ВИСОКА ТЕХНИЧКА ШКОЛА
СТРУКОВНИХ СТУДИЈА
у Новом Саду

COURSE CATALOGUE

INFORMATION TECHNOLOGIES

Master applied studies

NOVI SAD
November 2023

Table 5.2.a – Course catalogue of the INFORMATION TECHNOLOGIES master study programme

No.	Course code	Course title	Field of Study	Semester	Lecture	Lab	Other Forms of Teaching	Other Hours	ECTS
1	2M19	Software engineering	Electrical and Computer Engineering, Mathematical Sciences	1	4	3			10
2	2M18	Java programming	Electrical and Computer Engineering	1	4	3			10
3	Elective option 1 (select 1 of 2)								
	2M24	Applied risk modelling methods	Mechanical Engineering, Electrical and Computer Engineering	1	3	3			8
	2M25	Data acquisition and control systems	Electrical and Computer Engineering	1	3	3			8
4	2M32	Artificial intelligence algorithms	Electrical and Computer Engineering	2	3	2			8
5	2M20	Advanced web programming	Electrical and Computer Engineering, Mathematical Sciences	2	4	2			8
6	2M21	Functional programming	Electrical and Computer Engineering, Mathematical Sciences	2	4	2			8
7	2M28	Professional master practice 1	Electrical and Computer Engineering	2				6/0	4
8	Elective option 2 (select 1 of 2)								
	2M35	Business ethics	Cultural Studies and Communications, Industrial Engineering and Engineering Management	2	2	1			4
	2M36	Modern methods of education	Cultural Studies and Communications, Applied Arts and Design	2	2	1			4
9	2M22	Mobile application development	Electrical and Computer Engineering	3	4	3			8
10	2M23	Advanced database concepts	Electrical and Computer Engineering	3	4	2			8
11	Elective option 3 (select 1 of 2)								
	2M26	Software methods of experimental modelling	Electrical and Computer Engineering, Mechanical Engineering	3	4	3			8
	2M27	CAD/CAE/CAM and CIM systems	Mechanical Engineering	3	4	3			8
12	2M29	Professional master practice 2	Electrical and Computer Engineering	4				6/0	4
13	2M30	Preliminary master thesis	Electrical and Computer Engineering	4				0/20	16
14	2M31	Master thesis	Electrical and Computer Engineering	4				4/0	16

Table 5.2. – Course specification

Study programme: Information technologies

Course title: Software engineering

Teacher: Tanja Krunić

Course status: Compulsory

Number of ECTS: 10

Prerequisite course: None

Course objective

The aim of the course is for students to learn what a software product and software process are, what software process models are, how to do specification, modeling, design and implementation, verification and validation, also testing and debugging of a software product, as well as how to manage a software project.

Learning outcomes

Students are equipped to actively participate in all phases of software product development and manage software projects.

Course contents

Theoretical teaching

Introduction to software engineering. Software product. Software process, methods and tools. Software process models. Waterfall model. Evolutionary model. Formal development model. Reuse-oriented model. Incremental development model. Agile software development methods. Features of a software project. Software project management. Software manager's duties. Requirements and specifications. Documents describing requirements. The concept and importance of software application architecture. Types of software architecture: Client-server, peer-to-peer, Model-View-Controller (MVC), Microservices, event driven, layered and hexagonal architecture. Requirements prototyping. System modelling using UML diagrams. Designing a graphical user interface. User interaction with the system. Component-based development. Git – version control system. GitHub. Repository. Creating and configuring a repository. Public and private repositories. Cloning a repository. Uploading documents to a remote repository. Committing changes. Checking status. Push and Pull. Adding new branches. Collaboration within a project team using Git. Managing the software development process using the specialized application JIRA Software verification and validation. Software testing. Writing unit tests: Jest library. Maintenance and evolution. Software change strategies. Writing software documentation.

Practical teaching

Practical work in teams at all stages of software development, from requirements, through specifications to software testing. Software development using agile methods. Student collaboration on the project through version control systems. Management of software development phases using the specialized JIRA application.

