT METHOD	POINTS	
					Min	Max
Attendance at lectures (L), auditory exercises (AE), laboratory exercises (LE), design exercises (DE)	0,5	1,2,3,4,5,6	Lectures (L), Laboratory exercises (LE)	Attendance tracking. Minimum attendance percentage: 70%.	7	10
Preparation for laboratory exercises (LE), results analysis, report writing	1	3,4,5,6	Laboratory exercises (LE)	Preparation for LE, LE supervision, LE report assessment	6	15
Preparing for an oral exam and oral exam	1,5	1,2,3,4,5,6	Oral exam	Evaluation	12	25
Work on a project assignment	2	3,4,5,6	Group project work	Evaluation	25	50

1.10. Obligatory literature (at the time of submitting a study programme proposal)

1. Sundaravalli Narayanaswami, Intelligent Transportation Systems, Cambridge Scholars Publishing, 2023
2. Amina Adadi , Afaf Bouhoute, Explainable Artificial Intelligence for Intelligent Transportation Systems, CRC Press, 2023
3. Madison West, Intelligent Transportation Systems, States Academic Press, 2022
4. Mashrur Chowdhury, Amy Apon, Kakan Dey, Data Analytics for Intelligent Transportation Systems, Elsevier, 2017

1.11. Recommended additional literature (at the time of submitting a study programme proposal)

1. Loveleen Gaur, Biswa Mohan Sahoo, Explainable Artificial Intelligence for Intelligent Transportation Systems: Ethics and Applications, Springer, 2022
2. Sonali P. Botkar, Sachin P. Godse, Parikshit N. Mahalle, Gitanjali R. Shinde, VANET Challenges and Opportunities, CRC Press, 2021.

1.12. Number of obligatory literature copies in relation to the number of students currently taking the course

Title	*Number of copies*	*Number of students*

1.13. Quality assurance methods ensuring the acquisition of knowledge, skills and competences

Conducting university questionnaires about teachers (student-teacher relationship, transparency of assessment criteria, motivation for teaching, teaching clarity, etc.). Conducting Faculty surveys of courses (upon passing the exam, student self-assessment of the adopted learning outcomes and student workload in relation to the number of ECTS credits allocated to activities and courses as a whole).

General information

Lecturer(s)	**Josip Job, PhD, Full Professor, Ratko Grbić, PhD, Associate Professor**	
Course title	Internet of Things	
Study programme	Graduate university study programme in Computer Engineering	
Course status	Compulsory on module: Data Science Elective on modules: Computer Engineering, Artificial Intelligence and Robotics	
Year of study	2.	
ECTS credits and teaching methods	ECTS credits	5
	Number of classes (lectures + auditory exercises + laboratory exercises + design exercises + seminars)	30 + (0 + 15 + 15) + 0

1. COURSE DESCRIPTION

1.1. Course objectives

To familiarise students with basic theoretical knowledge and practical skills in the field of the Internet of Things and to prepare them for both independent and collaborative project work involving data collection, storage, processing, and visualisation, aligned with the Internet of Things paradigm.

1.2. Course enrolment requirements

Requirements for enrolment in the second year of study fulfilled.

1.3. Expected learning outcomes

1. Assess and justify the suitability of the elements of the given IoT system.
2. Assess the suitability of tools for the development of the microcontroller program code in a specific project.
3. Create your own software solution based on several suitable libraries for the use of sensors in the microcontroller based system.
4. Propose the design of an Internet of Things system for a given simple problem.
5. Integrate software support and circuitry into the functional system of the Internet of Things.
6. Design the architecture of software solutions for Internet of Things systems.

1.4. Course content

Introduction to the Internet of Things (IoT). IoT technologies (elements, circuits, communication, platforms and development environments). IoT architecture and infrastructure. Circuit-based objects. Data acquisition and storage(methods, protocols, applications and services). Data Access. User interfaces and Presentation of Data. Security in IoT systems. Application of the Internet of Things: industry, meteorology, agriculture, medicine, smart houses, smart cities.

1.5. Types of classes	☒ lectures ☐ seminars and workshops ☐ auditory exercises ☒ distance learning ☐ field work	☐ individual exercises ☐ multimedia and network ☒ laboratory exercises ☒ design exercises ☐ working with a supervisor ☐ other _____
1.6. Comments		

1.7. Student obligations

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.8. Monitoring and assessment of student work

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.9. Assessment and evaluation of student work during classes and in the final exam

STUDENT ACTIVITY	ECTS	LEARNING OUTCOME	TEACHING METHOD	ASSESSMENT METHOD	POINTS	
					Min	Max
Attendance at lectures (L), auditory exercises (AE), laboratory exercises (LE), design exercises (DE)	1,4	1, 2, 3, 4, 5, 6	Lectures (L), laboratory exercises (LE), design exercises (CE)	Attendance tracking. Minimum attendance percentage: 70%.	0	0
Preparation for laboratory exercises (LE), results analysis, report writing	0,6	1, 2, 3, 4, 5, 6	Laboratory exercises (LE)	Preparation for LE, LE supervision, LE report assessment	10	20
Problem-solving exercises	1	1, 2, 3, 4, 5, 6	Design exercises (DE)	Project evaluation	20	30
Preparing for an oral exam and oral exam	2	1, 2, 3, 4, 5, 6	Oral exam	Evaluation	20	50

1.10. Obligatory literature (at the time of submitting a study programme proposal)

1. Gary Smart, Practical Python Programming for IoT: Build advanced IoT projects using a Raspberry Pi 4, MQTT, RESTful APIs, WebSockets, and Python 3, Packt Publishing, 2020.

2. Bahga, A; Madisetti V. Internet of Things: A Hands-on-Approach, Arshdeep Bahga & Vijay Madisetti, 2014.

1.11. Recommended additional literature (at the time of submitting a study programme proposal)

1. Dieter Uckelmann, Mark Harrison, Florian Michahelles, Architecting the Internet of Things, Springer, 2011.
2. Charalampos Doukas, Building Internet of Things with the Arduino: 1, CreateSpace Independent Publishing Platform, 2012.

1.12. Number of obligatory literature copies in relation to the number of students currently taking the course

Title	*Number of copies*	*Number of students*
1. Gary Smart, Practical Python Programming for IoT: Build advanced IoT projects using a Raspberry Pi 4, MQTT, RESTful APIs, WebSockets, and Python 3, Packt Publishing, 2020.	0	36
2. Bahga, A; Madisetti V. Internet of Things: A Hands-on-Approach, Arshdeep Bahga & Vijay Madisetti, 2014.	1	36

1.13. Quality assurance methods ensuring the acquisition of knowledge, skills and competences

Conducting university questionnaires about teachers (student-teacher relationship, transparency of assessment criteria, motivation for teaching, teaching clarity, etc.). Conducting Faculty surveys of courses (upon passing the exam, student self-assessment of the adopted learning outcomes and student workload in

relation to the number of ECTS credits allocated to activities and courses as a whole).

General information		
Lead instructor(s)	**Krešimir Nenadić, PhD, Full Professor**	
Course title	iOS Programming	
Study programme	Graduate university study programme in Computer Engineering	
Course status	Elective on modules: Software Engineering and Data Science	
Year of study	2.	
ECTS credits and teaching methods	ECTS credits	5
	Number of classes (lectures + auditory exercises + laboratory exercises + design exercises + seminars)	30+(0+15+15)+0

1. COURSE DESCRIPTION		
1.1. Course objectives		
The aim of the course is to present tools, technologies and modern concepts for creating applications for the iOS platform		
1.2. Course enrolment requirements		
Requirements for the enrolment in the second year of study fulfilled.		
1.3. Expected learning outcomes		
1. Identify programming concepts specific to creating applications for mobile devices. 2. Use the platform to create applications for mobile devices. 3. Create a complex mobile application and programmatically implement the designed interface 4. Conduct structural and functional testing of the application on real mobile devices 5. Create application source code documentation 6. Recommend alternative approaches to solving a specific problem observed during testing		
1.4. Course content		
Getting to know the specifics of application development for the iOS platform, getting to know the tools for application development for the iOS platform (Xcode), iOS application life cycle, introduction to the programming language Swift (SwiftUI), Swift language elements, data types, functions, complex types, creation of simple ones examples and testing on emulator and real device.		
1.5. Types of classes	☒ lectures ☐ seminars and workshops ☐ auditory exercises ☒ distance learning ☐ field work	☐ individual exercises ☐ multimedia and network ☒ laboratory exercises ☒ design exercises ☐ working with a supervisor ☐ other _____
1.6. Comments		
1.7. Student obligations		
Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.		

1.8. Monitoring and assessment of student work

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.9. Assessment and evaluation of student work during classes and in the final exam

STUDENT ACTIVITY	ECTS	LEARNING OUTCOME	TEACHING METHOD	ASSESSMENT METHOD	POINTS	
					Min	Max
Attendance at lectures (L), auditory exercises (AE), laboratory exercises (LE), design exercises (DE)	1	1, 2	Lectures (L), laboratory exercises (LE), design exercises (CE)	Attendance tracking. Minimum attendance percentage: 70%.	0	20
Problem-solving exercises	1	3, 4	Revision exams (written exam)	Evaluation	15	20
Preparation for laboratory exercises (LE), results analysis, report writing	1	3, 4, 5, 6	Laboratory exercises (LE)	Preparation for LE, LE supervision, LE report assessment	20	30
Preparing for an oral exam and oral exam	2	1, 2, 3, 4, 5, 6	Oral exam	Evaluation	15	30

1.10. Obligatory literature (at the time of submitting a study programme proposal)

1. Keur, Christian, Hillegass, Aaron, iOS Programming: The Big Nerd Ranch (6th Ed.)
2. Sahar, Ahmad, iOS 14 Programming for Beginners: Get started with building iOS apps with Swift 5.3 and Xcode 12 (5th Ed.)

1.11. Recommended additional literature (at the time of submitting a study programme proposal)

1. Eguiluz Alebico, Mario; Baker, Chris; Wals, Donny; Mastering iOS 14 Programming: Build professional-grade iOS 14 applications with Swift 5.3 and Xcode 12.4 (4th Ed.)

2. Moon, Keith; Barker, Chris; Swift Cookbook (2nd Ed.)

1.12. Number of obligatory literature copies in relation to the number of students currently taking the course

Title	*Number of copies*	*Number of students*

1.13. Quality assurance methods ensuring the acquisition of knowledge, skills and competences

Conducting university questionnaires about teachers (student-teacher relationship, transparency of assessment criteria, motivation for teaching, teaching clarity, etc.). Conducting Faculty surveys of courses (upon passing the exam, student self-assessment of the adopted learning outcomes and student workload in relation to the number of ECTS credits allocated to activities and courses as a whole).

General information		
Lead instructor(s)	**Irena Galić, PhD, Full Professor**	
Course title	Research in Data Science	
Study programme	Graduate university study programme in Computer Engineering	
Course status	Compulsory on module: Data Science	
Year of study	2.	
ECTS credits and teaching methods	ECTS credits	5
	Number of classes (lectures + auditory exercises + laboratory exercises + design exercises + seminars)	30+(0+30+0)+0

1. COURSE DESCRIPTION		
1.1. Course objectives		
This course introduces students to key aspects of data-driven research. It involves research on a specific topic and improvement of presentation and communication skills, as well as writing skills, through discussions on the chosen subject with other students		
1.2. Course enrolment requirements		
Requirements for enrolment in the second year of study fulfilled.		
1.3. Expected learning outcomes		
1. Describe the state of the art in the field of data science. 2. Explain the basic terminology and fundamental concepts in the field of data science. 3. Identify scientific and technical problems and challenges in the field of data science. 4. Summarise current topics in the field and present them in a concise written report. 5. Prepare a report in accordance with recommendations for writing professional and scientific publications. 6. Evaluate the appropriateness of different solutions within a given specific topic.		
1.4. Course content		
Searching and analysing scientific and professional literature. Familiarisation with the state of the art. Developing simpler models. Conducting basic experiments. Reporting on current topics in the field and science. Connecting facts in recommended literature with knowledge acquired during studies. Text formatting tools (LaTeX and BibTeX). Preparing reports in accordance with recommendations for writing professional and scientific publications. Applying acquired communication skills for public presentations on the given topic.		
1.5. Types of classes	☒ lectures ☐ seminars and workshops ☐ auditory exercises ☒ distance learning ☐ field work	☐ individual exercises ☐ multimedia and network ☒ laboratory exercises ☐ design exercises ☐ working with a supervisor ☐ other _____
1.6. Comments		
1.7. Student obligations		
Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.		

1.8. Monitoring and assessment of student work

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.9. Assessment and evaluation of student work during classes and in the final exam

STUDENT ACTIVITY	ECTS	LEARNING OUTCOME	TEACHING METHOD	ASSESSMENT METHOD	POINTS	
					Min	Max
Attendance at lectures (L), auditory exercises (AE), laboratory exercises (LE), design exercises (DE)	2	1,2,3,4	Lectures (L), laboratory exercises (LE), design exercises (CE)	Attendance tracking. Minimum attendance percentage: 70%.	0	10
Problem-solving exercises	1	2,4,5	Revision exams (written exam)	Evaluation	10	20
Preparation for laboratory exercises (LE), results analysis, report writing	2	4,5	Laboratory exercises (LE)	Preparation for LE, LE supervision, LE report assessment	10	20
Preparing for an oral exam and oral exam	1	1,2,3,6	Oral exam	Evaluation	25	50

1.10. Obligatory literature (at the time of submitting a study programme proposal)

1. K. L. Turabian: A Manual for Writers of Research Papers, Theses, and Dissertations
2. T. Oetiker, M. Serewin, H. Partl, I. Hyna, E. Schlegl: The not so short introduction to LaTeX, available online: https://tobi.oetiker.ch/lshort/lshort.pdf
3. A. Samardzic, G. Nenadic, P. Jancic: LaTeX 2e za autore, available online: http://poincare.matf.bg.ac.rs/janicic/books/latex2e.pdf
4. J. M. Swales, C.B. Feak: Academic Writing for Graduate Students, Essential Tasks and Skills

1.11. Recommended additional literature (at the time of submitting a study programme proposal)

1. R. E. Berger (2014), A Scientific Approach to Writing for Engineers and Scientists
2. J. Zobel (2014), Writing for Computer Science, 3rd ed., Springer
3. G. Grätzer: More Math Into LaTeX, dostupno online: https://www.springer.com/gp/book/9780387688527
4. M. Hewings, C. Thaine, M. McCarthy: Cambridge Academic English
5. J. T. Vanderplas, J. VanderPlas: Python Data Science Handbook, O'Reilly Media
6. A. Zheng, A. Casari: Feature Engineering for Machine Learning
7. F. Chollet: Deep Learning with Python

1.12. Number of obligatory literature copies in relation to the number of students currently taking the course

Title	*Number of copies*	*Number of students*
K. L. Turabian: A Manual for Writers of Research Papers, Theses, and Dissertations	0	20
T. Oetiker, M. Serewin, H. Partl, I. Hyna, E. Schlegl: The not so short introduction to LaTeX	Available online	20
A. Samardzic, G. Nenadic, P. Jancic: LaTeX 2e za autore	Available online	20

1.13. Quality assurance methods ensuring the acquisition of knowledge, skills and competences
Conducting university questionnaires about teachers (student-teacher relationship, transparency of assessment criteria, motivation for teaching, teaching clarity, etc.). Conducting Faculty surveys of courses (upon passing the exam, student self-assessment of the adopted learning outcomes and student workload in relation to the number of ECTS credits allocated to activities and courses as a whole).

General information		
Lead instructor(s)	Irena Galić, PhD, Full Professor	
Course title	Quantum Computing	
Study programme	Graduate university study programme in Computer Engineering	
Course status	Compulsory on module: Data Science	
Year of study	2	
ECTS credits and teaching methods	ECTS credits	5
	Number of classes (lectures + auditory exercises + laboratory exercises + design exercises + seminars)	30+(15+15+0)+0

1. COURSE DESCRIPTION		
1.1. Course objectives		
Introduce and explain to students the fundamental concepts of quantum computing operation, as well as the implementation of basic quantum computing algorithms.		
1.2. Course enrolment requirements		
Requirements for enrolment in the second year of study fulfilled.		
1.3. Expected learning outcomes		
1. Explain the principles of quantum computers and concepts of quantum computing. 2. Analyse and create quantum logic gates, quantum circuits, and quantum teleportation. 3. Assess the reversibility of quantum logic circuits and the reversibility of classical logic operators. 4. Develop and implement quantum computing algorithms. 5. Apply methods for correcting quantum errors. 6. Examine the fundamental features of technologies considered for the physical realisation of widely applicable quantum computers.		
1.4. Course content		
Introduction to quantum phenomena: light polarisation, Malus' Law, probability, probability amplitude, interference, quantum bit. Explain the basic concepts of quantum computing. Analyse and create quantum logic gates, and implement quantum circuits with IBM Qiskit. Explain the reversibility of a quantum logic circuit and the reversibility of classical logic operators. Implement basic quantum computing algorithms. Implement quantum machine learning algorithms. Apply methods for correcting quantum errors. Physical realisation of quantum computers: DiVincenzo criteria, photonic systems, superconducting Josephson junction, quantum dots, nuclear magnetic resonance.		
1.5. Types of classes	☒ lectures ☐ seminars and workshops ☒ auditory exercises ☒ distance learning ☐ field work	☐ individual exercises ☐ multimedia and network ☒ laboratory exercises ☐ design exercises ☐ working with a supervisor ☐ other _____
1.6. Comments		
1.7. Student obligations		

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.8. Monitoring and assessment of student work

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.9. Assessment and evaluation of student work during classes and in the final exam

STUDENT ACTIVITY	ECTS	LEARNING OUTCOME	TEACHING METHOD	ASSESSMENT METHOD	POINTS	
					Min	Max
Attendance at lectures (L), auditory exercises (AE), laboratory exercises (LE), design exercises (DE)	2	1	Lectures (L), laboratory exercises (LE), design exercises (CE)	Attendance tracking. Minimum attendance percentage: 70%.	0	10
Problem-solving exercises	1	1, 2, 3, 4	Revision exams (written exam)	Evaluation	25	50
Preparation for laboratory exercises (LE), results analysis, report writing	1	4, 5	Laboratory exercises (LE)	Preparation for LE, LE supervision, LE report assessment	10	20
Preparing for an oral exam and oral exam	1	1, 2, 3, 6	Oral exam	Evaluation	10	20

1.10. Obligatory literature (at the time of submitting a study programme proposal)

I. Chuang, M. Nielsen: Quantum Computation and Quantum Information, available online: http://mmrc.amss.cas.cn/tlb/201702/W020170224608149940643.pdf

1.11. Recommended additional literature (at the time of submitting a study programme proposal)

N. D. Mermin, Quantum Computer Science: An Introduction

1.12. Number of obligatory literature copies in relation to the number of students currently taking the course

Title	*Number of copies*	*Number of students*
Isaac Chuang & Michael Nielsen: Quantum Computation and Quantum Information	Available online	20

1.13. Quality assurance methods ensuring the acquisition of knowledge, skills and competences

Conducting university questionnaires about teachers (student-teacher relationship, transparency of assessment criteria, motivation for teaching, teaching clarity, etc.). Conducting Faculty surveys of courses (upon passing the exam, student self-assessment of the adopted learning outcomes and student workload in relation to the number of ECTS credits allocated to activities and courses as a whole).

General information		
Lead instructor(s)	**Goran Martinović, PhD, Full Professor with Tenure** **Ivan Vidović, PhD, Assistant Professor**	
Course title	Software Support Testing Methods and Techniques	
Study programme	Graduate university study programme in Computer Engineering	
Course status	Compulsory on modules: Software Engineering and Data Science Elective on module: Computer Engineering	
Year of study	1. on modules: Software Engineering and Data Science 2. on module: Computer Engineering	
ECTS credits and teaching methods	ECTS credits	6
	Number of classes (lectures + auditory exercises + laboratory exercises + design exercises + seminars)	30 + 0 + 30 + 0 + 0

1. COURSE DESCRIPTION		
1.1. Course objectives		
To explain to students the models, methods and techniques of software testing, ways of manual and automated testing, planning and implementation of testing. Show and analyse the possibilities, selection and ways of using testing techniques throughout the software life cycle, according to various models, as well as static and dynamic, and functional and non-functional testing by using appropriate testing standards. Train students to design test cases and test scenarios, conduct testing in agile, embedded web and mobile programming environments with the use of appropriate tools for testing conduction and automation, as well as program code improvement.		
1.2. Course enrolment requirements		
Requirements for enrolment in the second year of study fulfilled.		
1.3. Expected learning outcomes		
1. Distinguish the basic terms, methods and practices of software engineering and their areas of application 2. Compare different software life cycle models and software process categories 3. Differentiate and select quality assurance practices in the development of software systems 4. Develop a complex plan of testing and quality assurance 5. Differentiate reliability models, select and apply methods, techniques and dynamics of software testing throughout its lifecycle 6. Design the necessary programming and testing environment for functional and non-functional testing 7. Use quantitative methods to evaluate the quality of software systems		
1.4. Course content		
Introduction and basic terms, software reliability, testing models, methods and techniques of software systems testing. Objectives and limitations of testing. Manual and automated testing. Testing planning and monitoring. Test Management. Testing throughout the software lifecycle. Testing technique selection. Testing by black, gray and white box models. Static and dynamic testing. Functional and non-functional testing methods: unit testing, integration testing, system testing, regression testing, acceptance testing, performance testing (load, stress), structural testing, model-based testing, testing of object-oriented programming solutions, usability, security, portability testing, user experience testing. Testing and error reporting documentation. Error/faults analysis. Test case and test scenario design. Testing conduction. Test-driven development. Improving code through testing. Testing standards (ISO/IEC 9126, 9241-11, 25000:2005, 12119, others). Testing risks. Testing in agile environments. Testing of embedded software. Testing of web and mobile applications. Tools for automated testing. Testing examples in considered environments.		
1.5. Types of classes	☒ lectures ☐ seminars and workshops	☒ individual exercises ☐ multimedia and network

	☐ auditory exercises ☒ distance learning ☐ field work	☒ laboratory exercises ☐ design exercises ☐ working with a supervisor ☐ other _____
1.6. Comments		

1.7. Student obligations

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.8. Monitoring and assessment of student work

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.9. Assessment and evaluation of student work during classes and in the final exam

STUDENT ACTIVITY	ECTS	LEARNING OUTCOME	TEACHING METHOD	ASSESSMENT METHOD	POINTS	
					Min	Max
Attendance at lectures (L) and laboratory exercises (LE)	1.5	1,2,3,4,5,6,7	Lectures (L) and laboratory exercises (LE)	Attendance tracking. Minimum attendance percentage: 70%.	0	5
Problem-solving exercises	1.5	2,3,4,5,6,7	Revision exams or written exam	Evaluation of revision exams or written exam	15	30
Preparation for laboratory exercises (LE), results analysis, report writing	1	3,4,5,6,7	Laboratory exercises (LE)	Evaluation of preparation for LE, LE supervision, LE report assessment	10	20
Preparing for an oral exam and oral exam	0.5	1,2,3,4,5,6,7	Oral exam	Evaluation of oral exam	7.5	15
Project work	1.5	1,2,3,4,5,6,7	Written report of project work	Evaluation of written report and solution of project work	15	30

1.10. Obligatory literature (at the time of submitting a study programme proposal)

1. R. Bierig, Essentials of Software Testing (1st Ed.), Cambridge University Press, 2021.
2. D. Graham, E. van Veenendaal, I. Evans, R. Black, Foundations of Software Testing ISTQB Certification (4th Ed.), Cengage Learning EMEA, 2019.
3. G. Mohan, Full Stack Testing: A Practical Guide for Delivering High Quality Software (1st Ed.), O'Reilly Media, 2022.
4. G.J. Myers, C. Sandler, The Art of Software Testing (3rd Ed.), Wiley, 2016.
5. G. Paskal, Test Automation in the Real World: Practical Lessons for Automated Testing. Independently published, 2017.

1.11. Recommended additional literature (at the time of submitting a study programme proposal)

1. G. Blokdyk, Mobile application testing The Ultimate Step-By-Step Guide. 5STARCooks, 2018.
2. B. Laboon, A Friendly Introduction to Software Testing, CreateSpace Independent Publishing Platform (1st Ed.), 2016.

3. R. Das, G. Johnson, Testing and Securing Web Applications (1st Ed.), Auerbach Publications, 2020.
4. S. Siddiqui, Learning Test-Driven Development: A Polyglot Guide to Writing Uncluttered Code (1st Ed.), O'Reilly Media, 2021.
5. M. Winteringham, Testing Web APIs, Manning, 2022.

1.12. Number of obligatory literature copies in relation to the number of students currently taking the course

Title	*Number of copies*	*Number of students*
1. R. Bierig, Essentials of Software Testing, Cambridge University Press, 1st Ed., 2021.	3	80
2. D. Graham, E. van Veenendaal, I. Evans, R. Black, Foundations of Software Testing ISTQB Certification, Cengage Learning EMEA; 4th Ed. 2019.	3	80
3. G. Mohan, Full Stack Testing: A Practical Guide for Delivering High Quality Software. O'Reilly Media; 1st Ed., 2022.	3	80
4. G.J. Myers, C. Sandler, The Art of Software Testing, Wiley; 3rd Ed., 2016.	3	80
5. G. Paskal, Test Automation in the Real World: Practical Lessons for Automated Testing. Independently published, 2017.	3	80

1.13. Quality assurance methods ensuring the acquisition of knowledge, skills and competences

Conducting university questionnaires about teachers (student-teacher relationship, transparency of assessment criteria, motivation for teaching, teaching clarity, etc.). Conducting Faculty surveys of courses (upon passing the exam, student self-assessment of the adopted learning outcomes and student workload in relation to the number of ECTS credits allocated to activities and courses as a whole).

General information

Lead instructor(s)	**Anita Katić, PhD, Assistant Professor**	
Course title	Statistical Data Analysis Methods	
Study programme	Graduate university study programme in Computer Engineering	
Course status	Compulsory on modules: Computer Engineering and Data Science Elective on module: Artificial Intelligence and Robotics	
Year of study	1. on modules: Computer Engineering and Data Science 2. on module: Artificial Intelligence and Robotics	
ECTS credits and teaching methods	ECTS credits	5
	Number of classes (lectures + auditory exercises + laboratory exercises + design exercises + seminars)	30 + (0+20+10)

1. COURSE DESCRIPTION

1.1. Course objectives

Use of statistical methods and tools for data analysis and processing in computer science. Formulation and application of methods using concrete examples. Acquisition of appropriate skills in the use of modern tools for statistical data analysis.

1.2. Course enrolment requirements

Requirements for enrolment in the second year of study fulfilled.

1.3. Expected learning outcomes

1. Recommend and argue statistical methods when statistically analysing data on a given data set
2. Combine the mathematical principles of basic statistical methods with their application to specific data
3. Analyse relationships between statistical variables using regression and correlation analysis techniques
4. Infer the appropriate method for the given regression model
5. Evaluate the results obtained by applying statistical tests to real data and draw reasoned conclusions
6. Evaluate and confirm or refute the appropriateness of the statistical inference procedure for the given data

1.4. Course content

Descriptive statistics and methods of data visualization. Sampling methods. Statistical inference. Hypothesis testing. Regression analysis. Correlation coefficient. Analysis of variance. Non-parametric methods.

1.5. Types of classes	☒ lectures ☐ seminars and workshops ☐ auditory exercises ☒ distance learning ☐ field work	☒ individual exercises ☐ multimedia and network ☒ laboratory exercises ☒ design exercises ☐ working with a supervisor ☐ other _____
1.6. Comments		

1.7. Student obligations

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.8. Monitoring and assessment of student work

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.9. Assessment and evaluation of student work during classes and in the final exam

STUDENT ACTIVITY	ECTS	LEARNING OUTCOME	TEACHING METHOD	ASSESSMENT METHOD	POINTS	
					Min	Max
Attendance at lectures (L), laboratory exercises (LE)	1	1,2,3,6	Attendance at lectures (L), laboratory exercises (LE)	Attendance tracking. Minimum attendance percentage: 70%.	5	10
Preparation for laboratory exercises (LE), results analysis, report writing	1	2,3,4,5,6	Laboratory exercises (LE)	Preparation for LE, LE supervision, LE report assessment	10	20
Preparing for an oral exam and oral exam	1.5	1,2,3,4,5,6	Oral exam	Evaluation	25	50
Problem-solving exercises. Design exercises (DE)	1.5	2,4,5,6	Design exercises (DE)	Evaluation	10	20

1.10. Obligatory literature (at the time of submitting a study programme proposal)

1. Montgomery, D.C., Runger, G.C., Applied Statistics and Probability for engineers. USA: Wiley, 2018.
2. Agresti, Alan, and Maria Kateri. Foundations of Statistics for Data Scientists: With R and Python. Chapman and Hall/CRC, 2021.

1.11. Recommended additional literature (at the time of submitting a study programme proposal)

1. Ott, R. Lyman, and Micheal T. Longnecker. An introduction to statistical methods and data analysis. Cengage Learning, 2015.
2. Kenett, R., Zacks, S., Gedeck,P., Modern Statistics: A Computer Based Approach with Python, Springer, 2022.

1.12. Number of obligatory literature copies in relation to the number of students currently taking the course

Title	*Number of copies*	*Number of students*
Montgomery, D.C., Runger, G.C., Applied Statistics and Probability for engineers. USA: Wiley, 2018.	1	
Agresti, Alan, and Maria Kateri. Foundations of Statistics for Data Scientists: With R and Python. Chapman and Hall/CRC, 2021.	0	

1.13. Quality assurance methods ensuring the acquisition of knowledge, skills and competences

Conducting university questionnaires about teachers (student-teacher relationship, transparency of assessment criteria, motivation for teaching, teaching clarity, etc.). Conducting Faculty surveys of courses (upon passing the exam, student self-assessment of the adopted learning outcomes and student workload in relation to the number of ECTS credits allocated to activities and courses as a whole).

General information

Lead instructor(s)	**Zdravko Krpić, PhD, Associate Professor**	
Course title	Modelling and Design of Software Systems	
Study programme	Graduate university study programme in Computer Engineering	
Course status	Compulsory on module: Software Engineering	
Year of study	1.	
ECTS credits and teaching methods	ECTS credits	7
	Number of classes (lectures + auditory exercises + laboratory exercises + design exercises + seminars)	30+(15+15+0)+0

1. COURSE DESCRIPTION

1.1. Course objectives

To teach students with the concepts of software engineering for complex software systems, categorise and explain software development models with the most important methods from each. To teach students the principles of modelling and design of software systems for software products and for customised software through various stages of software development (requirements analysis and specification, high and low level design, implementation, validation and verification, evolution). To produce the necessary models and core implementation for software based on the requirements specification and using appropriate tools and techniques.

1.2. Course enrolment requirements

Requirements for enrolment in the second year of study fulfilled.

1.3. Expected learning outcomes

1. Determine the specifics of software engineering of large-scale projects
2. Analyse and define the requirements of complex software products
3. Present data models, object models, contextual models and behavioural models using different representation techniques
4. Identify different software architectures and their elements (architectural styles, languages, connectors, middleware)
5. Create the design of a complex software product
6. Critically judge implementation issues such as modularity and coding standards

1.4. Course content

The course introduces the principles of modelling and design of large and complex software systems. Most of today's programming systems require a systematic approach to specification and design at a higher abstract level than programming languages. The course includes an introduction to general conceptual design, i.e. software architecture. Students will be provided with a theoretical basis for system design, UML, design rules (design patterns), design based on models and components. In addition, students will acquire practical knowledge of software system design through laboratory exercises and projects.

1.5. Types of classes

☒ lectures	☐ individual exercises
☐ seminars and workshops	☐ multimedia and network
☒ auditory exercises	☒ laboratory exercises
☒ distance learning	☐ design exercises
☐ field work	☐ working with a supervisor
	☐ other _____

1.6. Comments	

1.7. Student obligations

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.8. Monitoring and assessment of student work

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.9. Assessment and evaluation of student work during classes and in the final exam

STUDENT ACTIVITY	ECTS	LEARNING OUTCOME	TEACHING METHOD	ASSESSMENT METHOD	POINTS	
					Min	Max
Attendance at lectures (L), auditory exercises (AE), laboratory exercises (LE)	2	1, 2, 3, 4, 5	Lectures (L), laboratory exercises (LE)	Attendance tracking. Minimum attendance percentage: 70%.	7	10
Problem-solving exercises	1	2, 4, 5, 6	Revision exams (written exam)	Evaluation	20	40
Preparation for laboratory exercises (LE), results analysis, report writing	2	2, 3, 4, 5	Laboratory exercises (LE)	Preparation for LE, LE supervision, LE report assessment	10	20
Preparing for an oral exam and oral exam	1	1, 2, 4, 6	Oral exam	Evaluation	15	30

1.10. Obligatory literature (at the time of submitting a study programme proposal)

1. Sommerville: Software Engineering 10th edition, Pearson Education, 2016
2. Hassan Gomaa: Software Modeling and Design: UML, Use Cases, Patterns, and Software Architectures, Cambridge University Press, 2011.

1.11. Recommended additional literature (at the time of submitting a study programme proposal)

1. I. Sommerville: Engineering Software Products: An Introduction to Modern Software Engineering, Pearson, 2021.

1.12. Number of obligatory literature copies in relation to the number of students currently taking the course

Title	*Number of copies*	*Number of students*
Software Engineering 10th edition	1	
Software Modeling and Design: UML, Use Cases, Patterns, and Software Architectures	1	
Engineering Software Products: An Introduction to Modern Software Engineering	1	

1.13. Quality assurance methods ensuring the acquisition of knowledge, skills and competences

Conducting university questionnaires about teachers (student-teacher relationship, transparency of assessment criteria, motivation for teaching, teaching clarity, etc.). Conducting Faculty surveys of courses

(upon passing the exam, student self-assessment of the adopted learning outcomes and student workload in relation to the number of ECTS credits allocated to activities and courses as a whole).

General information

Lecturer(s)	Dražen Slišković, PhD, Full professor	
Course title	Data-Based Modelling	
Study programme	Graduate university study programme in Computer Engineering	
Course status	Compulsory on module: Artificial Intelligence and Robotics	
Year of study	1	
ECTS credits and teaching methods	ECTS credits	6
	Number of classes (lectures + auditory exercises + laboratory exercises + design exercises + seminars)	30+0+30+0+0

1. COURSE DESCRIPTION

1.1. Course objectives

To acquaint students with the basics of the methodology of extracting knowledge about the process contained in the available measurement data, and how to build a process model with the desired properties based on this information. Acquiring appropriate skills in working with available software tools for analysis and processing of measurement data, as well as tools for building process models based on this data. To acquaint students with the method of improving the automatic control system based on the knowledge extracted from measurement data and the method of building a fault-tolerant system, as the forms of introducing intelligence into the technical system.

1.2. Course enrolment requirements

Requirements for enrolment in the second year of study fulfilled.

1.3. Expected learning outcomes

1. collect, analyze and pre-process measurement data, and create data sets for model building
2. highlight the advantages and disadvantages of a particular method of identification for a particular problem of process identification
3. create a dynamic mathematical model of the process with several identification methods implemented in the Matlab software package
4. explain the problems of process monitoring and control system realization with the existence of a malfunction in the measurement system, as well as the existence of a difficult-to-measure process variable, and the way to solve the problem by applying a process variable estimator (soft-sensor)
5. explain the problem of building a model based on high-dimensional plant data and the mathematical basis for building a model with good predictive properties
6. evaluate the suitability of a particular method based on the projection of the input data space into the latent space for the given problem of process modeling and building of the process variable estimator
7. build a process variable estimator based on plant data, applying the analyzed methods, using the Matlab software package

1.4. Course content

Modeling processes and other functional relationships in data, based on measurement data. Measurement data obtained in a separate experiment and operational (plant) data. Informativeness of measurement data. Analysis and preprocessing of measurement data and the formation of data sets for the building of process models. Construction of a static and dynamic process model. Non-parametric and parametric methods of identification. Non-recursive and recursive methods of model parameter estimation. Estimation of a process variable based on information about other quantities that are related to it. Selection of model structure and input variables. Building models based on high-dimensional data. Regression modeling and criteria for estimating model parameters. Methods based on the projection of the input space into the latent (sub)space. Linear and non-linear modeling methods. Application of artificial neural networks in data based modeling.

Validation/evaluation of models built on the basis on data. Application of the Matlab software package in data-based modeling. Virtual (soft) sensor and its implementation.

1.5. Types of classes	☒ lectures ☐ seminars and workshops ☐ auditory exercises ☒ distance learning ☐ field work	☐ individual exercises ☐ multimedia and network ☒ laboratory exercises ☐ design exercises ☐ working with a supervisor ☒ other - project assignment of choice
1.6. Comments		

1.7. Student obligations

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.8. Monitoring and assessment of student work

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.9. Assessment and evaluation of student work during classes and in the final exam

STUDENT ACTIVITY	ECTS	LEARNING OUTCOME	TEACHING METHOD	ASSESSMENT METHOD	POINTS	
					Min	Max
Attendance at lectures (L), laboratory exercises (LE)	2	1, 2, 3, 4, 5, 6, 7	Lectures (L), laboratory exercises (LE), design exercises (CE)	Attendance tracking. Minimum attendance percentage: 70%.	5	8
Preparation for laboratory exercises (LE), results analysis, report writing	1.5	1, 3, 6, 7	Laboratory exercises (LE)	Preparation for LE, LE supervision, LE report assessment	15	30
Solving project assignment	1	1, 2, 3, 4, 5, 6, 7	Individual work	Project report assessment	0	25
Preparing for an oral exam and oral exam	1.5	1, 2, 4, 5, 6	Oral exam	Evaluation	20	37

1.10. Obligatory literature (at the time of submitting a study programme proposal)

1. Slišković, D., Modeliranje temeljeno na podacima – predavanja, zavodska skripta

1.11. Recommended additional literature (at the time of submitting a study programme proposal)

1. Fortuna, L., S. Graziani, A. Rizzo, M.G. Xibilia; Soft sensors for Monitoring and Control of Industrial Processes
2. Ljung, L., System Identification - Theory for the User
3. Martens, H., T. Naes, Multivariate Calibration, 2nd edition
4. Haykin, S., Neural Networks – A Comprehensive Foundation, 2nd edition

1.12. Number of obligatory literature copies in relation to the number of students currently taking the course

Title	*Number of copies*	*Number of students*

1.13. Quality assurance methods ensuring the acquisition of knowledge, skills and competences		
Conducting university questionnaires about teachers (student-teacher relationship, transparency of assessment criteria, motivation for teaching, teaching clarity, etc.). Conducting Faculty surveys of courses (upon passing the exam, student self-assessment of the adopted learning outcomes and student workload in relation to the number of ECTS credits allocated to activities and courses as a whole).		

General information		
Lead instructor(s)	Tomislav Galba, PhD, Assistant Professor Alfonzo Baumgartner, PhD, Associate Professor	
Course title	Advanced Data Structures and Algorithms	
Study programme	Graduate university study programme in Computer Engineering	
Course status	Compulsory on modules: Software Engineering and Data Science Elective on modules: Computer Engineering and Artificial Intelligence and Robotics	
Year of study	1.	
ECTS credits and teaching methods	ECTS credits	5
	Number of classes (lectures + auditory exercises + laboratory exercises + design exercises + seminars)	30+(15+15+0)+0

1. COURSE DESCRIPTION		
1.1. Course objectives		
Introduction and analysis of different advanced data structures like dictionaries, hash tables, AVL trees, Red-Black trees and graphs. Implementation of different algorithms that use advanced data structures for sorting, pattern matching, searching, insertion, deletion of data, etc.		
1.2. Course enrolment requirements		
Passed and attended Algorithms and data structures course.		
1.3. Expected learning outcomes		
1. Choose and apply a suitable method for an algorithm analysis. 2. Prove the accuracy of the algorithm and analyse its efficiency. 3. Choose and evaluate a suitable algorithm for the already known (and solved) problem. 4. Choose an algorithm design technique for an unknown problem. 5. Design a custom algorithm for an unknown problem and compare it with other algorithms.		
1.4. Course content		
Part 1: Dictionaries and Hash tables as data structures with corresponding algorithms. Part 2: Binary search trees, AVL trees, Red-Black trees, Splay trees with corresponding algorithms for search, insert and delete operations. Part 2: Graphs and their representation. Strongly connected components identification problem, bipartite matching, negative cycles in the graph, minimum spanning tree. Part 4: Pattern matching algorithms: Boyer-Moore, Knuth-Morris-Pratt.		
1.5.	☒ lectures ☐ seminars and workshops ☒ auditory exercises ☒ distance learning ☐ field work	☒ individual exercises ☐ multimedia and network ☒ laboratory exercises ☐ design exercises ☐ working with a supervisor ☐ other _____
1.6. Comments		
1.7. Student obligations		

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.8. Monitoring and assessment of student work

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.9. Assessment and evaluation of student work during classes and in the final exam

STUDENT ACTIVITY	ECTS	LEARNING OUTCOME	TEACHING METHOD	ASSESSMENT METHOD	POINTS	
					Min	Max
Attendance at lectures (L), auditory exercises (AE), laboratory exercises (LE), design exercises (DE)	1	1,2,3,4	Lectures (L), laboratory exercises (LE), design exercises (CE)	Attendance tracking. Minimum attendance percentage: 70%.	0.7	1
Problem-solving exercises	2	3,4,5	Revision exams (written exam)	Evaluation	1	2
Preparation for laboratory exercises (LE), results analysis, report writing	1	3,4,5	Laboratory exercises (LE)	Preparation for LE, LE supervision, LE report assessment	0.5	1
Preparing for an oral exam and oral exam	1	1, 2, 3, 4	Oral exam	Evaluation	0.5	1

1.10. Obligatory literature (at the time of submitting a study programme proposal)

1. Cormen, Thomas H.; Leiserson, Charles E.; Rivest, Ronald L.; Stein, Clifford (2009) [1990]. Introduction to Algorithms (3rd ed.). MIT Press and McGraw-Hill. ISBN 0-262-03384-4.

1.11. Recommended additional literature (at the time of submitting a study programme proposal)

1. Data Structures using C – A. S. Tanenbaum, Y. Langsam, and M.J. Augenstein, PHI/Pearson Education.

2. Art of Computer Programming, The: Volume 3: Sorting and Searching, Addison-Wesley Professional; 2nd edition (April 24, 1998)

1.12. Number of obligatory literature copies in relation to the number of students currently taking the course

Title	*Number of copies*	*Number of students*
Introduction to Algorithms (3rd ed.)	?	?

1.13. Quality assurance methods ensuring the acquisition of knowledge, skills and competences

Conducting university questionnaires about teachers (student-teacher relationship, transparency of assessment criteria, motivation for teaching, teaching clarity, etc.). Conducting Faculty surveys of courses (upon passing the exam, student self-assessment of the adopted learning outcomes and student workload in relation to the number of ECTS credits allocated to activities and courses as a whole).

General information		
Lead instructor(s)	**Robert Cupec, PhD, Full Professor with Tenure** **Damir Filko, PhD, Associate Professor**	
Course title	Advanced Robot Systems	
Study programme	Graduate university study programme in Computer Engineering	
Course status	Compulsory on modules: Artificial Intelligence and Robotics	
Year of study	2.	
ECTS credits and teaching methods	ECTS credits	5
	Number of classes (lectures + auditory exercises + laboratory exercises + design exercises + seminars)	30+(0+30+15)+0

1. COURSE DESCRIPTION		
1.1. Course objectives		
Introduction to advanced methods for controlling robotic manipulators. To analyse approaches for planning manipulator trajectories and grasping strategies. To introduce students to reinforcement learning of robot operations and human-robot interaction. Introduce students to multi-robot systems and industrial robot systems.		
1.2. Course enrolment requirements		
Requirements for enrolment in the second year of study fulfilled.		
1.3. Expected learning outcomes		
1. Explain inverse model control of the robot. 2. Explain and apply contact force control and explain impedance control. 3. Create a program for path planning of a robot manipulator and manipulation of objects. 4. Explain the application of reinforcement learning of robot operations. 5. Explain the methods of human-robot interaction. 6. Explain the problems of multi-robot and industrial robot systems.		
1.4. Course content		
Inverse model control. Force control. Impedance control. Robot manipulator trajectory planning and object grasping. Visual servoing. Behaviour tree. Reinforcement learning of robot operations. Human-robot interaction. Mobile manipulators. Multi-robot systems. Industrial robot systems.		
1.5. Types of classes	☒ lectures ☐ seminars and workshops ☐ auditory exercises ☒ distance learning ☐ field work	☐ individual exercises ☐ multimedia and network ☒ laboratory exercises ☒ design exercises ☐ working with a supervisor ☐ other _____
1.6. Comments		
1.7. Student obligations		
Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.		
1.8. Monitoring and assessment of student work		

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.9. Assessment and evaluation of student work during classes and in the final exam

STUDENT ACTIVITY	ECTS	LEARNING OUTCOME	TEACHING METHOD	ASSESSMENT METHOD	POINTS	
					Min	Max
Attendance at lectures (L), laboratory exercises (LE), design exercises (DE)	2	1, 2, 3, 4, 5, 6	Lectures (L), laboratory exercises (LE), design exercises (DE)	Attendance tracking. Minimum attendance percentage: 70%.	7	10
Problem-solving exercises set during design exercises (DE)	1	1, 2, 3, 4, 5, 6	Design exercises (DE)	Evaluation	8	30
Preparation for laboratory exercises (LE), results analysis, report writing	1	1,2, 3, 4, 5, 6	Laboratory exercises (LE)	Preparation for LE, LE supervision, LE report assessment	10	30
Preparing for an oral exam and oral exam	1	1, 2, 4, 5, 6	Oral exam	Evaluation	10	30

1.10. Obligatory literature (at the time of submitting a study programme proposal)

1. J. J. Craig, Introduction to Robotics: Mechanics and Control, Pearson Prentice Hall, Upper Saddle River, New Jersey, 2005

1.11. Recommended additional literature (at the time of submitting a study programme proposal)

1. F. L. Lewis, K. Hengster-Movric, H. Zhang, A. Dasgupta: Cooperative Control of Multi-Agent Systems: Optimal and Adaptive Design Approaches, Springer, 2014

2. C. Bartneck, Human-Robot Interaction: An Introduction, Cambridge University Press, 2020.

1.12. Number of obligatory literature copies in relation to the number of students currently taking the course

Title	*Number of copies*	*Number of students*
1. J. J. Craig, Introduction to Robotics: Mechanics and Control, Pearson Prentice Hall, Upper Saddle River, New Jersey, 2005	0	16

1.13. Quality assurance methods ensuring the acquisition of knowledge, skills and competences

Conducting university questionnaires about teachers (student-teacher relationship, transparency of assessment criteria, motivation for teaching, teaching clarity, etc.). Conducting Faculty surveys of courses (upon passing the exam, student self-assessment of the adopted learning outcomes and student workload in relation to the number of ECTS credits allocated to activities and courses as a whole).