Literature

Krunić T., Softversko inženjerstvo – materijal sa predavanja u elektronskom obliku, 2022
Popović J., Osnove softverskog inženjerstva, Računarski fakultet, Beograd, 2019
Popović J., Testiranje softvera u praksi, Računasrki fakultet, Beograd, 2012
Freeman E., Robson E, Um caruje:Projektni obrasci, Mikroknjiga, 2021
Martin C. R., Čisto agilno – razvoj softvera, Kompjuter biblioteka, 2021

Active teaching classes: 105	**Theoretical teaching: 60**	**Practical teaching: 45**

Teaching methods

Monologic, interactive

Assessment (maximum number of points 100)

Pre-exam obligations	**Points**	**Course exam**	**Points**
Active participation in classes		Written exam	50
Practical work		Oral exam	
Interim exam/s			
Seminar paper/s	50		

Back to top

Study programme: Information technologies

Course title: Java programming

Teacher: Lazar Kopanja

Course status: Compulsory

Number of ECTS: 10

Prerequisite course: None

Course objective

Introduction to the concept of a virtual machine and the principles of object-oriented programming in Java. Acquisition of skills in developing and implementing Java desktop and web applications, as well as creating supporting documentation.

Learning outcomes

Students are equipped to independently and collaboratively develop, test, and document desktop and web Java applications integrated with databases.

Course contents

Theoretical teaching

The concept of a virtual machine. Object-oriented programming in Java: classes, objects, encapsulation, methods, references, inheritance, polymorphism, abstract classes, interfaces. Exception handling. Input-output subsystem. Graphical user interface: AWT, Swing (labels, buttons, text fields, sliders, panels, menus, lists, etc.). Generics. Java Collections Framework. Working with relational databases – JDBC. Servlet technology. Session tracking. Java Server Pages (JSP). JSP expressions. JSP scriptlets. JSP declarations. JSP directives. JavaBeans. JavaServer Faces. Facelets. Using the Javadoc tool for generating Java API documentation in HTML format.

Practical teaching

It aligns with the lecture topics through the implementation of introductory exercises, followed by the development of project-based seminar papers, completed either individually or in teams. The course project consists of an independently developed and tested Java application featuring a graphical user interface and a relational database backend.

Literature

Cadenhead, R., Naučite samostalno -Java za 21 dan, Kompjuter biblioteka, Beograd, 2020

Lavieri E., Verhas P, Java 9, Kompjuter biblioteka, 2018

Karanam R., Naučite Spring 5, Kompjuter biblioteka, 2017

Bloch J., Java efikasno: 90 načina kako da pišete bolje programe na Javi, Mikroknjiga, 2021

Schildt H., Java kompletan priručnik,Sveobuhvatan opis jezika Java, Mikroknjiga, 2018

Active teaching classes: 105	**Theoretical teaching: 60**	**Practical teaching: 45**

Teaching methods

Monologic, dialogic, interactive

Assessment (maximum number of points 100)

Pre-exam obligations	**Points**	**Course exam**	**Points**
Active participation in classes		Written exam	50
Practical work		Oral exam	
Interim exam/s			
Seminar/s	50		

Back to top

Study programme: Information technologies

Course title: Applied risk modelling methods

Teacher: Božo Ilić, Nenad Janjić, Branko Savić

Course status: Elective

Number of ECTS: 8

Prerequisite course: None

Course objective

Gaining knowledge about different types of risks, as well as methods for their modelling, with an emphasis on computer modelling.

Learning outcomes

Students will be able to recognize different types of risks and model them using various methods, especially computer modelling.

Course contents

Theoretical teaching

Research: Stages of research work. Research methods. Empirical method. Observation method. Experimental research. Research using the modelling method. Mathematical modelling: Empirical, deterministic, population and stochastic models. Application of the Monte Carlo method in modelling stochastic processes. Statistical Research: Application of Computers in Statistical Analysis. Statistical Population, Variable, and Sample. Simple correlation and regression analysis. Probability theory: Classical definition of probability. Statistical definition of probability. computer modelling and process simulation: Application of the MATLAB software package in computer modelling and process simulation. Ecological modelling: Phases of ecological model development. Mathematical model formulation. Computer model development. Computer model verification (program testing). Model validation and application. Risk: Concept of risk. Risk assessment. System modelling for quantitative risk assessment. Environmental risk assessment methods. Examples of risk modelling: Environmental impact of a product or manufacturing process. Life Cycle Assessment (LCA) of a Product. Environmental risk assessment. Model for assessing environmental risks caused by the NATO bombing of the Republic of Serbia in 1999. Forest fire risk management. Computer modelling and simulation of forest fire spread. Application of GIS model in hazard and flood risk mapping. Financial risk modelling of a wind farm project.

Practical teaching

Assignments related to the topics covered in the lectures are performed.

Literature

Ilić, B., Primenjene metode modelovanja rizika 1 i 2, skripta u elektronskom obliku, Visoka tehnička škola strukovnih studija u Novom Sadu, 2018

Petrić, M., Statistika, Visoka tehnička škola strukovnih studija u Novom Sadu, 2015

Ž. Adamović, V. Veljković, D. Malešević, Metodologija istraživanja u industriji i obrazovanju, Društvo za energetsku efikasnost, Banja Luka, 2010

Ž. Adamović, Đ. Nadrljanski, Metodologija naučno-istraživačkog rada – Statističke metode u istraživanju, Društvo za tehničku dijagnostiku Srbije, Pedagoški fakultet, Sombor, 2011

A. Veljović, M. Zahorjanski, Modeliranje informacionih sistema, CET, Beograd, 2016

Active teaching classes: 90	**Theoretical teaching: 45**	**Practical teaching: 45**

Teaching methods: Monologic, dialogic, interactive

Assessment (maximum number of points 100)

Pre-exam obligations	**Points**	**Course exam**	**Points**
Active participation in classes	10	Written exam	50
Practical work		Oral exam	
Interim exam/s	40		
Seminar paper/s			

Back to top

Study programme: Information technologies

Course title: Data acquisition and control systems

Teacher: Branislav Santrač

Course status: Elective

Number of ECTS: 8

Prerequisite course: None

Course objective

Acquiring knowledge and understanding of the fundamental theoretical principles in the field of measurement and data acquisition. Understanding the place and role of acquisition and control systems within the overall structure of information systems. Essential concepts and block configurations of the field, and their reinforcement through practical assignments.

Learning outcomes

Understanding the core concepts of the field. The ability to specify system elements based on measurement and acquisition requirements, regarding both the computer information component and the hardware execution component. Knowledge of the specific characteristics of individual system components and their integration into a real-world

Course contents

Theoretical teaching

Introductory lecture: organization and content of the course, method of taking the exam. About measurement acquisition systems: history, needs, types. Types of architecture: centralized, distributed, WASCAD systems. Quantities being measured, types of sensors. SCADA system concept. Elements of SCADA system. Measuring equipment and executive bodies. Remote U/I (input/output modules). Remote stations. Communication systems. Central station. SCADA system architecture. SCADA systems in power engineering and automation. SCADA systems in thermal power plants. SCADA systems and commercial PLC controllers. Their mutual similarities and the application of PLC as a form of SCADA. Adaptation of SCADA to the Internet of Things system. Security aspects of the system, solutions. Advantages and disadvantages. Examples of SCADA application in industry, medicine, transport, agriculture and military applications. Common elements and differences. Programming software. Types of programming languages. Programming levels: ladder diagrams, programming languages.

Practical teaching

Presentation of relevant assignments from all topics, group problem-solving exercises. Company site visits. Seminar paper development, presentation, and defense before an audience.

Literature

Proficy iFIX 6.5 Building a SCADA System, General Electric manual, 2021

Calculations Toolbox, SPL, SPIDER manual, 2022

ISIS skripta - SSADA, FTN, 2017

SCADA System Fundamentals, Tracy Adams, P.E., 2014

Active teaching classes: 90	**Theoretical teaching: 45**	**Practical teaching: 45**

Teaching methods

Monologic, dialogic, interactive.

Assessment (maximum number of points 100)

Pre-exam obligations	**Points**	**Course exam**	**Points**
Active participation in classes		Written exam	50
Practical work		Oral exam	
Interim exam/s			
Seminar paper/s	50		

Back to top

Study programme: Information technologies

Course title: Artificial intelligence algorithms

Teacher: Lazar Kopanja

Course status: Compulsory

Number of ECTS: 8

Prerequisite course: None

Course objective

The aim of the course is to enable the student to independently apply and develop artificial intelligence algorithms.

Learning outcomes

Based on the mastered course material, the student is capable of applying and developing artificial intelligence algorithms.