General information		
Lead instructor(s)	**Ivica Lukić, PhD, Associate Professor**	
Course title	Advanced Web Programming	
Study programme	Graduate university study programme in Computer Engineering	
Course status	Compulsory on modules: Computer Engineering and Software Engineering Elective on modules: Artificial Intelligence and Robotics and Data Science	
Year of study	2.	
ECTS credits and teaching methods	ECTS credits	5
	Number of classes (lectures + auditory exercises + laboratory exercises + design exercises + seminars)	30+(0+30+0)+0

1. COURSE DESCRIPTION		
1.1. Course objectives		
The aim of the course is to explain to students the process of designing user interfaces as well as the underlying application when developing web applications. Students will become familiar with complex programming frameworks used in web application development, which significantly differ from conventional approaches to developing web applications without using development frameworks. They will be introduced to the most renowned frameworks for the rapid development of high-quality and interactive web applications.		
1.2. Course enrolment requirements		
Requirements for enrolment in the second year of study fulfilled.		
1.3. Expected learning outcomes		
1. Compare different client-side technologies for developing web applications 2. Evaluate various technologies and server-side solutions for building web applications 3. Create complex software solutions based on advanced web technologies and services 4. Analyse and solve specific problems, combining different technologies and programming frameworks for web application development		
1.4. Course content		
1.5. Types of classes	☒ lectures ☐ seminars and workshops ☐ auditory exercises ☒ distance learning ☐ field work	☒ individual exercises ☐ multimedia and network ☒ laboratory exercises ☐ design exercises ☐ working with a supervisor ☐ other _____
1.6. Comments		
1.7. Student obligations		
Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.		
1.8. Monitoring and assessment of student work		

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.9. Assessment and evaluation of student work during classes and in the final exam

STUDENT ACTIVITY	ECTS	LEARNING OUTCOME	TEACHING METHOD	ASSESSMENT METHOD	POINTS	
					Min	Max
Attendance at lectures (L), auditory exercises (AE), laboratory exercises (LE), design exercises (DE)	2	1,2,3	Lectures (L), laboratory exercises (LE), design exercises (CE)	Attendance tracking. Minimum attendance percentage: 70%.	6	10
Problem-solving exercises	1	2,3,4	Revision exams (written exam)	Evaluation	15	30
Preparation for laboratory exercises (LE), results analysis, report writing	1	2,3,4	Laboratory exercises (LE)	Preparation for LE, LE supervision, LE report assessment	15	20
Preparing for an oral exam and oral exam	1	1,2,3	Oral exam	Evaluation	20	40

1.10. Obligatory literature (at the time of submitting a study programme proposal)

Christopher Pitt, Pro PHP 8 MVC: Model View Controller Architecture-Driven Application Development, Apress, 2021.
David Herron, Node.js Web Development: Server-side web development made easy with Node 14 using practical examples, 5th Edition, Packt Publishing, 2020.

1.11. Recommended additional literature (at the time of submitting a study programme proposal)

Ethan Brown, Web Development with Node and Express: Leveraging the JavaScript Stack, O' Reilly Media, Inc., 2019.

Matt Stauffer, Laravel: Up & Running, 2nd Edition, O' Reilly Media, Inc., 2019.

L. Ullman, PHP Advanced and Object-Oriented Programming: Visual QuickPro Guide (3rd Edition),

Peachpit Press, 1301 Sansome Street, San Francisco, CA 94111, 2012.

1.12. Number of obligatory literature copies in relation to the number of students currently taking the course

Title	*Number of copies*	*Number of students*

1.13. Quality assurance methods ensuring the acquisition of knowledge, skills and competences

Conducting university questionnaires about teachers (student-teacher relationship, transparency of assessment criteria, motivation for teaching, teaching clarity, etc.). Conducting Faculty surveys of courses (upon passing the exam, student self-assessment of the adopted learning outcomes and student workload in relation to the number of ECTS credits allocated to activities and courses as a whole).

General information

Lead instructor(s)	**Krešimir Romić, PhD, Assistant Professor**	
Course title	NoSQL Databases	
Study programme	Graduate university study programme in Computer Engineering	
Course status	Compulsory on modules: Software Engineering	
Year of study	2.	
ECTS credits and teaching methods	ECTS credits	5
	Number of classes (lectures + auditory exercises + laboratory exercises + design exercises + seminars)	30+0+30+0+0

1. COURSE DESCRIPTION

1.1. Course objectives

Acquire knowledge related to NoSQL databases. Distinguish between NoSQL databases and relational databases and recognize the appropriate application for individual NoSQL database models. Learn how to store and process data through examples of NoSQL database implementation.

1.2. Course enrolment requirements

Requirements for enrolment in the second year of study fulfilled.

1.3. Expected learning outcomes

1. Distinguish and apply the concepts of advanced data storage models including non-relational databases.
2. Choose storage methods suitable for special types of data such as temporal, multimedia and spatial.
3. Select and apply different techniques for processing data stored in advanced data storage systems.
4. Identify and use knowledge extracted from large volumes of data.

1.4. Course content

The concept of NoSQL (non-relational) databases and comparison with relational databases. Division into models and highlighting the most commonly used NoSQL databases with regard to application. Key-value model (Key-value store). Document-based model. Column-based model. Graph-based model. Application of different models of NoSQL databases through implementation examples.

1.5. Types of classes	☒ lectures ☐ seminars and workshops ☐ auditory exercises ☒ distance learning ☐ field work	☐ individual exercises ☐ multimedia and network ☒ laboratory exercises ☐ design exercises ☐ working with a supervisor ☐ other _____
1.6. Comments		

1.7. Student obligations

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.8. Monitoring and assessment of student work

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.9. Assessment and evaluation of student work during classes and in the final exam

STUDENT ACTIVITY	ECTS	LEARNING OUTCOME	TEACHING METHOD	ASSESSMENT METHOD	POINTS	
					Min	Max
Attendance at lectures (L), laboratory exercises (LE)	0.5	1,2,3,4	Lectures (L), laboratory exercises (LE)	Attendance tracking. Minimum attendance percentage: 70%.	3.5	5
Problem-solving exercises	2	1,2,3,4	Revision exams (written exam)	Evaluation	20	40
Preparation for laboratory exercises (LE), results analysis, report writing	1	2,3,4	Laboratory exercises (LE)	Preparation for LE, LE supervision, LE report assessment	12.5	25
Preparing for an oral exam and oral exam	1.5	1,2,3	Oral exam	Evaluation	15	30

1.10. Obligatory literature (at the time of submitting a study programme proposal)

1. Christof Strauch: NoSQL Databases
2. Pramod J. Sadalage, Martin Fowler: NoSQL Distilled - A Brief Guide to the Emerging World of Polyglot Persistence

1.11. Recommended additional literature (at the time of submitting a study programme proposal)

1. MongoDB Documentation (https://www.mongodb.com/docs/)
2. Redis Documentation (https://redis.io/docs/)
3. Neo4j Documentation (https://neo4j.com/docs/)

1.12. Number of obligatory literature copies in relation to the number of students currently taking the course

Title	*Number of copies*	*Number of students*
NoSQL Databases	1	
NoSQL Distilled - A Brief Guide to the Emerging World of Polyglot Persistence	1	

1.13. Quality assurance methods ensuring the acquisition of knowledge, skills and competences

Conducting university questionnaires about teachers (student-teacher relationship, transparency of assessment criteria, motivation for teaching, teaching clarity, etc.). Conducting Faculty surveys of courses (upon passing the exam, student self-assessment of the adopted learning outcomes and student workload in relation to the number of ECTS credits allocated to activities and courses as a whole).

General information		
Lead instructor(s)	**Anita Katić, PhD, Assistant Professor**	
Course title	Numerical Methods in Computer Science	
Study programme	Graduate university study programme in Computer Engineering	
Course status	Elective on module: Data Science	
Year of study	1.	
ECTS credits and teaching methods	ECTS credits	5
	Number of classes (lectures + auditory exercises + laboratory exercises + design exercises + seminars)	30 + (0+20+10)+0

1. COURSE DESCRIPTION		
1.1. Course objectives		
Acquire theoretical and practical knowledge of methods and algorithms in the field of numerical mathematics to solve applied numerical problems. Argue about the accuracy and limitations of methods and algorithms. Apply programming and numerical modelling skills to perform, design, implement, test and measure numerical algorithms.		
1.2. Course enrolment requirements		
Requirements for enrolment in the second year of study fulfilled.		
1.3. Expected learning outcomes		
1. Implement the algorithms of appropriate methods with the aim of solving given numerical models. 2. Evaluate and critically interpret the results obtained when solving numerical problems and suggest possible improvements to the solution. 3. Evaluate and justify your conclusions about the errors of numerical methods. 4. Compare and create a function based on data analysis using appropriate approximation and interpolation methods. 5. Analyse and create the solution of the given model by solving systems of linear and non-linear equations. 6. Apply basic knowledge of least squares problems to the solution of practical mathematical models. 7. Interpret the basic principles of approximate solution of differential equations.		
1.4. Course content		
Error analysis. Methods and algorithms for solving linear and nonlinear equations. Methods and algorithms for solving systems of linear and nonlinear equations. Methods and algorithms for solving interpolation and approximation problems. Methods and algorithms for solving least squares problems. Methods and algorithms for solving problems with eigenvalues and eigenvectors. Analysis and methods of numerical integration and numerical differentiation. Methods and algorithms for the approximate solution of differential equations.		
1.5. Types of classes	☒ lectures ☐ seminars and workshops ☐ auditory exercises ☒ distance learning ☐ field work	☒ individual exercises ☐ multimedia and network ☒ laboratory exercises ☒ design exercises ☐ working with a supervisor ☐ other _____
1.6. Comments		

1.7. Student obligations

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.8. Monitoring and assessment of student work

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.9. Assessment and evaluation of student work during classes and in the final exam

STUDENT ACTIVITY	ECTS	LEARNING OUTCOME	TEACHING METHOD	ASSESSMENT METHOD	POINTS	
					Min	Max
Attendance at lectures (L), laboratory exercises (LE)	1	1,2,3,4,5,6,7	Attendance at lectures (L), laboratory exercises (LE)	Attendance tracking. Minimum attendance percentage: 70%.	5	10
Preparation for laboratory exercises (LE), results analysis, report writing	1	1,3,4,5,6,7	Laboratory exercises (LE)	Preparation for LE, LE supervision, LE report assessment	10	20
Preparing for an oral exam and oral exam	1.5	1, 2, 3, 4,5,6,7	Oral exam	Evaluation	25	50
Problem-solving exercises. Design exercises (DE)	1.5	1, 2, 5,6	Design exercises (DE)	Evaluation	10	20

1.10. Obligatory literature (at the time of submitting a study programme proposal)

1. Chapra, Steven C., and David E. Clough. Applied Numerical Methods with Python for Engineers and Scientists. McGrawHill Education, 2022.
2. Scitovski, R. Numerička matematika. Osijek: Sveučilište J.J. Strossmayera u Osijeku, Odjel za matematiku, 2015.

1.11. Recommended additional literature (at the time of submitting a study programme proposal)

3. A. Gilat, Numerical Methods for engineers and scientists, Wiley; 2013.
4. Chapra, Steven. Applied Numerical Methods with MATLAB for Engineers and Scientists. McGraw Hill, 2011.
5. A. Kharab, R. Guenther, An Introduction to Numerical Methods, CRC Press; 2021.
6. Jaan Kiusalaas, Numerical Methods in Engineering with Python 3, Cambridge University Press, 2013.
Jaan Kiusalaas-Numerical methods in engineering with MATLAB-Cambridge University Press, 2010.

1.12. Number of obligatory literature copies in relation to the number of students currently taking the course

Title	*Number of copies*	*Number of students*
Chapra, Steven C., and David E. Clough. Applied Numerical Methods with Python for Engineers and Scientists. McGraw Hill Education, 2022.	0	

1.13. Quality assurance methods ensuring the acquisition of knowledge, skills and competences

Conducting university questionnaires about teachers (student-teacher relationship, transparency of assessment criteria, motivation for teaching, teaching clarity, etc.). Conducting Faculty surveys of courses (upon passing the exam, student self-assessment of the adopted learning outcomes and student workload in relation to the number of ECTS credits allocated to activities and courses as a whole).

General information

Lecturer(s)	**Petra Pejić, PhD, Assistant Professor, Josip Job, PhD, Full Professor**	
Course title	Natural Language Processing	
Study programme	Graduate university study programme in Computer Engineering	
Course status	Compulsory on module: Data Science Elective on modules: Artificial Intelligence and Robotics and Software Engineering	
Year of study	2.	
ECTS credits and teaching methods	ECTS credits	6
	Number of classes (lectures + auditory exercises + laboratory exercises + design exercises + seminars)	30 + (0 + 30 + 0) + 0

1. COURSE DESCRIPTION

1.1. Course objectives

Introduction to the principles and methods of natural language processing and its applications. Acquiring appropriate skills in working with programming tools for the design and implementation of natural language processing methods that enable solving specific problems in various areas of technology and human activity.

1.2. Course enrolment requirements

Requirements for enrolment in the second year of study fulfilled.

1.3. Expected learning outcomes

1. Design and implement computer programs for automatic processing, analysis, and understanding of text and speech.
2. Propose a system design adapted to the target language.
3. Differentiate between semantic and syntactic analysis of text and speech.
4. Recommend suitable methods for solving specific problems in automatic processing and analysis of text and speech.
5. Evaluate methods for automatic processing and analysis of text and speech.

1.4. Course content

Theoretical foundations of natural language processing. Word categorization (part-of-speech tagging, POS). Named-entity recognition (NER). Text and speech classification using machine learning. Semantic and syntactic analysis of text and speech. Finding the topic of the text (eng. topic modeling). Application of deep learning in natural language processing. Software packages for natural language processing (text and speech).

1.5. Types of classes	☒ lectures ☐ seminars and workshops ☐ auditory exercises ☒ distance learning ☐ field work	☐ individual exercises ☐ multimedia and network ☒ laboratory exercises ☐ design exercises ☐ working with a supervisor ☐ other _____
1.6. Comments		

1.7. Student obligations

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.8. Monitoring and assessment of student work

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.9. Assessment and evaluation of student work during classes and in the final exam

STUDENT ACTIVITY	ECTS	LEARNING OUTCOME	TEACHING METHOD	ASSESSMENT METHOD	POINTS	
					Min	Max
Attendance at lectures (L), auditory exercises (AE), laboratory exercises (LE), design exercises (DE)	1,4	1, 2, 3, 4, 5	Lectures (L), laboratory exercises (LE)	Attendance tracking. Minimum attendance percentage: 70%.	0	0
Preparation for laboratory exercises (LE), results analysis, report writing	2,1	1, 2, 3, 4, 5	Laboratory exercises (LE)	Preparation for LE, LE supervision, LE report assessment	25	50
Preparing for an oral exam and oral exam	2,5	1, 2, 3, 4, 5, 6	Oral exam	Evaluation	25	50

1.10. Obligatory literature (at the time of submitting a study programme proposal)

1. Dan Jurafsky and James H. Martin. Speech and Language Processing (3rd ed. draft) https://web.stanford.edu/~jurafsky/slp3/

2. Jacob Eisenstein. Natural Language Processing https://github.com/jacobeisenstein/gt-nlp-class/blob/master/notes/eisenstein-nlp-notes.pdf

1.11. Recommended additional literature (at the time of submitting a study programme proposal)

1. Yoav Goldberg. A Primer on Neural Network Models for Natural Language Processing https://u.cs.biu.ac.il/~yogo/nnlp.pdf

2. Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep Learning https://www.deeplearningbook.org/

1.12. Number of obligatory literature copies in relation to the number of students currently taking the course

Title	*Number of copies*	*Number of students*

1.13. Quality assurance methods ensuring the acquisition of knowledge, skills and competences

Conducting university questionnaires about teachers (student-teacher relationship, transparency of assessment criteria, motivation for teaching, teaching clarity, etc.). Conducting Faculty surveys of courses (upon passing the exam, student self-assessment of the adopted learning outcomes and student workload in relation to the number of ECTS credits allocated to activities and courses as a whole).

General information		
Lead instructor(s)	**Irena Galić, PhD, Full Professor**	
Course title	Image Processing and Computer Vision	
Study programme	Graduate university study programme in Computer Engineering	
Course status	Compulsory on modules: Artificial Intelligence and Robotics and Data Science Elective on modules: Computer Engineering and Software Engineering	
Year of study	1.	
ECTS credits and teaching methods	ECTS credits	6
	Number of classes (lectures + auditory exercises + laboratory exercises + design exercises + seminars)	45+(0+30+0)+0

1. COURSE DESCRIPTION		
1.1. Course objectives		
Introduce students to the fundamental methods used in image processing and computer vision, from basic image transformations, image enhancement, and feature extraction to fundamental computer vision algorithms. Through programming assignments, familiarize students with the functioning of image processing and computer vision algorithms.		
1.2. Course enrolment requirements		
Fulfilled requirements for study enrolment.		
1.3. Expected learning outcomes		
1. Define and describe the concepts of image processing and computer vision. 2. Describe methods of image processing and computer vision. 3. Apply the foundations of image processing and computer vision and assess the results. 4. Analyse practical issues in digital image processing. 5. Use and adapt basic algorithms for image processing and computer vision and interpret the results. 6. Connect acquired knowledge and apply methods for image processing and computer vision in open-source applications, interpreting the outcomes.		
1.4. Course content		
Types of images. Discretisation. Degradation of digital images. Image transformations: continuous Fourier transform, discrete Fourier transform, image pyramids, wavelet transformation. Colour perception and colour spaces. Image compression. Image interpolation. Image enhancement: point operations, linear filters, wavelet, median, M-smoothers, morphological filters, discrete variational methods, Fourier methods, and deconvolution. Image feature extraction: edges, edges in multi-channel images, and corners. Texture analysis. Image segmentation: classical method, optimisation method. Sequence image analysis: local method, variational method. 3D reconstruction: camera geometry, stereo, shape-from-shading. Object recognition: invariants, eigenspace methods.		
1.5. Types of classes	☒ lectures ☐ seminars and workshops ☐ auditory exercises ☒ distance learning ☐ field work	☐ individual exercises ☐ multimedia and network ☒ laboratory exercises ☐ design exercises ☐ working with a supervisor ☐ other _____

1.6. Comments	

1.7. Student obligations

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.8. Monitoring and assessment of student work

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.9. Assessment and evaluation of student work during classes and in the final exam

STUDENT ACTIVITY	ECTS	LEARNING OUTCOME	TEACHING METHOD	ASSESSMENT METHOD	POINTS	
					Min	Max
Attendance at lectures (L), auditory exercises (AE), laboratory exercises (LE), design exercises (DE)	2	1	Lectures (L), laboratory exercises (LE), design exercises (CE)	Attendance tracking. Minimum attendance percentage: 70%.	0	10
Problem-solving exercises	1	3,4,5	Revision exams (written exam)	Evaluation	10	20
Preparation for laboratory exercises (LE), results analysis, report writing	2	3,4,5,6	Laboratory exercises (LE)	Preparation for LE, LE supervision, LE report assessment	10	20
Preparing for an oral exam and oral exam	1	1,2,3,6	Oral exam	Evaluation	25	50

1.10. Obligatory literature (at the time of submitting a study programme proposal)

1. R.C.G. Gonzalez, R. E. Woods: Digital Image Processing

1.11. Recommended additional literature (at the time of submitting a study programme proposal)

1. E. Trucco, A. Verri: Introductory Techniques for 3-D Computer Vision
2. J. Bigun: Vision with Direction

1.12. Number of obligatory literature copies in relation to the number of students currently taking the course

Title	*Number of copies*	*Number of students*
R.C.G. Gonzalez, R. E. Woods: Digital Image Processing	1	20

1.13. Quality assurance methods ensuring the acquisition of knowledge, skills and competences

Conducting university questionnaires about teachers (student-teacher relationship, transparency of assessment criteria, motivation for teaching, teaching clarity, etc.). Conducting Faculty surveys of courses (upon passing the exam, student self-assessment of the adopted learning outcomes and student workload in relation to the number of ECTS credits allocated to activities and courses as a whole).

General information

Lead instructor(s)	**Emmanuel Karlo Nyarko, PhD, Associate Professor**	
Course title	Optimisation Methods	
Study programme	Graduate university study programme in Computer Engineering	
Course status	Compulsory on module: Artificial Intelligence and Robotics Elective on modules: Computer Engineering, Software Engineering and Data Science	
Year of study	1.	
ECTS credits and teaching methods	ECTS credits	6
	Number of classes (lectures + auditory exercises + laboratory exercises + design exercises + seminars)	(30+0+30+0+0)

1. COURSE DESCRIPTION

1.1. Course objectives

Familiarity with local and global optimisation methods. Formulation and application of the optimisation problem model. Acquisition of relevant skills using modern libraries and application frameworks for optimisation

1.2. Course enrolment requirements

Requirements for enrolment in the first year of study fulfilled.

1.3. Expected learning outcomes

1. Recognise and analyse a given problem as an optimisation problem
2. Relate a given problem to the same or similar problems that have already been solved
3. Propose a suitable optimisation method for solving a given optimisation problem
4. Adapt the chosen optimisation method to the optimisation problem
5. Evaluate different metaheuristic optimisation methods

1.4. Course content

Basic classification of optimisation methods. Deterministic and heuristic approaches. The concept of the criterion function. Linear and nonlinear least squares problems. Direct search algorithms (Hooke-Jeeves method; Nelder-Mead simplex method). Gradient methods (gradient descent; Newton's method; Levenberg-Marquadt algorithm). Stopping criteria. Convex optimisation problems. Metaheuristic algorithms (evolutionary algorithms; particle swarm algorithm). Multi-objective optimisation. Pareto optimality. Hybrid optimisation methods.

1.5. Types of classes	☒ lectures ☐ seminars and workshops ☐ auditory exercises ☒ distance learning ☐ field work	☒ individual exercises ☐ multimedia and network ☒ laboratory exercises ☐ design exercises ☐ working with a supervisor ☐ other _____
1.6. Comments		

1.7. Student obligations

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.8. Monitoring and assessment of student work

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.9. Assessment and evaluation of student work during classes and in the final exam

STUDENT ACTIVITY	ECTS	LEARNING OUTCOME	TEACHING METHOD	ASSESSMENT METHOD	POINTS	
					Min	Max
Attendance at lectures (L), laboratory exercises (LE)	2	1, 2, 3, 4, 5	Lectures (L), laboratory exercises (LE)	Attendance tracking. Minimum attendance percentage: 70%.	7	10
Preparation for laboratory exercises (LE), results analysis, report writing	1.5	2, 3, 4, 5	Laboratory exercises (LE)	Preparation for LE, LE supervision, LE report assessment	15	30
Solving project task	1	2, 3, 4, 5	Individual exercises	Verification of the solution of the project task, presentation of the solution	0	25
Preparing for oral exam and oral exam	1.5	1, 2, 3, 4, 5	Oral exam	Evaluation	18	35

1.10. Obligatory literature (at the time of submitting a study programme proposal)

1. S. S. Rao, Engineering Optimization: Theory and Practice, 5th Edition, Wiley, 2019.
2. S. Luke, Essentials of Metaheuristics, Lulu, 2nd Edition, 2013. (Available for free at: http://cs.gmu.edu/~sean/book/metaheuristics/)

1.11. Recommended additional literature (at the time of submitting a study programme proposal)

1. X.-S. Yang, Engineering Optimization: An Introduction with Metaheuristic Applications, Wiley, 2014.

1.12. Number of obligatory literature copies in relation to the number of students currently taking the course

Title	*Number of copies*	*Number of students*
S. S. Rao, Engineering Optimization: Theory and Practice, 5th Edition, Wiley, 2019	0	
X.-S. Yang, Engineering Optimization: An Introduction with Metaheuristic Applications, Wiley, 2014.	0	

1.13. Quality assurance methods ensuring the acquisition of knowledge, skills and competences

Conducting university questionnaires about teachers (student-teacher relationship, transparency of assessment criteria, motivation for teaching, teaching clarity, etc.). Conducting Faculty surveys of courses (upon passing the exam, student self-assessment of the adopted learning outcomes and student workload in relation to the number of ECTS credits allocated to activities and courses as a whole).

General information

Lecturer(s)	**Damir Blažević, Phd, Full Professor**	
Course title	Software Quality Assurance	
Study programme	Graduate university study programme in Computer Engineering	
Course status	Compulsory at Computer Engineering Elective at Software Engineering Elective at Data Science	
Year of study	1. on module:Computer Engineering 2. on modulesSoftware Engineering and Data Science	
ECTS credits and teaching methods	ECTS credits	7
	Number of classes (lectures + auditory exercises + laboratory exercises + design exercises + seminars)	30+(15+15+0)+O

1. COURSE DESCRIPTION

1.1. Course objectives

Master the ways to determine quality, quality assurance processes and life cycle management of program support development. Become familiar with methods and techniques for managing development, planning, implementation, testing and retirement of software support.

1.2. Course enrolment requirements

Requirements for enrolment in the first or second year of study fulfilled.