Course contents

Theoretical teaching

Artificial Intelligence – concept and objectives. History of artificial intelligence development. Classification and determinants of artificial intelligence systems. Real-world applications of artificial intelligence. Fundamentals of mathematical logic. Non-classical mathematical logics. Intuition of artificial intelligence. Basics of search. Identifying problems solvable by search algorithms. Designing basic search algorithms for problem-solving. Designing heuristics for guided search. Designing guided search algorithms. Designing search algorithms for two-player games. Evolutionary algorithms. Problem-solving with evolutionary algorithms. Understanding the genetic algorithm life cycle. Developing genetic algorithms for optimization problems. Advanced evolutionary approaches. Swarm intelligence: Ants. Swarm intelligence: Particles. Problem-solving using swarm intelligence algorithms. Design and application of the Ant Colony Optimization algorithm. Design and application of the Particle Swarm Optimization algorithm. Machine learning. Artificial neural networks. Understanding the inspiration and intuition behind artificial neural networks. Identifying problems solvable by artificial neural networks. Understanding and applying backpropagation for network training. Designing artificial neural network architectures for various problems. Reinforcement learning using Q-learning.

Practical teaching

Analysis, application and development of specific artificial intelligence algorithms.

Literature

Hurbans, R., Algoritmi veštačke inteligencije, Kompjuter biblioteka, Beograd, 2021

Raschka S., Mašinsko učenje PyTorch i Scikit-Learn, Kompjuter biblioteka, 2022

Geron A, Mašinsko učenje: Scikit-Learn, Keras i TensorFlow: koncepti, alati i tehnike za izgradnju inteligentnih sistema, Mikro knjiga, 2021

Raschka S., Mirjalili V, Python mašinsko učenje, prevod trećeg izdanja, Kompjuter biblioteka, 2020

Explorations in Artificial Intelligence and Machine Learning, CRC Press Free eBook, 2018

Active teaching classes: 75	**Theoretical teaching: 45**	**Practical teaching: 30**

Teaching methods

Monologic, dialogic, interactive.

Assessment (maximum number of points 100)

Pre-exam obligations	**Points**	**Course exam**	**Points**
Active participation in classes	10	Written exam	50
Practical work		Oral exam	
Interim exam/s	20		
Seminar paper/s	20		

Back to top

Study programme: Information technologies

Course title: Advanced web programming

Teacher: Tanja Krunić

Course status: Compulsory

Number of ECTS: 8

Prerequisite course: None

Course objective: Introducing students to advanced web application development techniques: Application of React in the development of dynamic web applications and their styling using the CSS preprocessor LESS.

Learning outcomes: Students use React to develop dynamic web applications, as well as advanced techniques for styling and adapting web pages to various devices using the LESS CSS preprocessor.

Course contents

Theoretical teaching

CSS preprocessors. LESS. Working environments for LESS. Tools for translating LESS into CSS (online services and work environments, Crunch framework). Automated translation of LESS to CSS. Using variables and operations on variables. Nesting. Mixins. Application of mixins for better code organization. Parametric mixins. Inheritance – extend and extend all. Flow control – mixin guards. Recursion. Importing code from other files. Splitting the style into files by purpose and importing them into the main document. Import by reference. Using import by reference to download certain classes from other projects or class libraries such as Bootstrap, Foundation and others. Application of LESS in adapting content to mobile phones and tablets. Block element modifier (BEM) class naming convention that allows writing styles in a modular and maintainable way. Application of BEM in LESS. Application advantages. Block and its properties. Element. Modifier. Link between: block name, element name, and element modifier name. Node.js – Installation. Node package manager, basic commands and their application. GitBash – implementation. Running .js files in GitBash. Running files in a console window and web browser. React.js. JSX. Implementing React cdn. Using React within an HTML page. ReactDOM.render method and its application. Using components in React. A class and functional approach to writing code. Passing parameters. Using properties (props). Using style objects. Working with states – React state. Separating components into different files – creating an application. Working with events in React. Working with forms. Processing form data. Validation. Creating a navigation menu using React Router. Creation of Single page application. React Hooks and their application.

Practical teaching

Applying the above mentioned advanced features of React in creating dynamically generated web pages and styling them using LESS.

Literature

Krunić T., Materijal sa predavanja iz predmeta Napredno veb programiranje u elektronskom formatu, 2019

Činatambi K., Naučite React, Mikro knjiga, 2018

Grebe S., GraphQL i React ful stek razvoj, Kompjuter biblioteka, 2019

Boduch A, React i React Native: Izgradnja međuplatformskih JavaScript aplikacija, Kompjuter biblioteka, 2023

Less, Tutorials Point, free eBook, 2017

Active teaching classes: 90	**Theoretical teaching: 60**	**Practical teaching: 30**

Teaching methods: Monologic, dialogic, interactive.

Assessment (maximum number of points 100)

Pre-exam obligations	**Points**	**Course exam**	**Points**
Active participation in classes		Written exam	50
Practical work – project presentation	50	Oral exam	
Interim exam/s			
Seminar paper/s			

[Back to top]

Study programme: Information technologies

Course title: Functional programming

Teacher: Tanja Krunić

Course status: Compulsory

Number of ECTS: 8

Prerequisite course: None

Course objective

Introducing students to the basic principles of functional programming with applications in JavaScript in order to create interactive web pages with transparent code that is fast to execute, easy to maintain and extendable.

Learning outcomes

Students apply the basic principles of functional programming when creating interactive web pages in JavaScript.

Course contents

Theoretical teaching

Concept and principles of functional programming. Advantages of functional programming: Clear and reviewed code, modularity, reusability, independence, accuracy. Comparison of functional and object-oriented programming methods. Functional programming in JavaScript. The concept of a true (source) function. Closures. Higher order functions. Anonymous functions. Self-calling functions. Lambda expressions and arrow functions. Chain method. Recursions and their application – the principle of avoiding loops. The forEach(), map(), filter(), and reduce() commands and their role in avoiding loops. Callback functions and their application. Application of Callback function to array elements. The concat() function for merging arrays without using loops. Application of native array functions: reverse(), sort(), every(), some(). Using the Array, prototype constructor to define array functions without using loops. Application of the spread operator on arrays. Implementation of functional programming techniques - Function manipulation: The apply() and call() functions and the "this" keyword. Currying (transforming functions with multiple arguments into a function with a single argument). Application of the bind() method. Function composition. The compose() function. Comparing the application of function composition and method chaining. Event manipulation in functional programming. Connecting functional and object-oriented programming in JavaScript. Inheritance in functional programming. Improving program performance using Lazy Evaluation – the Call-by-Need Principle and Deferred Execution. Code refactoring techniques aimed at improving code quality.

Practical teaching

Applying the principles of functional programming in JavaScript to create interactive web pages with clean and transparent code that executes at optimal speed.

Literature

Krunić T., Materijal sa predavanja iz predmeta Funkcionalno programiranje u elektronskom formatu, 2019

Normand E., Otkrivanje jednostvanosti, Funkcionalno programiranje za kroćenje složenog softvera, Kompjuter biblioteka, 2021

Klausen K., Pet linija koda. Kako i kada refaktorisati, Kompjuter biblioteka, 2022

Widman J., Naučite funkcionalno programiranje, CET, 2023

Lemaire M., Refaktorisanje po meri: Preuzmite kontrolu nad svojim kodom, Mikro knjiga, 2021

Fowler M., Refaktorisanje: Poboljšanje dizajna postojećeg koda, CET, 2020

Active teaching classes: 90	**Theoretical teaching: 60**	**Practical teaching: 30**

Teaching methods

Monologic, dialogic, interactive.

Assessment (maximum number of points 100)

Pre-exam obligations	**Points**	**Course exam**	**Points**
Active participation in classes		Written exam	50
Practical work		Oral exam	
Interim exam/s	50		
Seminar paper/s			

Back to top

Study programme: Information technologies

Course title: Professional master practice 1

Teacher: Head of the study programme

Course status: Compulsory

Number of ECTS: 4

Prerequisite course: None

Course objective

Acquisition, application and improvement of current practical knowledge and skills related to the study programme from the subject area dealt with in the first year of the master study, which is carried out in the selected host company, as well as introduction to the work process of the company.

Learning outcomes

Practical experience gained in the application of current knowledge and skills acquired during the first year of master studies, on specific work processes in the selected host company.

Course contents

Practical teaching

Professional master practice 1 is carried out in the 1st year of professional master studies, lasting 90 hours. It is performed in companies dealing with production, service and other activities, according to general and individual programme contents, agreed between the co-mentor from the host company, the mentor in charge of practice from the School and the student. The student can also perform Professional master practice 1 at the Higher Education Technical School of Professional Studies in Novi Sad, according to a predetermined and approved plan.

Forms related to Professional master practice 1 are available to students registered to take the course.

The student in Professional master practice 1 performs general and special tasks. General tasks imply that the student learns about: the history of the company, the organisational structure and the production programme, that is, the work processes in the company.

Student's professional tasks that should be performed during Professional master practice 1 are defined by the co-mentor from the company and the mentor. They refer to thematic units the student has attended and passed in the courses during the first year of master studies, so now that knowledge is applied in practice in the selected company.