1.3. Expected learning outcomes

1. Distinguish and evaluate ways of determining the quality of computer support (software)
2. Select and apply existing standards for software development in a specific case.
3. Assess the complexity of the computer support project and determine the necessary resources.
4. Organize, lead and participate in the computer support team.
5. Organize and plan computer support testing.
6. Participate in procedures and perform computer support testing.

1.4. Course content

An overview of the development of software engineering and the need for software quality assurance. Recognizing the emergence of a software crisis. Measures to mitigate the crisis. Introduction and management of software quality assurance program. Software development life cycle. Life cycle models. Life cycle management. Software process maturity levels. Measures and procedures for improving the software development life cycle. Software testing.

1.5. Types of classes	☒ lectures ☐ seminars and workshops ☒ auditory exercises ☒ distance learning ☐ field work	☐ individual exercises ☐ multimedia and network ☒ laboratory exercises ☐ design exercises ☐ working with a supervisor ☐ other _____
1.6. Comments		

1.7. Student obligations

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.8. Monitoring and assessment of student work

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.9. Assessment and evaluation of student work during classes and in the final exam

STUDENT ACTIVITY	ECTS	LEARNING OUTCOME	TEACHING METHOD	ASSESSMENT METHOD	POINTS	
					Min	Max
Attendance at lectures (L), auditory exercises (AE), laboratory exercises (LE), design exercises (DE)	2	1,2,3,4,5,6	Lectures (L), laboratory exercises (LE), design exercises (CE)	Attendance tracking. Minimum attendance percentage: 70%.	0	10
Problem-solving exercises	1,5	4,5	Revision exams (written exam)	Evaluation	8	20
Preparation for laboratory exercises (LE), results analysis, report writing	1,5	3,4,5	Laboratory exercises (LE)	Preparation for LE, LE supervision, LE report assessment	25	50
Preparing for an oral exam and oral exam	2	1,2,3,4,5	Oral exam	Evaluation	6	20

1.10. Obligatory literature (at the time of submitting a study programme proposal)

Roger S. Pressman, Software Engineering, McGraw – Hill Higher Education, 2001.
Ian Sommerville, Software engineering, Addison Wesley, 2011.
Claude Y. Laporte, Alain April, Software Quality Assurance, IEEE Press, 2018.

1.11. Recommended additional literature (at the time of submitting a study programme proposal)

K. Naik, P. Tripathy, Software Testing and Quality Assurance, John Wiley & Sons, 2008.

G. Gordon Schulmeyer, Handbook of Software Quality Assurance, Artech House, 2008.

1.12. Number of obligatory literature copies in relation to the number of students currently taking the course

Title	*Number of copies*	*Number of students*
Roger S. Pressman, Software Engineering, McGraw – Hill Higher Education, 2001.		20
Ian Sommerville, Software engineering, Addison Wesley, 2011.		20
Claude Y. Laporte, Alain April, Software Quality Assurance, IEEE Press, 2018.		20

1.13. Quality assurance methods ensuring the acquisition of knowledge, skills and competences

Conducting university questionnaires about teachers (student-teacher relationship, transparency of assessment criteria, motivation for teaching, teaching clarity, etc.). Conducting Faculty surveys of courses (upon passing the exam, student self-assessment of the adopted learning outcomes and student workload in relation to the number of ECTS credits allocated to activities and courses as a whole).

General information		
Lead instructor(s)	**Emmanuel Karlo Nyarko, PhD, Associate Professor** **Mario Lovrić, PhD, Assistant Professor**	
Course title	Basics of Bioinformatics	
Study programme	Graduate university study programme in Computer Engineering	
Course status	Elective on modules: Artificial Intelligence and Robotics and Data Science	
Year of study	2.	
ECTS credits and teaching methods	ECTS credits	5
	Number of classes (lectures + auditory exercises + laboratory exercises + design exercises + seminars)	(30+0+30+0+0)

1. COURSE DESCRIPTION		
1.1. Course objectives		
Introduction to methods and algorithms used in bioinformatics and corresponding applications. Solving problems in the field of supervised and unsupervised learning in the domain of biological and chemical sciences as well as biomedicine. Acquisition of relevant skills in working with modern libraries and application frameworks for bio- and chemo-informatics.		
1.2. Course enrolment requirements		
Requirements for enrolment in the second year of study fulfilled.		
1.3. Expected learning outcomes		
1. Explain basic terms from biology and chemistry. 2. Understand how the most important bioinformatic tools work. 3. Explain data sources and methods for collecting chemical and biological data. 4. Assess the suitability of a particular bio/cheminformatics algorithm for a given problem. 5. Use modern application frameworks for chemical and biological data processing and modelling. 6. Apply algorithms to generate chemical and biological feature representations. 7. Apply machine learning algorithms in solving chemical and biological problems.		
1.4. Course content		
Introduction to the field of chemo- and bioinformatics. Algorithms for processing chemical and biological data such as molecules and protein structures as well as genetic sequences. Preparation of structures, proteins, sequences for modelling toxicity, biological and chemical activity and interaction with drugs. Conversion of structures into representations for machine learning. Processing and preparation of genetic sequences for statistical processing. Introduction to "omics" research, measurement methods and applications. Various applications of machine learning and examples in the chemical and biological sciences.		
1.5. Types of classes	☒ lectures ☐ seminars and workshops ☐ auditory exercises ☒ distance learning ☐ field work	☒ individual exercises ☐ multimedia and network ☒ laboratory exercises ☐ design exercises ☐ working with a supervisor ☐ other _____
1.6. Comments		

1.7. Student obligations

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.8. Monitoring and assessment of student work

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.9. Assessment and evaluation of student work during classes and in the final exam

STUDENT ACTIVITY	ECTS	LEARNING OUTCOME	TEACHING METHOD	ASSESSMENT METHOD	POINTS	
					Min	Max
Attendance at lectures (L), laboratory exercises (LE)	2	1,2,3,4,6,7	Lectures (L), laboratory exercises (LE)	Attendance tracking. Minimum attendance percentage: 70%.	7	10
Preparation for laboratory exercises (LE), results analysis, report writing	1	5,6,7	Laboratory exercises (LE)	Preparation for LE, LE supervision, LE report assessment	15	30
Solving project task	0.5	5,6,7	Individual exercises	Verification of the solution of the project task, presentation of the solution	0	25
Preparing for oral exam and oral exam	1.5	1,2,3,5,6,7	Oral exam	Evaluation	18	35

1.10. Obligatory literature (at the time of submitting a study programme proposal)

1. A. Varnek, Tutorials in Chemoinformatics, Wiley, 1st edition, 2017

1.11. Recommended additional literature (at the time of submitting a study programme proposal)

1. K. Youens-Clark, Mastering Python for Bioinformatics, O'Reilly Media, 2021
2. V. Buffalo, Bioinformatics Data Skills, O'Reilly Media, 2015

1.12. Number of obligatory literature copies in relation to the number of students currently taking the course

Title	*Number of copies*	*Number of students*
A. Varnek, Tutorials in Chemoinformatics, Wiley, 1st edition, 2017	0	
K. Youens-Clark, Mastering Python for Bioinformatics, O'Reilly Media, 2021	0	
V. Buffalo, Bioinformatics Data Skills, O'Reilly Media, 2015	0	

1.13. Quality assurance methods ensuring the acquisition of knowledge, skills and competences

Conducting university questionnaires about teachers (student-teacher relationship, transparency of assessment criteria, motivation for teaching, teaching clarity, etc.). Conducting Faculty surveys of courses (upon passing the exam, student self-assessment of the adopted learning outcomes and student workload in relation to the number of ECTS credits allocated to activities and courses as a whole).

General information

Lead instructor(s)	**Robert Cupec, PhD, Full Professor with Tenure**	
Course title	Basics of Robotics	
Study programme	Graduate university study programme in Computer Engineering	
Course status	Compulsory on module: Artificial Intelligence and Robotics Elective on modules: Computer Engineering, Software Engineering and Data Science	
Year of study	1.	
ECTS credits and teaching methods	ECTS credits	5
	Number of classes (lectures + auditory exercises + laboratory exercises + design exercises + seminars)	30+(0+15+15)+0

1. COURSE DESCRIPTION

1.1. Course objectives

To provide basic knowledge in the field of robotics: basic configurations and applications of robots, forward and inverse kinematics, dynamic model of a robot manipulator, path and trajectory planning. Students should learn how to describe spatial relationships mathematically. To introduce students with the sensors and actuators used in robotics. Enable students to understand and apply methods from the field of robotics for the realization of control software for robot manipulators.

1.2. Course enrolment requirements

Requirements for enrolment in the first year of study fulfilled.

1.3. Expected learning outcomes

1. Formulate the kinematic model of a robotic manipulator based on its mechanical parameters using the Denavit-Hartenberg method
2. Create a computer program for robot tool positioning by solving the inverse kinematics problem for a 6-axis robot manipulator with revolute joints, where the last three axes intersect at one point
3. Create a computer program to generate trajectories for a robot manipulator
4. Implement a digital controller for controlling the robot joints
5. List the types of motors and sensors used in robotics and explain the basic principles of their use
6. Create a program for performing robot operations

1.4. Course content

Introduction to basic concepts in robotics: basic notions, basic robot configurations and applications. Mathematical description of spatial relationships. Forward and inverse kinematics of robotic manipulators. Denavit-Hartenberg convention. Differential kinematics. Trajectory planning. Dynamic model of robotic manipulator. Joint control. Sensors and actuators used in robotics. Basics of robot programming languages.

1.5. Types of classes	☒ lectures ☐ seminars and workshops ☐ auditory exercises ☒ distance learning ☐ field work	☒ individual exercises ☐ multimedia and network ☒ laboratory exercises ☒ design exercises ☐ working with a supervisor ☐ other _____
1.6. Comments		

1.7. Student obligations

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.8. Monitoring and assessment of student work

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.9. Assessment and evaluation of student work during classes and in the final exam

STUDENT ACTIVITY	ECTS	LEARNING OUTCOME	TEACHING METHOD	ASSESSMENT METHOD	POINTS	
					Min	Max
Attendance at lectures (L), laboratory exercises (LE)	2	1, 2, 3, 4, 5, 6	Lectures (L), laboratory exercises (LE), design exercises (DE)	Attendance tracking. Minimum attendance percentage: 70%.	7	10
Preparation for laboratory exercises (LE), results analysis, report writing	1	1, 2, 3, 4, 6	Laboratory exercises (LE)	Preparation for LE, LE supervision, LE report assessment	10	30
Preparing for an oral exam	1	3, 4, 5, 6	Oral exam	Evaluation	15	30
Problem-solving exercises set during design exercises (DE)	1	1, 2, 3, 4, 6	Design exercises (DE)	Evaluation	10	30

1.10. Obligatory literature (at the time of submitting a study programme proposal)

1. Z. Kovačić, S. Bogdan, V. Krajči, Osnove robotike
2. R. Cupec, Osnove robotike – nastavni materijali

1.11. Recommended additional literature (at the time of submitting a study programme proposal)

1. J. J. Craig, Introduction to Robotics: Mechanics and Control

1.12. Number of obligatory literature copies in relation to the number of students currently taking the course

Title	*Number of copies*	*Number of students*
Z. Kovačić, S. Bogdan, V. Krajči, Osnove robotike	5	16

1.13. Quality assurance methods ensuring the acquisition of knowledge, skills and competences

Conducting university questionnaires about teachers (student-teacher relationship, transparency of assessment criteria, motivation for teaching, teaching clarity, etc.). Conducting Faculty surveys of courses (upon passing the exam, student self-assessment of the adopted learning outcomes and student workload in relation to the number of ECTS credits allocated to activities and courses as a whole).

General information		
Lead instructor(s)	**Zdravko Krpić, PhD, Associate Professor** **Tomislav Matić, PhD, Associate Professor**	
Course title	Parallel Programming	
Study programme	Graduate university study programme in Computer Engineering	
Course status	Compulsory on module: Software Engineering	
Year of study	2.	
ECTS credits and teaching methods	ECTS credits	6
	Number of classes (lectures + auditory exercises + laboratory exercises + design exercises + seminars)	30+(0+30+0)+0

1. COURSE DESCRIPTION		
1.1. Course objectives		
This course aims to equip students with the skills to independently develop parallel algorithms for various computing architectures. It covers fundamental models and architectures of modern parallel computer systems, introduces basic models and paradigms of parallel programs, and provides insights into problem decomposition for parallel execution. Students will learn to choose appropriate computing resources and apply parallel programming paradigms such as MPI, OpenMP, CUDA, and OpenCL to practical software		
1.2. Course enrolment requirements		
Requirements for enrolment in the second year of study fulfilled.		
1.3. Expected learning outcomes		
1. Identify models of parallel computer systems and parallel programming 2. Distinguish between different high-performance computer architectures using Flynn's taxonomy 3. Explain the features of shared and shared memory architectures 4. Design a parallel algorithm for shared and shared memory systems 5. Create and implement a scalable parallel algorithm for different HPC architectures 6. Evaluate the performance of parallel implementation of algorithms		
1.4. Course content		
Shared and distributed memory computing systems. Multicore and Manycore systems. Parallel accelerators. SIMD instructions, and their application in CISC and ARM architectures. MPI and OpenMP. Data and functional decomposition. Parallel implementations of known algorithms. Graphics accelerators - GPGPU programming. Nvidia CUDA platform. CUDA programming model. Memory structures of the CUDA model. Comparison with the OpenCL programming model. Advantages and disadvantages of GPGPU platforms. Examples of application of GPGPU platforms for parallel processing.		
1.5. Types of classes	☒ lectures ☐ seminars and workshops ☐ auditory exercises ☒ distance learning ☐ field work	☐ individual exercises ☐ multimedia and network ☒ laboratory exercises ☐ design exercises ☐ working with a supervisor ☐ other _____
1.6. Comments		

1.7. Student obligations

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.8. Monitoring and assessment of student work

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.9. Assessment and evaluation of student work during classes and in the final exam

STUDENT ACTIVITY	ECTS	LEARNING OUTCOME	TEACHING METHOD	ASSESSMENT METHOD	POINTS	
					Min	Max
Attendance at lectures (L), laboratory exercises (LE)	2	1, 2, 3, 4, 5, 6	Lectures (L), laboratory exercises (LE)	Attendance tracking. Minimum attendance percentage: 70%.	7	10
Problem-solving exercises	1	4, 5, 6	Revision exams (written exam)	Evaluation	20	40
Preparation for laboratory exercises (LE), results analysis, report writing	2	1, 2, 4, 5, 6	Laboratory exercises (LE)	Preparation for LE, LE supervision, LE report assessment	10	20
Preparing for an oral exam and oral exam	1	1, 2, 3	Oral exam	Evaluation	15	30

1.10. Obligatory literature (at the time of submitting a study programme proposal)

1. Thomas Rauber, Gudula Rünger: Parallel Programming for Multicore and Cluster Systems (second edition), Springer, 2013.
2. Shane Cook: CUDA Programming: A Developer's Guide to Parallel Computing with GPUs (Applications of GPU Computing) 1st Edition, Morgan Kaufmann, 2012.
3. Aaftab Munshi: OpenCL Programming Guide 1st Edition, Addison-Wesley Professional, 2011.

1.11. Recommended additional literature (at the time of submitting a study programme proposal)

1.12. Number of obligatory literature copies in relation to the number of students currently taking the course

Title	*Number of copies*	*Number of students*
Parallel Programming for Multicore and Cluster Systems	1	
CUDA Programming: A Developer's Guide to Parallel Computing with GPUs (Applications of GPU Computing)	1	
OpenCL Programming Guide 1st Edition	1	

1.13. Quality assurance methods ensuring the acquisition of knowledge, skills and competences

Conducting university questionnaires about teachers (student-teacher relationship, transparency of assessment criteria, motivation for teaching, teaching clarity, etc.). Conducting Faculty surveys of courses (upon passing the exam, student self-assessment of the adopted learning outcomes and student workload in relation to the number of ECTS credits allocated to activities and courses as a whole).

General information

Lead instructor(s)	**Tomislav Matić, PhD, Associate professor**	
Course title	Computer System Reliability and Diagnostics	
Study programme	Graduate university study programme in Computer Engineering	
Course status	Compulsory on module: Computer Engineering Elective on module: Software Engineering	
Year of study	2.	
ECTS credits and teaching methods	ECTS credits	7
	Number of classes (lectures + auditory exercises + laboratory exercises + design exercises + seminars)	45+(15+15+0)+0

2. COURSE DESCRIPTION

1.1. Course objectives

To present students with theoretical and practical knowledge in the field of reliability and diagnostics of electronic components, digital circuits, computers and systems.

1.2. Course enrolment requirements

Requirements for enrolment in the second year of study fulfilled.

1.3. Expected learning outcomes

1. explain concepts related to computer system reliability
2. analyse reliability models and methods for increasing reliability
3. compare different methods for fault detection
4. management of system design activities for predetermined reliability and fault tolerance
5. evaluate different procedures of static and dynamic redundancy of computer systems
6. evaluate hardware and software reliability parameters
7. develop and apply hardware and software reliability models.

1.4. Course content

Introduction and historical development. Failures, faults and errors in computer systems: sources and types. Failure models and failure distribution. Basic parameters and features of system reliability, availability and maintainability. Reliability of components, circuits and systems. Increasing reliability. Redundancy and fault avoidance methods. Fault detection, self-diagnostic systems. Reliability of program support and models. Methods of specification and evaluation of computer systems, verification and validation.

1.5. Types of classes	☒ lectures ☐ seminars and workshops ☒ auditory exercises ☒ distance learning ☐ field work	☐ individual exercises ☐ multimedia and network ☒ laboratory exercises ☐ design exercises ☐ working with a supervisor ☐ other _____
1.6. Comments		

1.7. Student obligations

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.8. Monitoring and assessment of student work

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.9. Assessment and evaluation of student work during classes and in the final exam

STUDENT ACTIVITY	ECTS	LEARNING OUTCOME	TEACHING METHOD	ASSESSMENT METHOD	POINTS	
					Min	Max
Attendance at lectures (L), auditory exercises (AE), laboratory exercises (LE), design exercises (DE)	1	1, 2, 3, 4, 5, 6	Lectures (L), laboratory exercises (LE), design exercises (CE)	Attendance tracking. Minimum attendance percentage: 70%.	7	10
Problem-solving exercises	2,5	3, 4, 5, 6, 7	Revision exams (written exam)	Evaluation	20	40
Preparation for laboratory exercises (LE), results analysis, report writing	1,5	4, 5, 6, 7	Laboratory exercises (LE)	Preparation for LE, LE supervision, LE report assessment	10	20
Preparing for an oral exam and oral exam	2	1, 2, 3, 4, 5, 6	Oral exam	Evaluation	15	30

1.10. Obligatory literature (at the time of submitting a study programme proposal)

1. Kapur K. C.; Pecht, M. Reliability Engineering
2. Pezzé; M; Young, M. Software Testing and Analysis: Process, Principles, and Techniques

1.11. Recommended additional literature (at the time of submitting a study programme proposal)

1. B. W. Johnson. Design and Analysis of Fault-Tolerant Digital System
2. A. C. Brombacher. Reliability by Design, CAE Techniques for Electronic Components and Systems
3. H. Pham, ed. Handbook of Reliability Engineering
4. D. Siewiorek, E. Swarz. The Theory and Practice of Reliable System Design
5. M. A. Breuer, A. D. Friedman. Diagnosis & Reliable Design of Digital Systems
6. P. P. O Connor, A. Kleyner. Practical Reliability Engineering

1.12. Number of obligatory literature copies in relation to the number of students currently taking the course

Title	*Number of copies*	*Number of students*
Kapur K. C.; Pecht, M. Reliability Engineering	1	30
Pezzé; M; Young, M. Software Testing and Analysis: Process, Principles, and Techniques	1	30

1.13. Quality assurance methods ensuring the acquisition of knowledge, skills and competences
Conducting university questionnaires about teachers (student-teacher relationship, transparency of assessment criteria, motivation for teaching, teaching clarity, etc.). Conducting Faculty surveys of courses (upon passing the exam, student self-assessment of the adopted learning outcomes and student workload in relation to the number of ECTS credits allocated to activities and courses as a whole).

General information

Lead instructor(s)	**Damir Filko, PhD, Associate Professor**	
Course title	Robot Programming and Simulation	
Study programme	Graduate university study programme in Computer Engineering	
Course status	Compulsory on module: Artificial Intelligence and Robotics Elective on module: Computer Engineering	
Year of study	1.	
ECTS credits and teaching methods	ECTS credits	6
	Number of classes (lectures + auditory exercises + laboratory exercises + design exercises + seminars)	30+0+30+0+0

1. COURSE DESCRIPTION

1.1. Course objectives

Inform students on the capabilities of the Robot Operating System (ROS) and associated tools. Show how ROS nodes are created and how they can communicate through the ROS infrastructure. Explain how mobile robots and robot manipulators are modelled and simulated. Demonstrate the capabilities of the rosbag package for recording data. Present the possibilities of data visualization.

1.2. Course enrolment requirements

Requirements for enrolment in the first year of study fulfilled.

1.3. Expected learning outcomes

1. Employ Linux operating system.
2. Program in Python and C++ programming languages.
3. Explain the structure of the Robot Operating System and its role in robot control.
4. Implement ROS nodes and facilitate their communication.
5. Model and simulate mobile robots and robot manipulators.
6. Apply data recording and data visualization tools.

1.4. Course content

Basic terminology and structure of the Robot Operating System. Basics of Linux operating system. Programming languages Python and C++. ROS node. ROS topics, services and actions. Recording data and application of the rosbag tool. Modelling of a mobile robot platform. Modelling of a robot manipulator. Simulators (Gazebo, Stage, Webots). Data visualization in ROS.

1.5. Types of classes	☒ lectures ☐ seminars and workshops ☐ auditory exercises ☒ distance learning ☐ field work	☒ individual exercises ☐ multimedia and network ☒ laboratory exercises ☐ design exercises ☐ working with a supervisor ☐ other _____
1.6. Comments		

1.7. Student obligations

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.8. Monitoring and assessment of student work

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.9. Assessment and evaluation of student work during classes and in the final exam

STUDENT ACTIVITY	ECTS	LEARNING OUTCOME	TEACHING METHOD	ASSESSMENT METHOD	POINTS	
					Min	Max
Attendance at lectures (L), laboratory exercises (LE),	2	1, 2, 3, 4, 5, 6	Lectures (L), laboratory exercises (LE),	Attendance tracking. Minimum attendance percentage: 70%.	7	10
Problem-solving of individual projects	2	1, 2, 3, 4, 5, 6	Developing projects	Evaluation and presentation of results	0	30
Preparation for laboratory exercises (LE), results analysis, report writing	1	1,2, 3, 4, 5, 6	Laboratory exercises (LE)	Preparation for LE, LE supervision, LE report assessment	10	30
Preparing for an oral exam and oral exam	1	1, 2, 3, 4, 5, 6	Oral exam	Evaluation	18	30

1.10. Obligatory literature (at the time of submitting a study programme proposal)

1. Quigley, M., Gerkey B., Smart, W.D., Programming Robots with ROS. O'Reilly Media, 2015.

1.11. Recommended additional literature (at the time of submitting a study programme proposal)

1. C. Schroder Linux Cookbook O Reilly, New York, 2004.
2. Lutz, M. Learning Python, 5th Edition. O'Reilly Media, 2013.
3. Fairchild, C., Harman, T.L., ROS Robotics by Example, 2nd edition, Packt Publishing, 2017.
4. ROS Wiki (http://wiki.ros.org/)

1.12. Number of obligatory literature copies in relation to the number of students currently taking the course

Title	*Number of copies*	*Number of students*
1. Quigley, M., Gerkey B., Smart, W.D., Programming Robots with ROS. O'Reilly Media, 2015.	1	16

1.13. Quality assurance methods ensuring the acquisition of knowledge, skills and competences

Conducting university questionnaires about teachers (student-teacher relationship, transparency of assessment criteria, motivation for teaching, teaching clarity, etc.). Conducting Faculty surveys of courses (upon passing the exam, student self-assessment of the adopted learning outcomes and student workload in relation to the number of ECTS credits allocated to activities and courses as a whole).