The assignments provide adequate opportunities for the student to demonstrate abilities in several categories: application of professional knowledge; practical skills; analytical skills, the ability to solve concrete problems; innovation and originality; development skills; time management; written and oral expression, all with the aim of training for the future profession. The co-mentor in the company regularly cooperates with the student, instructs him and monitors their work. During the practice, the student keeps the internship diary.

After completing the Professional master practice 1, the student submits a report that, in terms of content and form, corresponds to the instructions defined at the beginning of the internship.

Literature

Technical documentation from the company and other relevant literature agreed between the student, the co-mentor from the company and the mentor.

Active teaching classes: 90

Teaching methods:

Practical, laboratory, individual work.

Assessment (maximum number of points 100)

Pre-exam obligations	**Points**	**Course exam**	**Points**
Active participation in classes		Written exam	
Practical work	50	Oral exam	
Interim exam/s		Practical exam	50
Seminar paper/s			

Back to top

Study programme: Information technologies

Course title: Business ethics

Teacher: Petra Balaban, Jovana Kopanja

Course status: Elective

Number of ECTS: 4

Prerequisite course: None

Course objective

Introducing students to the concepts of ethics and morality in the business environment. The course covers the fundamentals of philosophical ethical theories and their application in business. Through numerous examples, students will become familiar with the practical application of ethics in various problematic situations that have become prominent in recent years: business ethics, whistleblowing, data privacy, conflict of interest, bribery and corruption, and discrimination. Additionally, students are introduced to the principles of corporate social responsibility.

Learning outcomes

The student is equipped to analyse the impact of economic challenges on ethics. They possess the skills to recognize and resolve ethical problems in business. The student is encouraged to analyse their own value system, recognizing the importance of individual values in the business environment and the workplace. The student acts in a socially responsible manner.

Course contents

Theoretical teaching

Corporate culture – concept, features. The importance of symbols for cultural development. The relationship between economy and culture. The concept and principles of personal ethics. General and business ethics. Fundamental principles of business ethics. Development stages of business ethics. Ethics as a segment of corporate culture and ethical principles. Entrepreneurial business culture as a prerequisite for business ethics. Codes of ethics. Business morality. Obligations and rights of employees as a framework for moral challenges. Business ethics and moral pluralism. Conflict of interest and business ethics. Moral dilemmas of employees. Bribery and corruption. The emergence of whistleblowers. Data privacy. Discrimination. Establishing and applying moral standards in business. Ethics in a global world. The subject of business ethics as an academic discipline. Stakeholders and stockholders of ethical reasoning. The concept of social responsibility. Types, concepts, and models of social responsibility. Levels of social responsibility. Social responsibility in marketing and public relations. Social responsibility and environmental protection.

Practical teaching

Analysis of practical examples from the material presented in the lectures. Active participation of students in discussions.

Literature

Vuković M., Voza D., Vuković A., Poslovna etika, Visoka škola strukovnih studija za mendžment u saobraćaju, 2010
Elaković, S., Poslovna etika i komuniciranje, Univerzitet Singidunum, Beograd, 2011
Miljević M., Poslovna etika i komuniciranje, Univerzitet Singidunum, 2010
Cvijanović D., Mihajlović B., Pejanović R., Poslovna etika i komuniciranje, Institut za ekonomiku poljoprivrede, 2012
Pavić Ž., Etika i psolovne komunikacije, Univerzitet Singidunum, 2011
Jalešnjak B., Krkač K., Poslovna etika, Korporacijaksa društvena odgovornost i održivost, Mate, 2016

Active teaching classes: 45	**Theoretical teaching: 30**	**Practical teaching: 15**

Teaching methods:

Monologic, dialogic, interactive, demonstrative.

Assessment (maximum number of points 100)

Pre-exam obligations	**Points**	**Course exam**	**Points**
Active participation in classes		Written exam	50
Practical work		Oral exam	
Interim exam/s	50		
Seminar paper/s			

Back to top

Study programme: Information technologies

Course title: Modern methods of education

Teacher: Jovana Kopanja, Karolina Mudrinski-Palanački

Course status: Elective

Number of ECTS: 4

Prerequisite course: None

Course objective

Introducing students to modern educational technologies and standards. Enabling students to apply modern technologies and methods in order to present their own engineering results.

Learning outcomes

Students understand the applications of information and communication technologies, and adopt methods for their selection as well as the relevant standards applied in the process of presenting and evaluating results. The student recognizes the importance of applying modern educational technology and is familiar with the pedagogical and psychological effects caused by its implementation.

Course contents

Theoretical teaching

History of educational technologies and the concept of e-learning. Modern technologies and tools in education. Types of modern education. Information and communication technologies (ICT) and modern education. Hardware and software infrastructure. E-learning platforms (LMS – Learning Management System) used for the creation, administration, and tracking of training and education. Intelligent tutoring systems. E-learning standards. Standards for the presentation of electronic materials. Standards for the presentation of the instructional process. Open education. Pedagogical-psychological effects of applying modern technologies in education. Problems in the application of modern educational technology and mechanisms for overcoming them. Modern educational technologies and media literacy.

Practical teaching

Development of a project assignment from a selected area of modern education. Analysis and evaluation of knowledge within the adopted and enhanced education system.

Literature

Đorđević J., Individualizacija i inoviranje nastave i učenja u školi 21. veka, Pedagoška stvarnost, Novi Sad, 2009

Ivić I., Pešikan A., Antić S., Aktivno učenje 2, Institut za psihologiju, Beograd, 2001

Aggarwal J. C., Kaur M., Jaswal S.S., Bajwa Ch., Chabra S., Contemporary issues in education, Tripura University, 2017

Illeris K., Contemporary theories of learning, Routledge Taylor & Franscis group, 2009

Tirri K., Contemorary teacher education, MDPI, 2021

Active teaching classes: 45	**Theoretical teaching: 30**	**Practical teaching: 15**

Teaching methods

Monologic, dialogic, interactive.

Assessment (maximum number of points 100)

Pre-exam obligations	**Points**	**Course exam**	**Points**
Active participation in classes		Written exam	50
Practical work		Oral exam	
Interim exam/s			
Project assignment	50		
Seminar paper/s			

Back to top

Study programme: Information technologies

Course title: Mobile application development

Teacher: Dragan Rastovac, Lazar Kopanja

Course status: Compulsory

Number of ECTS: 8

Prerequisite course: None

Course objective

The aim of the course is for students to acquire skills in designing, developing, testing, publishing and deploying applications for mobile devices and mobile systems.

Learning outcomes

Students are equipped to independently and within a team develop, test, and document Android mobile applications.

Course contents

Theoretical teaching

Introduction to mobile development and Android application development. Android versions. Integrated development environment. Eclipse. NetBeans. Android studio. Android Studio Tools. Application development tools. Error detection tools. Other more important tools. Dalvik and ART virtual machines. Using an Android emulator. Creating an Android project. Android application program logic. Activities. Services. Content providers. Fragments. Intents. Resources and resource qualifiers. The Android Manifesto. Gradle. Android user interface. Designing the user interface. Standard components of the graphical interface. Layouts. Event handling. Data persistence. Content providers. Writing data to files. Working with mobile databases. ORM. Location services. Networking. Android service development. Working with web services. SOAP and REST. Retrofit and GSON. Messaging. Publishing Android applications.

Practical teaching

Fully follows the lecture content through the development of simple tasks, followed by the development of projects/seminar papers, which can be individual or team-based. The project is an independently developed, tested, and published Android application.

Literature

Bojer, R.: Android studio IDE kuvar za razvoj aplikacija, Kompjuter biblioteka, 2016

Forrester A, Kotlin za Android aplikacije, prevod 2 izdanja, Kompjuter biblioteka, 2023

Darwin I., Android kuvar, Mikro knjiga, 2013

Android application development, Tutorialspoint, free eBook, 2014

Java fundamentals for Android development, Android ATC Team, 2017

Živković, M., Razvoj mobilnih aplikacija, Univerzitet Singidunum, Beograd, free eBook, 2020

Active teaching classes: 105	**Theoretical teaching: 60**	**Practical teaching: 45**

Teaching methods

Monologic, dialogic, interactive.

Assessment (maximum number of points 100)

Pre-exam obligations	**Points**	**Course exam**	**Points**
Active participation in classes	10	Written exam	50
Practical work		Oral exam	
Interim exam/s			
Seminar paper/s	40		

[Back to top]

Study programme: Information technologies

Course title: Advanced database concepts

Teacher: Lazar Kopanja

Course status: Compulsory

Number of ECTS: 8

Prerequisite course: None

Course objective

To introduce students to NoSQL database management systems and their advantages and disadvantages compared to relational database management systems.

Learning outcomes

Students are equipped to use NoSQL database management systems.