General information

Lecturer(s)	**Damir Blažević, Phd, Full Professor**	
Course title	Computer Networks Design	
Study programme	Graduate university study programme in Computer Engineering	
Course status	Compulsory on module: Computer Engineering	
Year of study	2.	
ECTS credits and teaching methods	ECTS credits	5
	Number of classes (lectures + auditory exercises + laboratory exercises + design exercises + seminars)	30+(0+30+0)+O

1. COURSE DESCRIPTION

1.1. Course objectives

To provide participants with practical and theoretical knowledge in the field of designing computer networks. Through lectures and exercises, train them to analyze user needs, design, project, configure, implement, analyze and eliminate irregularities in the operation of the computer network. To acquaint the participants with legal and technical regulations in the field of design and construction. Special emphasis should be placed on the creation of project documentation, cost sheets, configuration files of network devices (special purpose computers), their implementation and maintenance. Acquaint the participants with a practical approach to the implementation of service quality in a specific network environment.

1.2. Course enrolment requirements

Requirements for enrolment in the second year of study fulfilled.

1.3. Expected learning outcomes

1. Analyze requirements and distinguish and evaluate specifics in the construction and management of modern computer communication networks.
2. Design and evaluate a simple computer communication network based on current techniques, methods and knowledge in computer communication.
3. Plan and design a local computer network, choose and justify the selection of passive and active network equipment.
4. Adjust the network device to work according to the given conditions, perform implementation on the network device and analyze the operation of the device.
5. Categorize network traffic types, create and test network traffic filtering lists, and propose QoS settings.

1.4. Course content

Introduction to legal and technical regulations related to the design of computer networks. Creating project documentation. Computer networks. Types and division of computer networks. Passive and active network devices. Computer hardware and software. Creation of configuration files for network nodes. Computer network design, equipment specification, construction and maintenance. Implementation of quality of service settings. Creation of access lists.

1.5. Types of classes	☒ lectures ☐ seminars and workshops ☐ auditory exercises ☒ distance learning ☐ field work	☐ individual exercises ☐ multimedia and network ☒ laboratory exercises ☐ design exercises ☐ working with a supervisor

		☐ other _____
1.6. Comments		

1.7. Student obligations

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.8. Monitoring and assessment of student work

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.9. Assessment and evaluation of student work during classes and in the final exam

STUDENT ACTIVITY	ECTS	LEARNING OUTCOME	TEACHING METHOD	ASSESSMENT METHOD	POINTS	
					Min	Max
Attendance at lectures (L), auditory exercises (AE), laboratory exercises (LE), design exercises (DE)	1	1,2,5	Lectures (L), laboratory exercises (LE), design exercises (CE)	Attendance tracking. Minimum attendance percentage: 70%.	4	10
Problem-solving exercises	1	3,4	Revision exams (written exam)	Evaluation	6	30
Preparation for laboratory exercises (LE), results analysis, report writing	1	1,2,3,4	Laboratory exercises (LE)	Preparation for LE, LE supervision, LE report assessment	10	20
Preparing for an oral exam and oral exam	2	1,2,5	Oral exam	Evaluation	10	40

1.10. Obligatory literature (at the time of submitting a study programme proposal)

N. Olifer, V.Olifer, Computer Networks, Wiley, 2005.
L. L. Peterson, B. S. Davie, Computer Networks: A System Approach, Elsiver Science, 2003.

1.11. Recommended additional literature (at the time of submitting a study programme proposal)

K. Naik, P. Tripathy, Software Testing and Quality Assurance, John Wiley & Sons, 2008.

G. Gordon Schulmeyer, Handbook of Software Quality Assurance, Artech House, 2008.

1.12. Number of obligatory literature copies in relation to the number of students currently taking the course

Title	*Number of copies*	*Number of students*
N. Olifer, V.Olifer, Computer Networks, Wiley, 2005.		20
L. L. Peterson, B. S. Davie, Computer Networks: A System Approach, Elsiver Science, 2003.		20

1.13. Quality assurance methods ensuring the acquisition of knowledge, skills and competences

Conducting university questionnaires about teachers (student-teacher relationship, transparency of assessment criteria, motivation for teaching, teaching clarity, etc.). Conducting Faculty surveys of courses (upon passing the exam, student self-assessment of the adopted learning outcomes and student workload in relation to the number of ECTS credits allocated to activities and courses as a whole).

General information		
Lead instructor(s)	**Dominika Crnjac Milić, PhD, Full Professor**	
Course title	Project Management	
Study programme	Graduate university study programme in Computer Engineering	
Course status	Compulsory on all modules	
Year of study	2.	
ECTS credits and teaching methods	ECTS credits	4
	Number of classes (lectures + auditory exercises + laboratory exercises + design exercises + seminars)	30+(15+0+0)+0

1. COURSE DESCRIPTION

1.1. Course objectives

To teach students about project management as a management process in which knowledge, skills, tools and techniques are applied to project activities in order to meet the requirements and needs of projects, but also to fulfil the strategic goals of a business organization. To encourage students to work in teams in such a way that, with the mentoring of the teacher, they jointly elaborate the content of the assigned project and its main goals. Teach them about the identification of the main activities on the project and about the work breakdown structure (WBS). To give directions related to the time planning of each individual activity and to determine the critical points and paths that could be used to solve the obstacles on the way to realization. Give them directions related to capacity planning, bottleneck detection and capacity balancing.
Provide knowledge related to cost determination, project profitability calculation and risk analysis.
Through the course, students will be taught about all phases of project planning, implementation and management.

1.2. Course enrolment requirements

Requirements for enrolment in the second year of study fulfilled.

1.3. Expected learning outcomes

1. specify, explain basic management functions, define and explain enterprise management through projects
2. specify and describe the basic managers skills important for management
   projects on the quality way
3. explain the strategic dimension of project management
4. explain the operational dimension of project management, analyse the phases of work on the project, develop the premises of the business plan for a specific project and to select and formulate the main objectives of the project and rank them
5. develop and evaluate the main activities of the project and the work breakdown structure (WBS - Work Breakdown Structure), analyse the totality of project tasks, and construct an budget for a some project as example
6. implement your own solution to the proposed project task (assess the capacities needed for project realization /determine bottlenecks, balance activities, determine costs and risks/; to make time plan for the realization of individual project tasks and identify their mutual relationships addiction; apply project management methods and techniques in the planning and implementation of specific projects from the field of study in a team environment; document all phases of project management in accordance with applicable standards; apply appropriate software solutions for project management)

1.4. Course content

Projects and project-oriented business (concept and essential characteristics of a project; basic distinction of projects; project life cycle; project-oriented business); concept and context of project management; development strategy of project management (phases of project management development; project management through strategic development of project business); design of an organization for project management (design of an organization for the management of one-off projects; organizational design for the management of project processes; organization and development of the project management system); strategic dimension of project management (initiation and activation of project implementation; planning and organization of project implementation logistics; evaluation and completion of project implementation); operational dimension of project management (management of project integration; management of project organization; management of implementation of the system of primary project objectives; project management control and management of project changes; development perspectives of project management).

1.5. Types of classes	☒ lectures ☐ seminars and workshops ☒ auditory exercises ☒ distance learning ☐ field work	☐ individual exercises ☐ multimedia and network ☐ laboratory exercises ☐ design exercises ☐ working with a supervisor ☐ other _____
1.6. Comments		

1.7. Student obligations

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.8. Monitoring and assessment of student work

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.9. Assessment and evaluation of student work during classes and in the final exam

STUDENT ACTIVITY	ECTS	LEARNING OUTCOME	TEACHING METHOD	ASSESSMENT METHOD	POINTS	
					Min	Max
Attendance at lectures (L), auditory exercises (AE), laboratory exercises (LE), design exercises (DE)	0.0	1,2,3,4,5,6	Lectures (L), laboratory exercises (LE), design exercises (CE)	Attendance tracking. Minimum attendance percentage: 70%.	0	0
Answering theoretical questions	2.0	1,2,3,4,5,6	Revision exams (written exam)	Evaluation	30	60
Oral presentation of project proposals with the help of prepared project documentation and ppt presentation	0.5	4,5,6	Oral presentation	Evaluation	5	10
Creation of project proposal documentation	0.5	4,5,6	Students, under the mentorship of professor, create	In accordance with the given instructions for	5	10

			documentation for a given project proposal	creating the documentation of the project proposal, the professor checks what has been written		
Creation of a ppt presentation and presentation of the topic of the seminar paper	0.5	4,5,6	According to the professor's instructions, students prepare the content of the presentation on the given topic of the seminar paper, while following the content of the previously written of work. Teamwork.	After listening to the presentation of the topic of the seminar work with the help of a ppt presentation, the professor awards points for a successfully completed activity	5	10
Creating a seminar paper	0.5	4,5,6	Studying the literature related to the given topic of the seminar paper and writing the seminar paper. Teamwork.	According to the instructions for writing the seminar paper, the content and written expression of the written form are evaluated.	5	10

1.10. Obligatory literature (at the time of submitting a study programme proposal)

1. Project Management, Mislav Ante Omazić, Stipe Baljkas, Sinergija, Zagreb, 2005.
2. Project Management Knowledge Guide (PMBOK Guide) - Fourth Edition, Project Management Institute, Global Standard, Mate d.o.o., Zagreb, 2011.
3. Project Management, Vlado Majstorović, Sveučilište u Mostaru, Mostar, 2010.

1.11. Recommended additional literature (at the time of submitting a study programme proposal)

1. A guide to project management from start to the end, Gregory M. Horine, DVA I DVA, Zagreb, 2009.

1.12. Number of obligatory literature copies in relation to the number of students currently taking the course

Title	*Number of copies*	*Number of students*
Project Management	1	
Project Management Knowledge Guide (PMBOK Guide) - Fourth Edition	1	
Project Management	15	
A Guide to Project Management From Start to the End	1	

1.13. Quality assurance methods ensuring the acquisition of knowledge, skills and competences

Conducting university questionnaires about teachers (student-teacher relationship, transparency of assessment criteria, motivation for teaching, teaching clarity, etc.). Conducting Faculty surveys of courses (upon passing the exam, student self-assessment of the adopted learning outcomes and student workload in relation to the number of ECTS credits allocated to activities and courses as a whole).

General information

Lead instructor(s)	**Goran Martinović, PhD, Full Professor with Tenure**	
Course title	Real-Time Computer Systems	
Study programme	Graduate university study programme in Computer Engineering	
Course status	Compulsory on modules: Computer Engineering, Software Engineering	
Year of study	2.	
ECTS credits and teaching methods	ECTS credits	7
	Number of classes (lectures + auditory exercises + laboratory exercises + design exercises + seminars)	45 + (0 + 30 + 0) + 0

1. COURSE DESCRIPTION

1.1. Course objectives

To explain to students timing, functional and other important limitations in the usage of current computer systems and software support in time-critical applications. Demonstrate the properties and use of suitable algorithms, methodologies, hardware, programming principles and development tools that enable the hardware and software design and analysis of embedded, distributed and scalable service systems for real-time operation.

1.2. Course enrolment requirements

Requirements for enrolment in the second year of study fulfilled.

1.3. Expected learning outcomes

1. Evaluate temporal, functional and other properties important for real-time computer systems
2. Evaluate the requirements of the environment and select suitable hardware and software properties of computer systems with the aim of modelling and creating real-time systems
3. Design the hardware and software implementation of real-time computer systems, applying defined hardware and software methodologies, algorithms and programming development environments
4. Design hardware and software solutions for real-time computer systems
5. Select and apply a suitable method of analysis of the given algorithm
6. Identify and solve problems related to the use of extremely distributed, heterogeneous and unreliable computer resources
7. Analyse and evaluate realized solutions in embedded, distributed, ubiquitous and cyber-physical systems

1.4. Course content

Computer systems according to time requirements. Functional and meta-functional requirements for the hardware and software part of the real-time systems. The concept of time, time bases and limitations in time measurement. System modelling: task, time and event driven systems, interrupts. Resource handling in distributed, heterogeneous and unreliable environments, scheduling algorithms, complexity of algorithms, analysis and evaluation metrics. Communicating and synchronizing. Real-time operating systems and their customization. Specialized programming systems of embedded computers. Required properties of software tools for the implementation of systems for time-critical applications. Access to system components from higher-level languages. Programming languages for real-time computer systems implementation. Worst case execution time analysis (WCET) of programming code. Interfacing of the system with the environment. Designing real-time systems: specification, design, analysis and application-specific testing. Embedded, distributed, ubiquitous and scalable real-time computing systems. Open and single-board hardware platforms and programming environments for system development. Autonomic computer systems. Internet of Things (IoT), cyber-physical systems and data analysis in time-critical applications. Analysis and improvements of the hardware and software parts of the system.

1.5. Types of classes	☒ lectures ☐ seminars and workshops ☐ auditory exercises ☒ distance learning ☐ field work	☒ individual exercises ☐ multimedia and network ☒ laboratory exercises ☐ design exercises ☐ working with a supervisor ☐ other _____
1.6. Comments		

1.7. Student obligations

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.8. Monitoring and assessment of student work

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.9. Assessment and evaluation of student work during classes and in the final exam

STUDENT ACTIVITY	ECTS	LEARNING OUTCOME	TEACHING METHOD	ASSESSMENT METHOD	POINTS	
					Min	Max
Attendance at lectures (L) and laboratory exercises (LE)	1.3	1,2,3,4,5,6,7	Lectures (L), laboratory exercises (LE)	Attendance tracking. Minimum attendance percentage: 70%.	3	6
Problem-solving, modelling, simulation and programming tasks	1.5	2,3,4,5,7	Control assessment or Written exam	Evaluation of answers and solutions	15	30
Preparation for laboratory exercises (LE), results analysis, report writing	1.2	2,3,4,5,7	Laboratory exercises (LE)	Evaluation of preparation for LE, LE supervision, LE report assessment	0	18
Project assignment	1.5	2,3,4,5,7	Lectures (L), laboratory exercises (LE) and individual work	Evaluation of the accuracy and completeness of the project assignment	3	6
Preparing for a written exam and written exam	0.5	1,2,3,4,5,6,7	Written exam	Evaluation of the written exam	5	10
Preparing for an oral exam and oral exam	1	1, 2, 3, 4, 5, 6	Oral exam	Evaluation of the oral exam	15	30

1.10. Obligatory literature (at the time of submitting a study programme proposal)

1. A. Baliyan, K.S. Kaswan, et al., Cyber Physical Systems: Concepts and Applications, Chapman & Hall, 2022.
2. A.S. Berger, Debugging Embedded and Real-Time Systems: The Art, Science, Technology, and Tools of Real-Time System Debugging (1st Ed.), Newnes, 2020.

3. H. Kopetz, W. Steiner, Real-Time Systems Design Principles for Distributed Embedded Applications (3rd Ed.), Springer, 2022.
4. C. Kormanyos, Real-Time C++: Efficient Object-Oriented and Template Microcontroller Programming (4th Ed.), Springer, 2021.
5. M.L. Pinedo, Scheduling: Theory, Algorithms, and Systems (5th Ed.), Springer, 2018.

1.11. Recommended additional literature (at the time of submitting a study programme proposal)

1. J. Cooling, Software Engineering for Real-Time Systems: A Software Engineering Perspective toward Designing Real-Time Systems (1st Ed.), Packt Publishing, 2019.
2. K. Erciyes, Distributed Real-Time Systems: Theory and Practice, Springer, 2019.
3. C. Koulamas, M.T. Lazarescu, Real-Time Sensor Networks and Systems for the Industrial IoT, Mdpi AG, 2020.
4. M. Mahrishi, K. Kant Hiran, G. Meena, P. Sharma, Machine Learning and Deep Learning in Real-Time Applications (1st Ed.), IGI Global, 2020.
5. R.L. Sites, Understanding Software Dynamics (1st Ed.), Addison-Wesley Professional, 2021.
6. K.C. Wang, Embedded and Real-Time Operating Systems (1st Ed.), Springer, 2017.

1.12. Number of obligatory literature copies in relation to the number of students currently taking the course

Title	*Number of copies*	*Number of students*
1. A. Baliyan, K.S. Kaswan, et al., Cyber Physical Systems: Concepts and Applications, Chapman & Hall, 2022.	3	60
2. A.S. Berger, Debugging Embedded and Real-Time Systems: The Art, Science, Technology, and Tools of Real-Time System Debugging (1st Ed.), Newnes, 2020.	3	60
3. H. Kopetz, W. Steiner, Real-Time Systems Design Principles for Distributed Embedded Applications (3rd Ed.), Springer, 2022.	3	60
4. C. Kormanyos, Real-Time C++: Efficient Object-Oriented and Template Microcontroller Programming (4th Ed.), Springer, 2021.	3	60
5. M.L. Pinedo, Scheduling: Theory, Algorithms, and Systems (5th Ed.), Springer, 2018.	3	60

1.13. Quality assurance methods ensuring the acquisition of knowledge, skills and competences

Conducting university questionnaires about teachers (student-teacher relationship, transparency of assessment criteria, motivation for teaching, teaching clarity, etc.). Conducting Faculty surveys of courses (upon passing the exam, student self-assessment of the adopted learning outcomes and student workload in relation to the number of ECTS credits allocated to activities and courses as a whole).

General information

Lead instructor(s)	**Robert Cupec, PhD, Full Professor with Tenure**	
Course title	Computer Control of Systems	
Study programme	Graduate university study programme in Computer Engineering	
Course status	Compulsory on module: Artificial Intelligence Elective on module: Computer Engineering	
Year of study	1.	
ECTS credits and teaching methods	ECTS credits	5
	Number of classes (lectures + auditory exercises + laboratory exercises + design exercises + seminars)	45+(0+15+15)+0

1. COURSE DESCRIPTION

1.1. Course objectives

To introduce the concept of automatic control and control theory. To teach students to formulate mathematical models of simple processes using differential equations, transfer functions and state space models. To teach students to program digital controllers. To introduce students to complex control structures. To teach students how to implement digital controllers in the form of a computer programme.

1.2. Course enrolment requirements

Requirements for enrolment in the first year of study fulfilled.

1.3. Expected learning outcomes

7. Describe the behaviour of simple dynamical systems using differential equations.
8. Explain the concept and basic principle of process identification.
9. Explain the purpose of the control system and its structure.
10. Explain the concept of control loop stability.
11. Design of a digital controller.
12. Application of control structures with multiple control loops.
13. Implementation of digital controllers in the form of computer programs.

1.4. Course content

Automatic control and its purpose. Basic structure and elements of a control loop. Digital control systems. Realisation of a control system. Process characteristics. Linearisation of the static characteristic. Dynamical system behaviour and mathematical description of the dynamical system behaviour. Description of linear and time-invariant systems in time domain and using Laplace transform. State space model. Introduction to process identification. Control loop and its properties. Static characteristics and accuracy of control loop. Control loop stability. Standard controller types. Concept of control loop synthesis. Some practical control loop synthesis methods. Root locus. Pole placement. Complex control structures. Realisation of digital controllers in the form of a computer program.

1.5. Types of classes	☒ lectures ☐ seminars and workshops ☐ auditory exercises ☒ distance learning ☐ field work	☐ individual exercises ☐ multimedia and network ☒ laboratory exercises ☒ design exercises ☐ working with a supervisor ☐ other _____

1.6. Comments	

1.7. Student obligations

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.8. Monitoring and assessment of student work

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.9. Assessment and evaluation of student work during classes and in the final exam

STUDENT ACTIVITY	ECTS	LEARNING OUTCOME	TEACHING METHOD	ASSESSMENT METHOD	POINTS	
					Min	Max
Attendance at lectures (L), laboratory exercises (LE)	2	1, 2, 3, 4, 5, 6, 7	Lectures (L), laboratory exercises (LE), design exercises (DE)	Attendance tracking. Minimum attendance percentage: 70%.	7	10
Preparation for laboratory exercises (LE), results analysis, report writing	1	1, 2, 5, 7	Laboratory exercises (LE)	Preparation for LE, LE supervision, LE report assessment	10	30
Preparing for an oral exam	1	1,2,3,4,5,6,7	Oral exam	Evaluation	10	30
Problem-solving exercises set during design exercises (DE)	1	1,2,5,7	Design exercises (DE)	Evaluation	8	30

1.10. Obligatory literature (at the time of submitting a study programme proposal)

1. N. Perić, Automatsko upravljanje - predavanja Zavodska skripta

1.11. Recommended additional literature (at the time of submitting a study programme proposal)

1. T. Šurina, Automatska regulacija

2. Z. Vukić, Lj. Kuljača, Automatsko upravljanje: analiza linearnih sustava

3. N. Perić, I. Petrović, Automatizacija postrojenja i procesa - predavanja, Zavodska skripta

4. J. Åström, B. Wittemark, Adaptive Control

5. R. Cupec, Diskretni sustavi upravljanja - nastavni materijali

6. R. Cupec, Sinteza digitalnog regulatora metodom postavljanja polova – nastavni materijali

1.12. Number of obligatory literature copies in relation to the number of students currently taking the course

Title	*Number of copies*	*Number of students*
N. Perić, Automatsko upravljanje - predavanja Zavodska skripta		

1.13. Quality assurance methods ensuring the acquisition of knowledge, skills and competences

Conducting university questionnaires about teachers (student-teacher relationship, transparency of assessment criteria, motivation for teaching, teaching clarity, etc.). Conducting Faculty surveys of courses (upon passing the exam, student self-assessment of the adopted learning outcomes and student workload in relation to the number of ECTS credits allocated to activities and courses as a whole).

General information		
Lead instructor(s)	**Goran Martinović, PhD, Full Professor with Tenure**	
Course title	Cloud Computing and Data Analysis	
Study programme	Graduate university study programme in Computer Engineering	
Course status	Compulsory on modules: Computer Engineering, Software Engineering and Data Science Elective on module: Artificial Intelligence and Robotics	
Year of study	2.	
ECTS credits and teaching methods	ECTS credits	
	Number of classes (lectures + auditory exercises + laboratory exercises + design exercises + seminars)	30 + 0 + 30 + 0 +0

1. COURSE DESCRIPTION		
1.1. Course objectives		
Explain architectures, models and operating principles of service systems and cloud computing. To acquaint students with the procedures for designing, implementation, testing and integrating software solutions for execution in the cloud computing environment on advanced architectures for data storage and processing. Analyse of requirements and apply data discovery and data analysis methods by using appropriate machine learning algorithms, statistical approaches and data analysis tools in cloud services environments.		
1.2. Course enrolment requirements		
Requirements for enrolment in the second year of study fulfilled.		
1.3. Expected learning outcomes		
1. Explain the issue of concurrency management and parallelism in data storage and processing 2. Select storage methods suitable for special types of data such as temporal, multimedia and spatial 3. Select and apply different techniques for processing data stored in advanced data storage systems 4. Identify and use knowledge extracted from big data 5. Explain the basic concepts of machine learning and identify the advantages and disadvantages of basic machine learning algorithms 6. Select and apply various classification and clustering algorithms 7. Evaluate and apply different machine learning methods in artificial intelligence (supervised, unsupervised, semi-supervised and reinforced learning) for a given problem		
1.4. Course content		
Service-based computing. Concurrency and parallelism in data storage and processing. Principles, role and possibilities of cloud computing. Cloud computing architecture and service management methods. Cloud computing service models. Defining the platform, infrastructure, application and display mode on examples. Transport formats for handling relational and non-relational data. Resource scalability, virtualization and cloud computing data storage. Design, implementation, testing and integration of software solutions and systems suitable for execution in cloud computing environment. Application containers. Storage and processing of data of different temporal and spatial characteristics. Big data. Data streams. Technologies for discovering, storing, handling and processing big data. Analysis of data streams. Technologies for data and data streams analysis. Application of machine learning (classification, grouping,) and statistical approaches in data analysis. Programming languages, tools, environments and technologies for data analysis in cloud computing.		
1.5. Types of classes	☒ lectures ☐ seminars and workshops	☒ individual exercises ☐ multimedia and network ☒ laboratory exercises

	☐ auditory exercises ☒ distance learning ☐ field work	☐ design exercises ☐ working with a supervisor ☐ other _____
1.6. Comments		

1.7. Student obligations

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.8. Monitoring and assessment of student work

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.9. Assessment and evaluation of student work during classes and in the final exam

STUDENT ACTIVITY	ECTS	LEARNING OUTCOME	TEACHING METHOD	ASSESSMENT METHOD	POINTS	
					Min	Max
Attendance at lectures (L) and laboratory exercises (LE)	1	1,2,3,4,5,6,7	Lectures (L) and laboratory exercises (LE)	Attendance tracking. Minimum attendance percentage: 70%.	3	6
Preparation for laboratory exercises (LE), results analysis, report writing	1.5	2,3,4,6,7	Laboratory exercises (LE)	Evaluation of preparation for LE, LE supervision, LE report assessment	12	24
Preparing for an oral exam and oral exam	1	1,2,4,5,7	Oral exam	Evaluation of answers	15	30
Solving of theoretical, problem and modelling exercises	1	5, 6	Written exam	Evaluation of solutions by written exam, LE preparation and reports	10	20
Project work	1	2,3,6,7	Written report of project work	Evaluation of project work	10	20

1.10. Obligatory literature (at the time of submitting a study programme proposal)

1. P.R. Chelliah, K.V.L. Sai Sampath, et al., Cloud-native Computing: Technologies and Tools towards Enterprise-scale Microservices-centric Applications, Wiley-IEEE Press, 2022.
2. M. Kleppmann, Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems (1st Ed.), O'Reilly Media, 2017.
3. H. Jiang, Machine Learning Fundamentals, Cambridge University Press, 2022.
4. L. Moroney, AI and Machine Learning for Coders: A Programmer's Guide to Artificial Intelligence, O'Reilly Media, 2022.

J. Reis, M. Housley, Fundamentals of Data Engineering: Plan and Build Robust Data Systems, O'Reilly Media, 2022.

1.11. Recommended additional literature (at the time of submitting a study programme proposal)

1. D. Comer, The Cloud Computing Book: The Future of Computing Explained (1st Ed.), Chapman and Hall/CRC, 2021.

2. EMC Education Services, Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data, Wiley, 2015.
3. F. Hueske, V. Kalavri, Stream Processing with Apache Flink, O'Reilly, 2019.
4. C. Korner, M. Alsdorf Mastering Azure Machine Learning: Execute Large-Scale End-to-End Machine Learning with Azure (2nd Ed.), Packt Publishing, 2022.
5. N. Marz, J. Warren, Big Data: Principles and Best Practices of Scalable Real-Time Data Systems (1st Ed.), Manning, 2015.

1.12. Number of obligatory literature copies in relation to the number of students currently taking the course

Title	*Number of copies*	*Number of students*
1. P.R. Chelliah, K.V.L. Sai Sampath, et al., Cloud-native Computing: Technologies and Tools towards Enterprise-scale Microservices-centric Applications, Wiley-IEEE Press, 2022.	3	80
2. M. Kleppmann, Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems (1st Ed.), O'Reilly Media, 2017.	3	80
3. H. Jiang, Machine Learning Fundamentals, Cambridge University Press, 2022.	3	80
4. L. Moroney, AI and Machine Learning for Coders: A Programmer's Guide to Artificial Intelligence, O'Reilly Media, 2022.	3	80
5. J. Reis, M. Housley, Fundamentals of Data Engineering: Plan and Build Robust Data Systems, O'Reilly Media, 2022.	3	80

1.13. Quality assurance methods ensuring the acquisition of knowledge, skills and competences

Conducting university questionnaires about teachers (student-teacher relationship, transparency of assessment criteria, motivation for teaching, teaching clarity, etc.). Conducting Faculty surveys of courses (upon passing the exam, student self-assessment of the adopted learning outcomes and student workload in relation to the number of ECTS credits allocated to activities and courses as a whole).

General information

Lead instructor(s)	**Goran Martinović, PhD, Full Professor with Tenure**	
Course title	Distributed Computer Systems	
Study programme	Graduate university study programme in Computer Engineering	
Course status	Compulsory on module: Software Engineering Elective on module: Artificial Intelligence and Robotics	
Year of study	2.	
ECTS credits and teaching methods	ECTS credits	7
	Number of classes (lectures + auditory exercises + laboratory exercises + design exercises + seminars)	45 + 0 + 15 + 0 + 0

1. COURSE DESCRIPTION

1.1. Course objectives

To give students an insight and provide basic knowledge about the properties, prerequisites and methods of establishment, usage and evaluation of distributed computer systems, parallel systems and service systems, as well as about advanced program architectures and software design patterns. Show the possibilities and explain the basics of using system and programming tools, and the development of application programs in a distributed and service computing environment, including the use of advanced programming architectures and software design patterns.

1.2. Course enrolment requirements

Requirements for enrolment in the second year of study fulfilled.

1.3. Expected learning outcomes

1. Describe the evolution and compare the types of advanced paradigms of distributed computing
2. Distinguish between different architectures of high-performance computers using Flynn's taxonomy
3. Explain the features of shared and distributed memory architectures
4. Design a parallel algorithm for systems with common and shared memory
5. Differentiate and apply different concepts of computer resource virtualization (such as memory, storage, process, network)
6. Determine the applicability of advanced paradigms of distributed computing to different types of computing tasks
7. Identify different software architectures and their elements (architectural styles, languages, connectors, middleware, ...)

1.4. Course content

Definition, goals, concepts and models of distributed computing systems. Flynn's classification of computer systems. Memory models of distributed and parallel systems. Data storage. Communication in distributed and parallel systems: layered protocols, calls of remote procedures and methods in objects, API. Middleware and programming environments. Processes and threads, multithreading, client and server processes, P2P environment, code migration, agents, virtualization. Naming system entities, distributed hash tables, hierarchical systems, mobile entities, attribute description of the system. Synchronization: logical and vector clocks, global state, selection and exclusion algorithms, transactions, location systems. Concept of consistency and replication. Fault tolerance in a distributed system. Security: security channels, management of access to resources and data. Distributed systems based on objects, documents, coordination and services. Distributed environments: computer clusters, grids and clouds. Programming of distributed and parallel computing environments: MPI, OpenMP. Distributed architectures for web and mobile services. Advanced patterns and programming architectures: RESTFul API, microservices, MVC, MVVM, MVP, Entity Framework, API development. Cloud computing: resource management, load balancing and scalability, models, standards,

algorithms, languages and system support. Green computing. Big data analysis application. Performance evaluation of complex software systems and architectures.

1.5. Types of classes	☒ lectures ☐ seminars and workshops ☐ auditory exercises ☒ distance learning ☐ field work	☒ individual exercises ☐ multimedia and network ☒ laboratory exercises ☐ design exercises ☐ working with a supervisor ☐ other _____
1.6. Comments		

1.7. Student obligations

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.8. Monitoring and assessment of student work

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.9. Assessment and evaluation of student work during classes and in the final exam

STUDENT ACTIVITY	ECTS	LEARNING OUTCOME	TEACHING METHOD	ASSESSMENT METHOD	POINTS	
					Min	Max
Attendance at lectures (L) and laboratory exercises (LE)	1.5	1,2,3,4,5,6,7	Lectures (L) and laboratory exercises (LE)	Attendance tracking. Minimum attendance percentage: 70%.	3	6
Problem-solving, modelling and programming exercises	2	3,4,5,6,7	Revision exams, written exam	Evaluation of solved exercises	15	30
Preparation for laboratory exercises (LE), results analysis, report writing	1.5	2,3,4,5,6	Laboratory exercises (LE)	Evaluation of the preparation for LE, LE supervision, LE report assessment	12	24
Practical programming tasks solving	1	3,4,6	Lectures (L) and laboratory exercises (LE)	Evaluation of the solution for given problem	5	10
Preparing for an oral exam and oral exam	1	1,2,3,4,5,6,7	Oral exam	Evaluation of the oral exam	5	30

1.10. Obligatory literature (at the time of submitting a study programme proposal)

1. M.L. Liu, Distributed Computing: Principles and Applications (1st Ed.), Pearson, 2019.
2. L. Malik, S. Arora, et al., Practical Guide to Distributed Systems in MPI, Independently published, 2017.
3. S. Newman, Building Microservices: Designing Fine-Grained Systems (2nd Ed.), O'Reilly Media, 2021.
4. P. Pacheco, M. Malensek, An Introduction to Parallel Programming, Morgan Kaufmann, 2020.
5. R. Robey, Y. Zamora, Parallel and High Performance Computing, Manning, 2021.
6. M. van Steen, A.S. Tanenbaum, Distributed Systems (3.01 Ed.), CreateSpace Independent Publishing Platform, 2017.

1.11. Recommended additional literature (at the time of submitting a study programme proposal)

1. EMC Education Services, Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data, Wiley, 2015.
2. S. Ghosh, Distributed Systems (2nd Ed.), Chapman & Hall, 2020.
3. I. Gorton, Foundations of Scalable Systems: Designing Distributed Architectures (1st Ed.), O'Reilly Media, 2022.
4. A. Kuzmiakova, Concurrent, Parallel and Distributed Computing, Arcler Press, 2022.
5. D. Raptis, Distributed Systems for Practitioners, B086551JHY, 2020.
6. R.L. Sites, Understanding Software Dynamics (1st Ed.), Addison-Wesley Professional, 2021.
7. O. Zimmermann, M. Stocker, Patterns for API Design: Simplifying Integration with Loosely Coupled Message Exchanges (1st Ed.), Addison-Wesley Professional, 2022.

1.12. Number of obligatory literature copies in relation to the number of students currently taking the course

Title	*Number of copies*	*Number of students*
1. M.L. Liu, Distributed Computing: Principles and Applications (1st Ed.), Pearson, 2019.	3	40
2. L. Malik, S. Arora, et al., Practical Guide to Distributed Systems in MPI, Independently published, 2017.	3	40
3. S. Newman, Building Microservices: Designing Fine-Grained Systems (2nd Ed.), O'Reilly Media, 2021.	3	40
4. P. Pacheco, M. Malensek, An Introduction to Parallel Programming, Morgan Kaufmann, 2020.	3	40
5. R. Robey, Y. Zamora, Parallel and High Performance Computing, Manning, 2021.	3	40

1.13. Quality assurance methods ensuring the acquisition of knowledge, skills and competences

Conducting university questionnaires about teachers (student-teacher relationship, transparency of assessment criteria, motivation for teaching, teaching clarity, etc.). Conducting Faculty surveys of courses (upon passing the exam, student self-assessment of the adopted learning outcomes and student workload in relation to the number of ECTS credits allocated to activities and courses as a whole).

General information		
Lead instructor(s)	Josip Balen, PhD, Associate Professor	
Course title	Mobile Applications Development	
Study programme	Graduate university study programme in Computer Engineering	
Course status	Compulsory on module: Software Engineering	
Year of study	1.	
ECTS credits and teaching methods	ECTS credits	5
	Number of classes (lectures + auditory exercises + laboratory exercises + design exercises + seminars)	30+0+30+15+0

1. COURSE DESCRIPTION		
1.1. Course objectives		
Introduce technologies for developing applications for mobile devices. Designing a software solution to solve real world problems using a software concept specific to the creation of mobile applications and application prototyping. Development of a mobile application, creation of a user interface, programming implementation of functionalities and connection of interface and functionalities into a meaningful mobile application. Conducting application testing on real devices and on a simulator. Creation of technical documentation of source code.		
1.2. Course enrolment requirements		
Requirements for enrolment in the first year of study fulfilled.		
1.3. Expected learning outcomes		
1. Distinguish the basic terms, methods and practices of software engineering and their areas of application 2. Analyse and define the requirements of complex software products 3. Create the design of a complex software product in the field of mobile applications 4. Create a prototype and perform validation of a complex software product for mobile devices 5. Distinguish and apply the concepts of advanced data storage models 6. Determine the users and analyse the requirements that the interface must support 7. Develop an application for network and mobile environments using modern development tools and software solutions within cloud computing 8. Conduct structural and functional testing of the application on real mobile devices 9. Create application source code documentation		
1.4. Course content		
Introduction to mobile application development tools. The main components of a mobile application. User interface design for mobile applications. Software solutions to real problems. The usage of a program-specific concepts for mobile applications development. Software design implementation. Programming implementation of various functionalities. Use and management of sensors embedded in mobile devices. Using a simulator for testing mobile application performance. Conduct structural and functional testing on real mobile devices. Source code documentation generation.		
1.5. Types of classes	☒ lectures ☐ seminars and workshops	☒ individual exercises ☐ multimedia and network ☒ laboratory exercises

	☐ auditory exercises ☒ distance learning ☐ field work	☒ design exercises ☒ working with a supervisor ☐ other _____
1.6. Comments		

1.7. Student obligations

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.8. Monitoring and assessment of student work

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.9. Assessment and evaluation of student work during classes and in the final exam

STUDENT ACTIVITY	ECTS	LEARNING OUTCOME	TEACHING METHOD	ASSESSMENT METHOD	POINTS	
					Min	Max
Attendance at lectures (L), auditory exercises (AE), laboratory exercises (LE), design exercises (DE)	0,5	1,2,3,4,5,6,7,8,9	Lectures (L), Laboratory exercises (LE), Design exercises (CE)	Attendance tracking. Minimum attendance percentage: 70%.	2,8	4
Problem-solving exercises	0,5	1,2,3,4,5	Design exercises (CE)	Evaluation	5	10
Preparation for laboratory exercises (LE), results analysis, report writing	1	1,2,3,4,5	Laboratory exercises (LE)	Preparation for LE, LE supervision, LE report assessment	6	16
Preparing for an oral exam and oral exam	1	1,2,3,4,5,6	Oral exam	Evaluation	15	30
Work on a project assignment	2	3,4,5,6,7,8,9	Independent project work	Evaluation	20	40

1.10. Obligatory literature (at the time of submitting a study programme proposal)

1. David Greenhalgh, Josh Skeen, Andrew Bailey, Big Nerd Ranch Guides; 2nd edition, 2021.
2. Dmitry Jemerov, Svetlana Isakova, Kotlin in Action, Second edition, Manning, 2023.
3. Dawn Griffiths, David Griffiths, Head First Kotlin, O'Reilly Media, Inc., 2019.

1.11. Recommended additional literature (at the time of submitting a study programme proposal)

1. Alex Forrester, Eran Boudjnah, Alexandru Dumbravan, Jomar Tigcal, How to Build Android Apps with Kotlin: A hands-on guide to developing, testing, and publishing your first apps with Android, Packt Publishing, 2021.
2. Ken Kousen, Kotlin Cookbook: A Problem-Focused Approach 1st Edition, O'Reilly Media; 1st edition, 2019.

1.12. Number of obligatory literature copies in relation to the number of students currently taking the course

Title	*Number of copies*	*Number of students*
David Greenhalgh, Josh Skeen, Andrew Bailey, Big Nerd Ranch Guides; 2nd edition, 2021.		
Dmitry Jemerov, Svetlana Isakova, Kotlin in Action, Second edition, Manning, 2023.		
Dawn Griffiths, David Griffiths, Head First Kotlin, O'Reilly Media, Inc., 2019.		

1.13. Quality assurance methods ensuring the acquisition of knowledge, skills and competences

Conducting university questionnaires about teachers (student-teacher relationship, transparency of assessment criteria, motivation for teaching, teaching clarity, etc.). Conducting Faculty surveys of courses (upon passing the exam, student self-assessment of the adopted learning outcomes and student workload in relation to the number of ECTS credits allocated to activities and courses as a whole).

General information		
Lead instructor(s)	**Časlav Livada, PhD, Associate Professor**	
Course title	Computer Games Development	
Study programme	Graduate university study programme in Computer Engineering	
Course status	Compulsory on modules: Data Science	
Year of study	1.	
ECTS credits and teaching methods	ECTS credits	5
	Number of classes (lectures + auditory exercises + laboratory exercises + design exercises + seminars)	30 + (0 + 30 + 0) + 0

1. COURSE DESCRIPTION		
1.1. Course objectives		
Expand the knowledge of object-oriented programming with the knowledge needed to create a computer game. Build a solid foundation for game design and development in order to master the Unity Game Engine. Develop highly transferable coding problem solving skills.		
1.2. Course enrolment requirements		
Requirements for enrolment in the second year of study fulfilled.		
1.3. Expected learning outcomes		
1. compare the elements for creating a computer game and, based on that, conclude which basic elements are needed to create a computer game 2. plan which tools and program libraries are needed to design a computer game 3. construct a simpler computer game based on the adopted theoretical foundations 4. interpret and valorise the design of a computer game		
1.4. Course content		
Introduction to computer game development. Object-oriented programming with emphasis on C# - classes and objects, interfaces, data access, data pre-processing. Selections. Textures. Character strings. Game management via mouse, keyboard, gamepad. 3D sound. 2D and 3D game models. Design of computer games. Interaction. Animation. Physics of computer games. Artificial intelligence.		
1.5. Types of classes	☒ lectures ☐ seminars and workshops ☐ auditory exercises ☒ distance learning ☐ field work	☐ individual exercises ☐ multimedia and network ☒ laboratory exercises ☐ design exercises ☐ working with a supervisor ☐ other _____
1.6. Comments		
1.7. Student obligations		
Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.		
1.8. Monitoring and assessment of student work		

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.9. Assessment and evaluation of student work during classes and in the final exam

STUDENT ACTIVITY	ECTS	LEARNING OUTCOME	TEACHING METHOD	ASSESSMENT METHOD	POINTS	
					Min	Max
Attendance at lectures (L), laboratory exercises (LE)	1	1	Lectures (L), laboratory exercises (LE)	Attendance tracking. Minimum attendance percentage: 70%.	5	10
Problem-solving exercises	1	2, 3	laboratory exercises (LE)	Evaluating a solution to a given problem	5	10
Preparation of the project for the oral exam and answering the questions orally	1	4	Oral exam	Evaluation of the given answers	5	10
Project	2	2, 3	Seminar work	Evaluating a solution to a given problem	35	70

1.10. Obligatory literature (at the time of submitting a study programme proposal)

1. Hocking, Joe: Unity in Action: Multiplatform Game Development in C# with Unity 5

1.11. Recommended additional literature (at the time of submitting a study programme proposal)

1. D. Graham: Game Coding Complete
2. S. Rogers: Level Up!: The Guide to Great Video Game Design
3. R. Penton: Beginning C# Game Programming
4. D. Schuller: C# Game Programming: For Serious Game Creation
5. J. Gibson: Introduction to Game Design, Prototyping, and Development: From Concept to Playable Game with Unity and C#

1.12. Number of obligatory literature copies in relation to the number of students currently taking the course

Title	*Number of copies*	*Number of students*
Hocking, Joe: Unity in Action: Multiplatform Game Development in C# with Unity 5		

1.13. Quality assurance methods ensuring the acquisition of knowledge, skills and competences

Conducting university questionnaires about teachers (student-teacher relationship, transparency of assessment criteria, motivation for teaching, teaching clarity, etc.). Conducting Faculty surveys of courses (upon passing the exam, student self-assessment of the adopted learning outcomes and student workload in relation to the number of ECTS credits allocated to activities and courses as a whole).

General information		
Lead instructor(s)	**Robert Cupec, PhD, Full Professor with Tenure**	
Course title	Robot Vision	
Study programme	Graduate university study programme in Computer Engineering	
Course status	Compulsory on modules: Artificial Intelligence and Robotics	
Year of study	1.	
ECTS credits and teaching methods	ECTS credits	5
	Number of classes (lectures + auditory exercises + laboratory exercises + design exercises + seminars)	30+(0+15+15)+0

1. COURSE DESCRIPTION

1.1. Course objectives

To introduce the applications of computer vision in robotics. To introduce the basic algorithms and data structures for representation and efficient analysis of images and 3D sensor data with application in robotics and intelligent autonomous systems. To provide knowledge of programming tools for processing images and 3D sensor data. To provide knowledge of implementing program solutions involving object detection and pose estimation from images and 3D sensor data.

1.2. Course enrolment requirements

Requirements for enrolment in the first year of study fulfilled.

1.3. Expected learning outcomes

1. Understand the basic principles of commonly used methods and tools for processing images and 3D sensor data.
2. Select appropriate computer vision methods for solving problems in robotics.
3. Select appropriate data structures to represent 2D and 3D geometric structures and their relationships.
4. Create a computer program for image processing using available programming tools.
5. Create a computer program to process 3D sensor data using available programming tools.
6. Create a program solution for object detection and pose estimation in 3D images.

1.4. Course content

Application of computer vision in robotics. Delaunay triangulation. Voronoi diagram. Hough transformation. Convex hull. Keypoint detection. Local descriptors. Random Sample Consensus algorithm (RANSAC). Generalized Hough transformation. Homography. 3D scene reconstruction from two camera images captured from different viewpoints. 3D scene reconstruction from multiple camera images. Point cloud. kD-tree. Point cloud registration. Voxel grid. Oriented 3D point. Segmentation of point clouds into planar surfaces. Segmentation of point clouds into objects. Truncated signed distance function. Marching cubes algorithm. Fusion of 3D sensor data. Iterative closest point algorithm (ICP). Object recognition and pose estimation from images and 3D sensor data. Environment map building using computer vision. Place recognition. Object detection and classification in images and 3D sensor data using machine learning. Obstacle detection.

1.5. Types of classes	☒ lectures ☐ seminars and workshops ☐ auditory exercises ☒ distance learning ☐ field work	☒ individual exercises ☐ multimedia and network ☒ laboratory exercises ☒ design exercises ☐ working with a supervisor ☐ other _____

1.6. Comments	

1.7. Student obligations

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.8. Monitoring and assessment of student work

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.9. Assessment and evaluation of student work during classes and in the final exam

STUDENT ACTIVITY	ECTS	LEARNING OUTCOME	TEACHING METHOD	ASSESSMENT METHOD	POINTS	
					Min	Max
Attendance at lectures (L), laboratory exercises (LE)	2	1, 2, 3, 4, 5, 6	Lectures (L), laboratory exercises (LE), design exercises (DE)	Attendance tracking. Minimum attendance percentage: 70%.	7	10
Preparation for laboratory exercises (LE), results analysis, report writing	1	1, 2, 3, 4, 5, 6	Laboratory exercises (LE)	Preparation for LE, LE supervision, LE report assessment	10	25
Preparing for an oral exam	1	1, 2, 3, 4, 5, 6	Oral exam	Evaluation	15	40
Problem-solving exercises set during design exercises (DE)	1	1, 2, 3, 4, 5, 6	Design exercises (DE)	Evaluation	10	25

1.10. Obligatory literature (at the time of submitting a study programme proposal)

1. G. Bradski, A. Kaehler, Learning OpenCV

1.11. Recommended additional literature (at the time of submitting a study programme proposal)

1. E. R. Davies, Machine Vision: Theory, Algorithms

2. R. Hartley, A. Zisserman, Multiple View Geometry in Computer Vision

3. O. Faugeras, Three-Dimensional Computer Vision: A Geometric Viewpoint

4. R. Cupec, Robotski vid – nastavni materijali

1.12. Number of obligatory literature copies in relation to the number of students currently taking the course

Title	*Number of copies*	*Number of students*
G. Bradski, A. Kaehler, Learning OpenCV	0	16

1.13. Quality assurance methods ensuring the acquisition of knowledge, skills and competences

Conducting university questionnaires about teachers (student-teacher relationship, transparency of assessment criteria, motivation for teaching, teaching clarity, etc.). Conducting Faculty surveys of courses

(upon passing the exam, student self-assessment of the adopted learning outcomes and student workload in relation to the number of ECTS credits allocated to activities and courses as a whole).

General information		
Lead instructor(s)	**Alfonzo Baumgartner, PhD, Associate Professor**	
Course title	System Programming	
Study programme	Graduate university study programme in Computer Engineering	
Course status	Compulsory on module: Software Engineering Elective on module: Artificial Intelligence and Robotics	
Year of study	1.	
ECTS credits and teaching methods	ECTS credits	6
	Number of classes (lectures + auditory exercises + laboratory exercises + design exercises + seminars)	45+(0+15+0)+0

1. COURSE DESCRIPTION		
1.1. Course objectives		
Introduce students with the capabilities and limitations of operating systems, as well as user and environment requirements. Introduce students with the development of moderately complex and effective system and application software projects with the help of modern programme principles and tools.		
1.2. Course enrolment requirements		
Requirements met for enrolling in the study programme		
1.3. Expected learning outcomes		
1. describe and use Windows API for file, memory and process management 2. solve complex problems with threads and use synchronisation mechanisms along with reliable multithreading models 3. use interprocess communication and implement ways of network communication 4. design system programmes which use asynchronous I/O and explain novelties in Win64 API 5. write system programmes which use Win32 API		
1.4. Course content		
Requirements on system and application software. Analysis of modern operating systems (Unix, Linux, Windows) in different complexity environments. Design of simple drivers and applications. Basic programming techniques. File and directory control. Control of input-output units and ports. Security services. Memory management. Dll files. Exceptions handling. Processes and threads in programmes: events and exclusion, multithreading. Signals. Interprocess communication: pipes and messages. Fundamentals of network programming: sockets. Design of system software in embedded systems and design of some Win32 and Win64 services. Graphical user interface: windows, controls. Timing function programming. System monitoring and measurement programmes. Approaches and models that enable an increase and evaluation of system performance.		
1.5. Types of classes	☒ lectures ☐ seminars and workshops ☐ auditory exercises ☒ distance learning ☐ field work	☒ individual exercises ☐ multimedia and network ☒ laboratory exercises ☐ design exercises ☐ working with a supervisor ☐ other _____
1.6. Comments		

1.7. Student obligations

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.8. Monitoring and assessment of student work

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.9. Assessment and evaluation of student work during classes and in the final exam

STUDENT ACTIVITY	ECTS	LEARNING OUTCOME	TEACHING METHOD	ASSESSMENT METHOD	POINTS	
					Min	Max
Attendance at lectures (L), auditory exercises (AE), laboratory exercises (LE), design exercises (DE)	2	1,2,3,4,5	Lectures (L), laboratory exercises (LE), design exercises (CE)	Attendance tracking. Minimum attendance percentage: 70%.	5	10
Tests	1.2	1,2,4	Digital exam through Merlin system	Automatic check of answers	0	20
Preparation for laboratory exercises (LE), results analysis, report writing	1	2, 3, 5	Laboratory exercises (LE)	Preparation for LE, LE supervision, LE report assessment	12	30
Preparing for an oral exam and oral exam	1.8	1,2,3,4,5	Oral exam	Evaluation	20	40

1.10. Obligatory literature (at the time of submitting a study programme proposal)

1. Hart, J.M.: "Windows System Programming (3rd Ed.)", Boston: Addison Wesley Professional, 2004.

1.11. Recommended additional literature (at the time of submitting a study programme proposal)

1. A.S. Tanenbaum: "Modern Operating Systems (2nd Ed.)", Prentice Hall, Englewood Cliffs, NJ, 2001.

2. Microsoft Windows Team Staff: "Microsoft Windows XP Professional Resource Kit", Microsoft Press, 2003.

3. R. Grehan, R. Moote, I. Cyliax: "Real-Time Programming: A Guide to 32-bit Embedded Development", Addison Wesley, New York, NY, 1999.

4. D. Vandevoorde, N.M. Josuttis: "C++ Templates: The Complete Guide", Addison-Wesley Professional, Boston, NY, 2002.

5. M.E. Russinovich, D.A. Solomon: "Microsoft Windows Internals (4th Ed.): Microsoft Windows Server(TM) 2003, Windows XP, and Windows 2000", Microsoft Press, 2004.

6. K.A. Robbins, S. Robbins: "Unix Systems Programming: Communication, Concurrency and Threads", Prentice Hall, Indianapolis, IN, 2003.

7. S. Walther: "Sams Teach Yourself Visual Studio.NET in 21 Days", Sams, Indianapolis, IN, 2003

1.12. Number of obligatory literature copies in relation to the number of students currently taking the course

Title	*Number of copies*	*Number of students*

Hart, J.M.: "Windows System Programming (3rd Ed.)", Boston: Addison Wesley Professional, 2004.	?	?

1.13. Quality assurance methods ensuring the acquisition of knowledge, skills and competences

Conducting university questionnaires about teachers (student-teacher relationship, transparency of assessment criteria, motivation for teaching, teaching clarity, etc.). Conducting Faculty surveys of courses (upon passing the exam, student self-assessment of the adopted learning outcomes and student workload in relation to the number of ECTS credits allocated to activities and courses as a whole).

General information		
Lead instructor(s)	**Ratko Grbić, PhD, Associate Professor** **Emmanuel Karlo Nyarko, PhD, Associate Professor**	
Course title	Machine Learning	
Study programme	Graduate university study programme in Computer Engineering	
Course status	Compulsory on modules: Artificial Intelligence and Robotics and Data Science	
Year of study	1.	
ECTS credits and teaching methods	ECTS credits	6
	Number of classes (lectures + auditory exercises + laboratory exercises + design exercises + seminars)	(45+(0+30+0)+0)

1. COURSE DESCRIPTION		
1.1. Course objectives		
Introduction to methods and algorithms in the field of machine learning and their respective applications. Problem-solving in the domains of supervised, unsupervised, and reinforcement learning. Acquisition of relevant skills using modern libraries and application frameworks for machine learning.		
1.2. Course enrolment requirements		
Requirements for enrolment in the first year of study fulfilled.		
1.3. Expected learning outcomes		
1. Explain the basic concepts in machine learning and identify the advantages and disadvantages of machine learning algorithms. 2. Evaluate the suitability of a particular machine learning algorithm for a given problem. 3. Explain the methods of selection and evaluation of built models. 4. Use modern application frameworks for machine learning algorithms development. 5. Select and implement algorithms in solving problems from the domain of unsupervised, (semi)supervised and reinforcement learning.		
1.4. Course content		
Introduction and basic terminology in the field of machine learning. Data preparation. Algorithms for unsupervised learning. Data clustering. Dimensionality reduction. Algorithms for supervised learning. Linear and logistic regression. Decision trees. Random forests. Support vector machines. Feedforward neural networks. Recurrent neural networks. Convolutional neural networks. Reinforcement learning. Markov decision process. Q-learning. Model evaluation. Different applications of machine learning and examples.		
1.5. Types of classes	☒ lectures ☐ seminars and workshops ☐ auditory exercises ☒ distance learning ☐ field work	☒ individual exercises ☐ multimedia and network ☒ laboratory exercises ☐ design exercises ☐ working with a supervisor ☐ other _____
1.6. Comments		
1.7. Student obligations		

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.8. Monitoring and assessment of student work

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.9. Assessment and evaluation of student work during classes and in the final exam

STUDENT ACTIVITY	ECTS	LEARNING OUTCOME	TEACHING METHOD	ASSESSMENT METHOD	POINTS	
					Min	Max
Attendance at lectures (L), laboratory exercises (LE)	2	1,2,3,4,5	Lectures (L), laboratory exercises (LE)	Attendance tracking. Minimum attendance percentage: 70%.	7	10
Preparation for laboratory exercises (LE), results analysis, report writing	1.5	3, 4, 5	Laboratory exercises (LE)	Preparation for LE, LE supervision, LE report assessment	15	30
Solving the project task	1	2, 3, 4, 5	Individual exercises	Verification of the solution of the project task, presentation of the solution	0	25
Preparing for an oral exam and oral exam	1.5	1, 2, 3, 5	Oral exam	Evaluation	18	35

1.10. Obligatory literature (at the time of submitting a study programme proposal)

1. S. Raschka, Y. (Hayden) Liu, V. Mirjalili, Machine Learning with PyTorch and Scikit-Learn: Develop machine learning and deep learning models with Python, 1st Edition, Packt Publishing, 2022.

1.11. Recommended additional literature (at the time of submitting a study programme proposal)

1. E. Alpaydin, Introduction to Machine Learning, 4th Edition, MIT Press, 2020.

1.12. Number of obligatory literature copies in relation to the number of students currently taking the course

Title	*Number of copies*	*Number of students*
S. Raschka, Y. (Hayden) Liu, V. Mirjalili, Machine Learning with PyTorch and Scikit-Learn: Develop machine learning and deep learning models with Python, 1st Edition, Packt Publishing, 2022.		
E. Alpaydin, Introduction to Machine Learning, 4th Edition, MIT Press, 2020.		

1.13. Quality assurance methods ensuring the acquisition of knowledge, skills and competences

Conducting university questionnaires about teachers (student-teacher relationship, transparency of assessment criteria, motivation for teaching, teaching clarity, etc.). Conducting Faculty surveys of courses (upon passing the exam, student self-assessment of the adopted learning outcomes and student workload in relation to the number of ECTS credits allocated to activities and courses as a whole).

General information		
Lead instructor(s)		
Course title	Practical Training	
Study programme	Graduate university study programme in Computer Engineering	
Course status	Compulsory on all modules	
Year of study	2.	
ECTS credits and teaching methods	ECTS credits	10
	Number of classes (lectures + auditory exercises + laboratory exercises + design exercises + seminars)	0+(0+0+200)+0

1. COURSE DESCRIPTION

1.1. Course objectives

Introduce the student to the working environment within the company, the organisational structure of the production-business system, managers and their responsibilities, the production technology used in the company, and the prescribed occupational safety measures and procedures related to the technology employed by the company. The student becomes acquainted with engineering tasks and responsibilities and, under the mentor's supervision, may actively engage in these tasks while adhering to safety measures, professional and technological rules, as well as other company regulations. Upon completion of the practical training, the student prepares a report on the completed training, following the typical format of engineering communication.

1.2. Course enrolment requirements

Requirements for enrolment in the second year of study fulfilled.

1.3. Expected learning outcomes

1. Evaluate the organisational structure of the production-business system, as well as the tasks and roles of managers within it.
2. Assess engineering tasks, as well as the necessary knowledge and skills related to the production technology in the company.
3. Evaluate and master the prescribed occupational safety measures and procedures related to the production technology in the company.
4. Acquire knowledge of engineering communication and apply it.
5. Identify the need for independent acquisition of knowledge and skills necessary for successfully solving a given complex project task based on self-assessment of one's own competencies.
6. Implement an activity plan for solving a given complex project task in the field of computer engineering.
7. Document one's own solution to the given complex project task by preparing a Master's thesis and/or a relevant technical report, as well as by preparing presentation materials.

1.4. Course content

Students complete 200 hours of practical training (an average of 13 working hours a week). Each student individually has practical training in a company doing the tasks they have been prepared for throughout their education. Under the guidance of a mentor, the student gets to know the organisational structure of the production-business system, the production technology as well as occupational safety measures and procedures at work, and gets involved in engineering work, respecting the safety measures, professional and technological rules, as well as other rules of the company. During practical training, the student keeps a work diary. Practical training is organised by FERIT in cooperation with engineers employed in companies whose field of activity is computer engineering. These engineers are appointed as mentors by the Faculty that also

coordinates the student work plan with them. The organisation of practical training is prescribed by the Regulations on practical training of FERIT students.

1.5. Types of classes	☐ lectures ☐ seminars and workshops ☐ auditory exercises ☒ distance learning ☐ field work	☒ individual exercises ☐ multimedia and network ☐ laboratory exercises ☒ design exercises ☒ working with a supervisor ☐ other _____
1.6. Comments		

1.7. Student obligations

Defined by the Regulations on practical training.

1.8. Monitoring and assessment of student work

Defined by the Regulations on practical training.

1.9. Assessment and evaluation of student work during classes and in the final exam

STUDENT ACTIVITY	ECTS	LEARNING OUTCOME	TEACHING METHOD	ASSESSMENT METHOD	POINTS	
					Min	Max
Attendance at design exercises (DE)	6,5	1, 2, 3, 4	Design exercises	Attendance tracking. Minimum attendance percentage: 80%.	32	40
Problem-solving exercises set during design exercises (DE)	2,5	5, 6	Individual exercises	Solution evaluation of the given problem	15	30
Writing a report on the completed practical training	1	7	Practical training	Evaluation	15	30

1.10. Obligatory literature (at the time of submitting a study programme proposal)

1. Regulations on practical training of FERIT students
2. Regulations on occupational health and safety in the Republic of Croatia

1.11. Recommended additional literature (at the time of submitting a study programme proposal)

1.12. Number of obligatory literature copies in relation to the number of students currently taking the course

Title	*Number of copies*	*Number of students*
Regulations on practical training of FERIT students		
Regulations on occupational health and safety in the Republic of Croatia		

1.13. Quality assurance methods ensuring the acquisition of knowledge, skills and competences
Conducting university questionnaires about teachers (student-teacher relationship, transparency of assessment criteria, motivation for teaching, teaching clarity, etc.). Conducting Faculty surveys of courses (upon passing the exam, student self-assessment of the adopted learning outcomes and student workload in relation to the number of ECTS credits allocated to activities and courses as a whole).

General information		
Lecturer(s)	**Ivan Aleksi, PhD, Associate Professor** **Tomislav Matić, PhD, Associate Professor**	
Course title	Ubiquitous computing	
Study programme	Graduate university study programme in Computer Engineering	
Course status	Compulsory on module: Computer Engineering	
Year of study	2.	
ECTS credits and teaching methods	ECTS credits	5
	Number of classes (lectures + laboratory exercises)	30 + 30

1. COURSE DESCRIPTION		
1.1. Course objectives		
The course aims to provide a comprehensive understanding of ubiquitous computing, its fundamental concepts, and the challenges inherent in the field. Students are expected to gain skills in identifying and analyzing specific problems, along with developing both hardware and software support tailored to various computer systems. The course emphasizes the practical aspect of creating systems, encouraging students to construct and refine these systems. Additionally, it focuses on testing and evaluating the performance of these systems in diverse environments, ensuring their effectiveness and efficiency.		
1.2. Course enrolment requirements		
Requirements for enrolment in the second year of study fulfilled.		
1.3. Expected learning outcomes		
1. Explain the basic concepts of ubiquitous computing. 2. Identify specific problems in the field of ubiquitous computing. 3. Analyse requirements for a given application. 4. Develop appropriate hardware support for a computer system. 5. Develop appropriate software support for a given system. 6. Test the system's performance in a given environment.		
1.4. Course content		
Introduction and basic concepts of ubiquitous computing. Requirements for ubiquitous computing systems. Computer system interfaces. Autonomous systems. Smart devices. Communication. Localization. Examples of computer systems. The future of ubiquitous computing. Course objectives: To present theoretical and practical knowledge in the field of ubiquitous computing to students. To teach students to identify specific problems in the field of ubiquitous computing. To enable students to analyse and develop hardware and software support for computer systems in the field of ubiquitous computing.		
1.5. Types of classes	☒ lectures ☐ seminars and workshops ☐ auditory exercises ☒ distance learning ☐ field work	☐ individual exercises ☐ multimedia and network ☒ laboratory exercises ☐ design exercises ☐ working with a supervisor ☐ other _____
1.6. Comments		

1.7. Student obligations

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.8. Monitoring and assessment of student work

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.9. Assessment and evaluation of student work during classes and in the final exam

STUDENT ACTIVITY	ECTS	LEARNING OUTCOME	TEACHING METHOD	ASSESSMENT METHOD	POINTS	
					Min	Max
Attendance at lectures (L), laboratory exercises (LE)	2	1	Lectures (L), laboratory exercises (LE), design exercises (CE)	Attendance tracking. Minimum attendance percentage: 70%.	0	10
Problem-solving exercises	1	5, 6	Revision exams (written exam)	Evaluation	10	20
Preparation for laboratory exercises (LE), results analysis, report writing	1	1, 2, 3, 5	Laboratory exercises (LE)	Preparation for LE, LE supervision, LE report assessment	10	20
Preparing for an oral exam and oral exam	1	1, 2, 3, 4, 5, 6	Oral exam	Evaluation	25	50

1.10. Obligatory literature (at the time of submitting a study programme proposal)

1. Natalia Silvis-Cividjian Pervasive Computing: Engineering Smart Systems
2. Stefan Poslad Ubiquitous Computing: Smart Devices, Environments and Interactions

1.11. Recommended additional literature (at the time of submitting a study programme proposal)

1. John Krumm Ubiquitous Computing

1.12. Number of obligatory literature copies in relation to the number of students currently taking the course

Title	*Number of copies*	*Number of students*
Natalia Silvis-Cividjian, Pervasive Computing: Engineering Smart Systems	1	32
Stefan Poslad, Ubiquitous Computing: Smart Devices, Environments and Interactions	1	32
John Krumm, Ubiquitous Computing	1	32

1.13. Quality assurance methods ensuring the acquisition of knowledge, skills and competences

Conducting university questionnaires about teachers (student-teacher relationship, transparency of assessment criteria, motivation for teaching, teaching clarity, etc.). Conducting Faculty surveys of courses (upon passing the exam, student self-assessment of the adopted learning outcomes and student workload in relation to the number of ECTS credits allocated to activities and courses as a whole).

General information

Lead instructor(s)	**Tomislav Keser, PhD, Associate Professor**	
Course title	Embedded Computer Systems	
Study programme	Graduate university study programme in Computer Engineering	
Course status	Compulsory on modules: Computer Engineering and Artificial Intelligence and Robotics	
Year of study	1.	
ECTS credits and teaching methods	ECTS credits	6
	Number of classes (lectures + auditory exercises + laboratory exercises + design exercises + seminars)	30+0+30+0+0

1. COURSE DESCRIPTION

1.1. Course objectives

Students should become familiar with the versatility and ubiquity of the use of digital computer systems in applications that are not only related to computer science and information processing. To show them the principles of analysis, definition and synthesis of computer systems for specialized purposes in the function of managing and/or directing real-world processes using appropriate computer architecture. To teach them to recognize, analyse, define and design digital control systems based on microcomputers, microcontrollers and/or DSP systems. They will learn the basic principles of programming embedded computer systems, their circuit design and their implementation and installation in real control systems.

1.2. Course enrolment requirements

Requirements for enrolment in the second year of study fulfilled.

1.3. Expected learning outcomes

1. Differentiation between computer systems based on microprocessors, microcontrollers and digital signal processors.
2. Explain the special features of the use of microprocessors, microcontrollers and DSPs in integrated applications.
3. Formulate and evaluate requirements and select an embedded computer system based on application requirements.
4. Analyse and evaluate the applicability of an embedded computer system for real-time work.
5. Analyse and evaluate embedded computer systems in accordance with application requirements.
6. Design software support according to application requirements.
7. Design the circuit of the embedded computer system in selected CAD tools.

1.4. Course content

Basic concepts of computer science. Architecture and organisation of microprocessors, microcontrollers and digital signal processors. Characteristic features and special features of embedded computer systems. Structure and fundamentals of embedded computer systems. Equipment for circuit development. Design of printed circuits. Equipment for the creation of software support. Reliability and security of embedded systems. Testing, verification and validation of embedded systems. Applications of embedded systems. Application in intelligent measurement processes. Application in process control. Application in the monitoring, collection and distribution of data.

1.5. Types of classes	☒ lectures ☐ seminars and workshops ☐ auditory exercises	☒ individual exercises ☐ multimedia and network ☒ laboratory exercises ☐ design exercises

	☒ distance learning ☐ field work	☐ working with a supervisor ☐ other _____
1.6. Comments		

1.7. Student obligations

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.8. Monitoring and assessment of student work

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.9. Assessment and evaluation of student work during classes and in the final exam

STUDENT ACTIVITY	ECTS	LEARNING OUTCOME	TEACHING METHOD	ASSESSMENT METHOD	POINTS	
					Min	Max
Attendance at lectures (L), auditory exercises (AE), laboratory exercises (LE), design exercises (DE)	1	1-7	Lectures (L), laboratory exercises (LE), design exercises (CE)	Attendance tracking. Minimum attendance percentage: 70%.	5	10
Project task – individual problem-solving exercises	3	3-7	Individual work/exercise (project work)	Project task evaluation	25	50
Problem-solving exercises	0	0	Revision exams (written exam)	Evaluation	0	0
Preparation for laboratory exercises (LE), results analysis, report writing	1	6, 7	Laboratory exercises (LE)	Preparation for LE, LE supervision, LE report assessment	8	15
Preparing for an oral exam and oral exam	1	1-7	Oral exam	Evaluation	12	25

1.10. Obligatory literature (at the time of submitting a study programme proposal)

1. E. White Making Embedded Systems O Reilly Media, 2011. (ISBN 978-1-4493-0214-6)
2. E. A. Lee, S. A. Seshia Introduction to Embedded Systems, A Cyber-Physical Systems Approach, Edition 1.5 2014. (ISBN 978-0-557-70857-4)

1.11. Recommended additional literature (at the time of submitting a study programme proposal)

1. Roger Young How Computers Work: Processor and Main Memory Roger Stephen Young, 2001.
2. Sophocles J. Orfanidis Optimum Signal Processing Rutgers University, 2nd Edition, 2007., eBook (free)
3. Michael J. Pont Patterns for Time-Triggered Embedded Systems Addison-Wesley, 2014.

1.12. Number of obligatory literature copies in relation to the number of students currently taking the course

Title	*Number of copies*	*Number of students*
Christopher T. Robertson, Printed Circuit Board Designer's Reference: Basics, Prentice Hall Professional, 2004.	0	15

Eric Bogatin, Signal and Power Integrity, Simplified, Prentice Hall, 2018.	0	15
David A. Weston, Electromagnetic Compatibility: Methods, Analysis, Circuits, and Measurement, CRC Press 2016.	0	15
1.13. Quality assurance methods ensuring the acquisition of knowledge, skills and competences		
Conducting university questionnaires about teachers (student-teacher relationship, transparency of assessment criteria, motivation for teaching, teaching clarity, etc.). Conducting Faculty surveys of courses (upon passing the exam, student self-assessment of the adopted learning outcomes and student workload in relation to the number of ECTS credits allocated to activities and courses as a whole).		

General information

Lecturer(s)	**Damir Blažević, Phd, Full Professor**	
Course title	Introduction to Artificial Intelligence	
Study programme	Graduate university study programme in Computer Engineering	
Course status	Compulsory on modules: Computer Engineering and Artificial Intelligence and Robotics,	
Year of study	1.	
ECTS credits and teaching methods	ECTS credits	6
	Number of classes (lectures + auditory exercises + laboratory exercises + design exercises + seminars)	30+(15+30+0)+O

1. COURSE DESCRIPTION

1.1. Course objectives

Master basic knowledge in the field of artificial intelligence. Familiarize them with the properties of intelligent agents needed to solve problems. Create a problem state space. Present the solution of problems written in first-order logic. Acquaint the participants with ways of recording knowledge, planning and making decisions with and without the presence of uncertainty.

1.2. Course enrolment requirements

Requirements for enrolment in the first year of study fulfilled.

1.3. Expected learning outcomes

1. Select and evaluate the necessary properties of an agent to solve a given problem.
2. Design an algorithm for solving a given problem adapted to the agent's actions.
3. Show the state space of the given problem and find a solution by applying the appropriate search.
4. Create and evaluate a solution to a problem written in first-order logic.
5. Prepare and shape information (knowledge) in a form suitable for the agent's processing.
6. Identify sources of uncertainty in the process and plan decisions with the existence of uncertainty.

1.4. Course content

Intelligent agents. Problems and search spaces. Search types. Undirected and directed search. Heuristic function. Propositional and predicate logic. Inference methods. Presentation of knowledge. Working with contradictory and indeterminate systems. Fuzzy logic. Damster-Shafer theory.

1.5. Types of classes	☒ lectures ☐ seminars and workshops ☒ auditory exercises ☒ distance learning ☐ field work	☐ individual exercises ☐ multimedia and network ☒ laboratory exercises ☐ design exercises ☐ working with a supervisor ☐ other _____
1.6. Comments		

1.7. Student obligations

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.8. Monitoring and assessment of student work

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.9. Assessment and evaluation of student work during classes and in the final exam

STUDENT ACTIVITY	ECTS	LEARNING OUTCOME	TEACHING METHOD	ASSESSMENT METHOD	POINTS	
					Min	Max
Attendance at lectures (L), auditory exercises (AE), laboratory exercises (LE), design exercises (DE)	1,5	1-6	Lectures (L), laboratory exercises (LE), design exercises (CE)	Attendance tracking. Minimum attendance percentage: 70%.	4	8
Problem-solving exercises	1,5	3,4	Revision exams (written exam)	Evaluation	16	32
Preparation for laboratory exercises (LE), results analysis, report writing	1	2,3,4	Laboratory exercises (LE)	Preparation for LE, LE supervision, LE report assessment	10	20
Preparing for an oral exam and oral exam	2	1-6	Oral exam	Evaluation	16	40

1.10. Obligatory literature (at the time of submitting a study programme proposal)

S. Russell, P. Norvig, Artificial Intelligence: A Modern Approach (4th Edition), Pearson, 2020

1.11. Recommended additional literature (at the time of submitting a study programme proposal)

M. Mitchell, Artaficial Intelligence: A guide for Thinking Humans, Farrar, Straus and Giroux, 2019.

E. J. Larson, The Myth of Artificial Intelligence: Why Computers Can't Think the Way We Do,Belknap Press, 2022.

1.12. Number of obligatory literature copies in relation to the number of students currently taking the course

Title	*Number of copies*	*Number of students*
S. Russell, P. Norvig, Artificial Intelligence: A Modern Approach (4th Edition), Pearson, 2020.		10

1.13. Quality assurance methods ensuring the acquisition of knowledge, skills and competences

Conducting university questionnaires about teachers (student-teacher relationship, transparency of assessment criteria, motivation for teaching, teaching clarity, etc.). Conducting Faculty surveys of courses (upon passing the exam, student self-assessment of the adopted learning outcomes and student workload in relation to the number of ECTS credits allocated to activities and courses as a whole).

General information

Lecturer(s)	**Josip Job, PhD, Full Professor, Časlav Livada, PhD, Associate Professor**	
Course title	Data Visualization	
Study programme	Graduate university study programme in Computer Engineering	
Course status	Compulsory on modules:Software Engineering and Data Science	
Year of study	1.	
ECTS credits and teaching methods	ECTS credits	5
	Number of classes (lectures + auditory exercises + laboratory exercises + design exercises + seminars)	30 + (0 + 15 + 15) + 0

1. COURSE DESCRIPTION

1.1. Course objectives

To familiarise students with the theoretical and practical foundations of data visualization. Instruct them on using tools for data visualization. Enable them to work independently and collaboratively on data visualization projects, fostering critical thinking and evaluation of data visualizations.

1.2. Course enrolment requirements

Fulfilled requirements for study enrollment.

1.3. Expected learning outcomes

1. Describe the basic elements of visualization.
2. Design and create your own data visualization using appropriate tools and programming libraries.
3. Propose the design of data visualization according to best practice and in accordance with theoretical foundations.
4. Interpret and analyze the visualization design.

1.4. Course content

Introduction to data visualization, importance of data visualization: information storage, decision support, information transfer. Types of data. Nominal, ordinal and quantitative data. Dimensions and measures. Visual coding variables. Visualization reference model. Data visualization design. Data analysis. Visualization of multidimensional data. Perception, human visual system, Gestalt psychology. Interaction. Animation. Cartography. Graphs and trees. Colors. Narrative visualization. Text visualization. Visualization evaluation. Data visualization tools.

1.5. Types of classes	☒ lectures ☐ seminars and workshops ☐ auditory exercises ☒ distance learning ☐ field work	☐ individual exercises ☐ multimedia and network ☒ laboratory exercises ☒ design exercises ☐ working with a supervisor ☐ other _____
1.6. Comments		

1.7. Student obligations

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.8. Monitoring and assessment of student work

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.9. Assessment and evaluation of student work during classes and in the final exam

STUDENT ACTIVITY	ECTS	LEARNING OUTCOME	TEACHING METHOD	ASSESSMENT METHOD	POINTS	
					Min	Max
Attendance at lectures (L), auditory exercises (AE), laboratory exercises (LE), design exercises (DE)	1,4	1, 2, 3, 4	Lectures (L), laboratory exercises (LE), design exercises (CE)	Attendance tracking. Minimum attendance percentage: 70%.	0	0
Preparation for laboratory exercises (LE), results analysis, report writing	0,3	1, 2, 3, 4	Laboratory exercises (LE)	Preparation for LE, LE supervision, LE report assessment	5	10
Preparation for design exercises (DE), results analysis, report writing	0,3	1, 2, 3, 4	Design exercises (DE)	Preparation for DE, DE supervision, DE report assessment	5	10
Problem-solving exercises	1	1, 2, 3, 4	Design exercises (DE)	Project evaluation	20	40
Preparing for an oral exam and oral exam	2	1, 2, 3, 4	Oral exam	Evaluation	20	40

1.10. Obligatory literature (at the time of submitting a study programme proposal)

1. E. R. Tufte The Visual Display of Quantitative Information, 2nd edition
2. Murray, S. Interactive Data Visualization for the Web

1.11. Recommended additional literature (at the time of submitting a study programme proposal)

1. M. Maclean D3 Tips & Tricks

1.12. Number of obligatory literature copies in relation to the number of students currently taking the course

Title	*Number of copies*	*Number of students*
1. E. R. Tufte The Visual Display of Quantitative Information, 2nd edition	1	60
2. Murray, S. Interactive Data Visualization for the Web	1	60

1.13. Quality assurance methods ensuring the acquisition of knowledge, skills and competences

Conducting university questionnaires about teachers (student-teacher relationship, transparency of assessment criteria, motivation for teaching, teaching clarity, etc.). Conducting Faculty surveys of courses (upon passing the exam, student self-assessment of the adopted learning outcomes and student workload in relation to the number of ECTS credits allocated to activities and courses as a whole).

General information

Lead instructor(s)	**Krešimir Nenadić, PhD, Full Professor**	
Course title	Web Programming	
Study programme	Graduate university study programme in Computer Engineering	
Course status	Compulsory on modules: Software Engineering and Data Science	
Year of study	2.	
ECTS credits and teaching methods	ECTS credits	7
	Number of classes (lectures + auditory exercises + laboratory exercises + design exercises + seminars)	45+(15+15+0)+0

1. COURSE DESCRIPTION

1.1. Course objectives

The aim of the course is to present modern client and server technologies that make it possible to create dynamic and modern websites.

1.2. Course enrolment requirements

Requirements for enrolment in the second year of study fulfilled.

1.3. Expected learning outcomes

1. Compare different technologies and use them in creating web documents and applications.
2. Identify client and server technologies and select appropriate technologies for creating a specific task in the form of a web page or web application.
3. Choose the appropriate way to access the database via the web, develop your own solution in the form of a website and server and client functionality into a meaningful web application.
4. Analyse and solve a specific problem, combine different technologies to create a web application and predict possible application extensions.

1.4. Course content

Advanced use of the JavaScript language. Using the jQuery library - selectors, events, dynamic creation of document content. Responsive Design - Media Queries in CSS3, Grid System responsive design. Bootstrap development framework - application of ready-made classes and functionality in the creation web document. Advanced use of PHP - object and PDO database access, work with sessions and cookies (SESSION, COOKIES), data processing from forms. Object oriented approach to creating PHP documents with a focus on data protection. AJAX technology using jQuery libraries.

1.5. Types of classes

☒ lectures	☐ individual exercises
☐ seminars and workshops	☐ multimedia and network
☒ auditory exercises	☒ laboratory exercises
☒ distance learning	☐ design exercises
☐ field work	☐ working with a supervisor
	☐ other _____

1.6. Comments	

1.7. Student obligations

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.8. Monitoring and assessment of student work

Defined by the Student evaluation criteria of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek and paragraph 1.9.

1.9. Assessment and evaluation of student work during classes and in the final exam

STUDENT ACTIVITY	ECTS	LEARNING OUTCOME	TEACHING METHOD	ASSESSMENT METHOD	POINTS	
					Min	Max
Attendance at lectures (L), auditory exercises (AE), laboratory exercises (LE), design exercises (DE)	1.5	1,2	Lectures (L), laboratory exercises (LE), design exercises (CE)	Attendance tracking. Minimum attendance percentage: 70%.	0	10
Problem-solving exercises	1	2,3	Revision exams (written exam)	Evaluation	15	20
Preparation for laboratory exercises (LE), results analysis, report writing	1.7	2,3,4	Laboratory exercises (LE)	Preparation for LE, LE supervision, LE report assessment	20	30
Preparing for an oral exam and oral exam	2.8	1, 2, 3, 5	Oral exam	Evaluation	15	30

1.10. Obligatory literature (at the time of submitting a study programme proposal)

1. Lukić, Ivica; Köhler, Mirko Osnove Internet programiranja
2. Sebesta, R.W. Programming the World Wide Web (2nd Ed.)

1.11. Recommended additional literature (at the time of submitting a study programme proposal)

1. T. Powell, Thomas Web Design: The Complete Reference

2. M. Hall, L. Brown Core Web programming, A Sun Microsystems

3. K. Kalata Internet Programming

4. F. Halsall Computer Networking and the Internet (5th Ed.)

5. H. Deitel, P. Deitel, T. Nieto, K. Steinbuhler The Complete Wireless Internet and Mobile Business

Programming Training Course

1.12. Number of obligatory literature copies in relation to the number of students currently taking the course

Title	*Number of copies*	*Number of students*

1.13. Quality assurance methods ensuring the acquisition of knowledge, skills and competences

Conducting university questionnaires about teachers (student-teacher relationship, transparency of assessment criteria, motivation for teaching, teaching clarity, etc.). Conducting Faculty surveys of courses (upon passing the exam, student self-assessment of the adopted learning outcomes and student workload in relation to the number of ECTS credits allocated to activities and courses as a whole).

6.4. Evaluation criteria

OKVIRI KRITERIJA OCJENJIVANJA STUDENATA FERIT-a

U Tablici 1 su prikazane moguće aktivnosti tijekom semestra, „pragovi", preporučeni udio pojedinačne aktivnosti u ukupnom broju bodova ostvarivih tijekom semestra i sl. Za svaki predmet potrebno je uz praćenje pohađanja nastave provoditi još najmanje dvije aktivnosti. Ako je za aktivnost potreban broj bodova/postotak naveden u obliku „od-do", nositelj predmeta za svoj predmet treba za tu aktivnost odrediti točno potreban broj bodova/postotak unutar tog raspona.

Ako studenti ne ostvare minimalno potreban uspjeh iz svih aktivnosti da bi se one smatrale uspješno položenim, tj. ako ne ostvare „pragove" iz svih aktivnosti, nemaju pravo prijaviti ispit, nego trebaju nadoknaditi aktivnost.

Ako student dobrovoljno želi neku aktivnost izvršavati ponovno sljedeće ak. godine, onda se podrazumijeva da niti jedna aktivnost na predmetu nije uspješno položena, tj. student mora ponovno polagati sve aktivnosti na predmetu. Studentu koji ponovno izvršava aktivnosti, predmetni nastavnik može u potpunosti ili djelomično priznati uspješno odrađene aktivnosti u prethodnoj godini (npr. uspješno pohađanje nastave ili bodove iz LV) te student ima pravo ponovno pristupiti kontrolnim zadaćama i u tom slučaju student je dužan na početku akademske godine nastavniku najaviti dolaske na kontrolne zadaće.

Ako je trajanje uspješno položene aktivnosti i/ili bodova vezano uz ispitni rok, onda to znači da je vezano za jedan ispitni termin u slučaju izvanrednih ispitnih rokova, odnosno za najviše oba ispitna termina redovitog ispitnog roka (zimski, ljetni, jesenski). Iznimno, ako se održava izvanredni ispitni rok u rujnu, onda uspješno položene aktivnosti i/ili bodovi u jesenskom roku obuhvaćaju i taj izvanredni ispitni rok.

Ukupan broj bodova (UBB) i konačna ocjena određuju se prema Tablici 2.

Za sve studente vrijede oni kriteriji koji su vrijedili pri prvom upisu predmeta. Ako student pri ponavljanju predmeta izvršava ponovno sve aktivnosti, tada za studenta vrijede oni kriteriji koji su definirani za ak. godini u kojoj student ponavlja predmet.

Studenti u statusu „dovršetka studija" po razini opterećenja jednaki su redovitim studentima, te se stoga na njih odnose sve odredbe na isti način kao i na redovite studente.

Pod terminom nastave smatra se razdoblje od najmanje jednog školskog sata istog oblika nastave iz istog predmeta tijekom kojeg nastavnik evidentira nazočnost studenata.

Studentu koji ometa izvođenje nastave nastavnika i/ili praćenje nastave ostalih studenata, odnosno izvođenje provjere znanja, nastavnik ima pravo poništiti evidentiranu nazočnost u dotičnom terminu, odnosno evidentirati za termin neopravdani izostanak, te ga uputiti da napusti prostoriju. Usto prema sveučilišnom „Pravilniku o stegovnoj odgovornosti studenata" nastavnik ima pravo studenta prijaviti za ometanje izvođenja nastave ili provjere znanja, odnosno za nedolično ponašanje prema nastavnicima, studentima i zaposlenicima.

Tablica 1. Moguće aktivnosti tijekom semestra, „pragovi", preporučeni udio pojedinačne aktivnosti u ukupnom broju bodova ostvarivih tijekom semestra i sl.

<table>
<tr><th>Moguće aktivnosti tijekom semestra</th><th>Maksimum bodova po uspješno položenoj aktivnosti (nastavnik određuje maksimum unutar dolje navedenog raspona)</th><th colspan="2">Minimalno potreban uspjeh iz aktivnosti da bi se smatrala uspješno položenom („prag")</th><th>Trajanje uspješno položene aktivnosti[1]</th><th>Trajanje bodova iz aktivnosti[2]</th><th>Nadoknada u slučaju neuspješno položene aktivnosti</th><th rowspan="11">Maksimalan zbroj bodova ostvarenih tijekom semestra mora biti fiksan za predmet, i to u rasponu od 40 do 70 bodova (v. Tablicu 2)
∑⇒</th></tr>
<tr><td>Pohađanje nastave (PR+AV+ +KV+LV)</td><td>od 0 do 10</td><td colspan="2">Ukupno (PR+AV+KV+LV) minimalno 70% nazočnosti[3,4,5].</td><td rowspan="6">Trajno</td><td rowspan="6">Do početka sljedećeg ciklusa nastave iz predmeta</td><td>Potrebno sljedeće ak. godine ponovno pohađati nastavu[6]</td></tr>
<tr><td>LV/KV[7]</td><td>od 0 do 30</td><td colspan="2">100 % kolokviranih vježbi</td><td>Moguće za do 30% vježbi[8]</td></tr>
<tr><td>Domaće zadaće</td><td>od 0 do 30</td><td colspan="2" rowspan="3">0 % do 50 % bodova</td><td rowspan="3">Moguće za do 20% bodova[h]</td></tr>
<tr><td>Seminarski rad</td><td>od 0 do 30</td></tr>
<tr><td>Dodatne aktivnosti[9]</td><td>od 0 do 30</td></tr>
<tr><td rowspan="2">Kontrolne zadaće[10]</td><td rowspan="2">od 0 do 50</td><td>Iz svake pojedinačno</td><td>Od 20 % do 50 %</td><td rowspan="2" colspan="2">Prvi sljedeći ispitni rok</td><td rowspan="2">Pismeni ispit (v. redak ispod za detalje)</td></tr>
<tr><td>Kumulativno</td><td>50 %</td></tr>
<tr><td>Pismeni ispit[11]</td><td>Jednako broju bodova za aktivnost „Kontrolne zadaće" [12]</td><td colspan="2">50 %[13]</td><td colspan="3">Na tekućem ispitnom roku</td></tr>
</table>

[1] Za vrijeme navedenog trajanja se smatra da je aktivnost uspješno položena (pa i u slučaju da je isteklo vrijeme „trajanja bodova iz aktivnosti", v. sljedeću fusnotu).

[2] Za vrijeme navedenog trajanja se računaju bodovi ostvareni iz aktivnosti , odnosno nakon isteka navedenog trajanja se bodovi izjednačavaju sa nulom, ali se aktivnost i dalje smatra uspješno položenom sve dok ne istekne vrijeme „trajanja uspješno položene aktivnosti" (v. prethodnu fusnotu).

[3] Navedeni prag se ne odnosi na izvanredne studente. Na polaznike Razlikovnih obveza odnosi se postotak definiran za svaki pojedinačni predmet, a koji može biti manji od 70%. Za predmete s konzultativnim izvođenjem obavezan je dolazak na barem pet termina konzultacija.

[4] Ovo je ujedno prag i za potpis u indeks (ovjera „urednog izvršavanja obveza").

[5] Za PR, i isto tako AV, nastavnik ne može tražiti više od 70% nazočnosti.

[6] U slučaju opravdanog izostanka s nastave, nastavnik studentu može odobriti nadoknadu: PR i AV (moguće do 50% sati) u obliku veće angažiranosti na nekoj od ostalih aktivnosti ili na nekoj dodatnoj aktivnosti, za LV i KV (moguće do 30% vježbi) kako je opisano pod fusnotom „h".

[7] Obavezno provoditi ako u izvedbenom planu postoje laboratorijske ili konstrukcijske vježbe. Kolokviranje LV/KV podrazumijeva sljedeće: napisana/popunjena priprema za svaku vježbu, uspješno odrađena svaka vježba, napisan/popunjen izvještaj za svaku vježbu, uspješno položene provjere znanja iz izvještaja (prag za provjere znanja iznosi 50%). Studenti ne mogu nadoknaditi vježbe na kojima nisu bili nazočni iz neopravdanih razloga. Nenapisana/nepopunjena priprema se smatra jednakom neopravdanom izostanku s vježbi, tj. student nema pravo prisustvovati vježbi, te taj izostanak može nadoknaditi tek sljedeće ak. godine. Neuspješna provjera znanja iz priprema, odnosno netočno popunjena/napisana priprema smatra se jednakom opravdanom izostanku s vježbi, tj. student nema pravo prisustvovati vježbi, ali može nadoknaditi vježbu.

[8] Potrebno nadoknaditi najkasnije prije početka prvog sljedećeg ispitnog roka (iznimno, ako je riječ o nekoj od specifičnih dodatnih aktivnosti, npr. praktični rad u laboratoriju, projektni zadatak, i sl., nastavnik može studentima odobriti duži rok za nadoknadu ako za to postoje opravdani razlozi). Neuspješna nadoknada ili veći iznos nadoknade može se odraditi tijekom sljedećeg ciklusa nastave iz predmeta. Pritom se odrađuju samo neizvršeni dijelovi aktivnosti (npr. ponovno se odrađuju samo neodrađene LV/KV, popravlja se prethodno započeti seminarski rad, itd.).

[9] Dodatne aktivnosti mogu biti grupni zadaci na predavanjima, studentske prezentacije, praktični rad u laboratoriju, projektni zadaci i sl.

[10] Obavezno provoditi ako u izvedbenom planu postoje auditorne vježbe kao oblik provođenja nastave. Tijekom semestra se organiziraju po dvije kontrolne zadaće. Kod ove neuspješno odrađene aktivnosti student iznimno ima pravo prijave ispita kako bi mogao pristupiti pismenom ispitu kao nadoknadi za ovu aktivnost.

[11] Pismeni ispit nije aktivnost tijekom semestra, nego je nadoknada za nepoložene kontrolne zadaće. Student može pristupiti pismenom ispitu jedino ako je uspješno položio ostale aktivnosti.

[12] Nakon uspješno položenog pismenog ispita i završnog usmenog ispita, pod aktivnost kontrolnih zadaća evidentira se broj bodova ostvarenih na pismenom ispitu.

[13] Nositelj predmeta na početku ak. godine definira je li pismeni ispit eliminacijski, tj. smije li student pristupiti usmenom dijelu ispita i ako nije uspješno položio pismeni ispit. Ako student ispit polaže pred ispitnim povjerenstvom (8. izlazak ili prigovor na ocjenu), povjerenstvo pregledava pismeni ispit koji ne mora biti eliminacijski, ali se od studenta u svakom

Tablica 2. Utvrđivanje ukupnog broja bodova (UBB) i konačne ocjene

<table>
<tr><td rowspan="2">Zbroj bodova ostvarenih tijekom semestra
$\sum\Rightarrow$</td><td rowspan="2">od 40 do 70 bodova</td><td rowspan="2">Zbroj bodova ostvarenih tijekom semestra i bodova na završnom usmenom ispitu
$\sum\Rightarrow$</td><td>Ukupan broj bodova (UBB)</td><td rowspan="5">Utvrđivanje ocjene na temelju UBB
$\Rightarrow$</td><td>UBB</td><td>Konačna ocjena</td></tr>
<tr><td rowspan="4">100 bodova</td><td>90≤UBB≤100</td><td>izvrstan (5)</td></tr>
<tr><td rowspan="2">Završni usmeni ispit[14]</td><td rowspan="2">od 60 do 30 bodova</td><td>75≤UBB<90</td><td>vrlo dobar (4)</td></tr>
<tr><td>60≤UBB<75</td><td>dobar (3)</td></tr>
<tr><td>UBB<60</td><td>dovoljan (2)</td></tr>
</table>

NAPOMENE VEZANE UZ PROVJERE ZNANJA I IZVEDBENI PLAN:

- kontrolne zadaće se trebaju realizirati unutar ukupno 16, odnosno 32 sata nastave za AV sa 15, odnosno 30 sati prema izv. planu. To povećanje satnice je moguće samo ako ga je nastavnik najavio satničarima najkasnije nakon odrađenih 8, odnosno 16 sati AV. Tijekom semestra se organiziraju po dvije kontrolne zadaće u trajanju od 45 do 60 minuta za predmete sa 15 sati AV, odnosno u trajanju od 60 do 90 minuta za predmete sa 30 sati AV.
- nadoknade LV/KV sa 15, odnosno 30 sati trebaju se realizirati unutar najviše 16, odnosno 32 sata nastave. To povećanje satnice je moguće samo ako ga je nastavnik najavio satničarima najkasnije nakon odrađenih 50% satnice. Ako je to povećanje nedovoljno za nadoknade, nadoknada se može provesti u obliku provjere znanja iz priprema i izvještaja iz nekolokviranih vježbi u terminima koje je nastavnik dogovorio s pojedinačnim studentima.
- provjere znanja iz LV/KV (iz priprema i izvještaja) trebaju biti provedene tijekom termina LV/KV (npr. na početku ili na kraju svakog pojedinačnog termina) ili u posebnim terminima. Pritom satnica posebnih termina ne ulazi u izvedbeni plan niti se računa pod realizacijom izvedbenog plana. Za posebne termine će satničari osigurati mjesto u rasporedu pod uvjetom da nositelj predmeta na početku ak. godine najavi održavanje posebnih termina za provjeru znanja iz LV/KV i navede trajanje svakog termina.
- budući da nisu obavezni, kolokviji (vezani za usmeni ispit) realiziraju se izvan satnice predviđene izvedbenim planom, tj. ta satnica ne ulazi u izvedbeni plan niti se računa pod realizacijom izvedbenog plana. Pritom će satničari osigurati potrebne termine i objaviti ih u rasporedu pod uvjetom da nositelj predmeta na početku ak. godine najavi održavanje dva kolokvija tijekom semestra i navede trajanje svakog kolokvija.

---

slučaju očekuje i da na usmenoj provjeri znanja pokaže i znanje koje je bilo potrebno za uspješno polaganje pismenog dijela ispita.

[14] Ispitni prag na završnom usmenom ispitu iznosi 50% uspješnosti na završnom usmenom ispitu. Završni usmeni ispit se može održati i u obliku dva kolokvija tijekom semestra (prag za svaki pojedinačni iznosi od 20% do 50%, kumulativno 50%). Uspješno položeni kolokviji vrijede prvi sljedeći ispitni rok. Pritom, u slučaju da je student na jednom kolokviju imao uspješnost najmanje 50%, ali kumulativno manje od 50%, nastavnik može odobriti studentu da na usmenom ispitu odgovara parcijalno, tj. samo tematske cjeline nepoloženog kolokvija.

Za usmeni ispit (odnosno kolokvije tijekom semestra) ispitivač treba definirati u prosjeku 2 do 5 ispitnih pitanja za svaki sat predavanja. Ispitivač nije dužan ispitivati strogo prema ispitnim pitanjima, odnosno ispitna pitanja služe kao smjernice studentima za pripremu za usmeni ispit (odnosno kolokvije tijekom semestra).