Course contents

Theoretical teaching

NoSQL databases. NoSQL and relational databases compared. NoSQL databases Firebase and MongoDB.
Getting to know the Firebase platform. Connecting Firebase to your existing platform. Setting up the environment. Creating a Firebase account. Creating a Firebase application. Creating a data structure. Creating a reference to a specific collection. Working with lists of objects. Using set(), update(), push(), key(), on() methods. Firebase-y data types: value, child_added, child_changed, child_removed. Firebase queries: orderByChild(), orderByKey(), orderByValue(). Data filtering: limitToFirst(), limitToLast(), startAt(), endAt(), equalTo(), orderByChild(). Authentication.
MongoDB – the basics. MongoDB document. Working with MongoDB on-premises and in the cloud. Creating a cluster. Introduction of MongoDB Shell (mongosh). Connecting to the database – Connection strings. MongoDB Query API. CRUDE operations. Creating a database using mongosh. Displaying existing bases. Changes in the existing database. Creating collections with mongosh. Working with methods: createCollection(), insertOne(), insertMany(), find(), findOne(), updateOne(), updateMany(), eleteOne(), deleteMany(). MongoDB Query operators. MongoDB Update operators. MongoDB Aggregation Pipeline. Indexing and Searching. Validation. Drivers. Connecting a Node.js application to a MongoDB database.

Practical teaching

Creating practical examples that follow the topics covered in the theoretical lectures.

Literature

DayLey, B., DayLey, B., DayLey, B., Node.js, MongoDB i Angular: integrisane alatke za razvoj veb strana, Kompjuter biblioteka, Beograd, 2018

Di Francesco H., Gao S., Isola V., Kirkbride Ph, Profesionalni JavaScript, CET, 2020

Herron D., Node.js veb razvoj, prevod petog izdanja, Kompjuter biblioteka, 2020

Learning MongoDB, Stack Overflow, Free eBook, 2017

Learning Firebase database, Stack Overflow, Free eBook, 2018

Active teaching classes: 90	**Theoretical teaching: 60**	**Practical teaching: 30**

Teaching methods

Monologic, dialogic, interactive.

Assessment (maximum number of points 100)

Pre-exam obligations	**Points**	**Course exam**	**Points**
Active participation in classes		Written exam	50
Practical work		Oral exam	
Interim exam/s			
Seminar paper/s	50		

Back to top

Study programme: Information technologies

Course title: Software methods of experimental modelling

Teacher: Božo Ilić, Nenad Janjić, Branko Savić

Course status: Elective

Number of ECTS: 8

Prerequisite course: None

Course objective

Gaining knowledge about different types of experiments, as well as methods for their modelling, with an emphasis on softwer-based modelling.

Learning outcomes

Students will be able to recognize different types of experiments and model them using various methods, with a particular focus on computational modelling.

Course contents

Theoretical teaching

Modelling: Computer-based modelling and process simulation. Application of the MATLAB software package in computational modelling and process simulation. Application of computational modelling in ecology. Programming languages that can be used for modelling. Evaluation of results obtained by computer simulation. Application of the Monte Carlo method in modelling stochastic processes. Problem solving methodology: Problem definition. Description of input and output operations. Manual problem-solving (or using a calculator) for a sample dataset. MATLAB solution development. Solution testing with various data. MATLAB environment: MATLAB windows, problem solving in MATLAB, working with matrices, operator precedence, MATLAB modules (Simulink Toolbox, Control System Toolbox, Simscape, Neural Network Blockset, Optimization Toolbox, Spline Toolbox, etc.). Simulink Modelling: Simulink libraries (Commonly Used Blocks, Continuous, Discontinuous, Discrete, Logic and Bit Operations, Look-Up Tables, Math Operations, Model Verification, Model-Wide Utilities, Ports & Subsystems, Signal Attributes, Signal Routing, Sinks, Simscapes, etc.). Creating a new model in Simulink. Basic element classes in Simulink. Examples of modelling and simulation in Simulink. Simscape Modelling: Basic libraries for modelling in Simscape. System modelling and simulation in Simscape. Structural modelling of systems in Simscape.

Practical teaching

Assignments related to the topics covered in the lectures are performed.

Literature

D. Etter, D. Kuncicky, Uvod u Matlab 7, CET, Beograd, 2005

A. Veljović, M. Zahorjanski, Modeliranje informacionih sistema, CET, Beograd, 2016

M. Zahorjanski, Zbirka zadataka iz informacionog modeliranja, CET, Beograd, 2016

G. Nestorović, Ž. Adamović, Modeliranje sklopova u Pro/Engineer-u Wildfire 5.0, TŠ Nikola Tesla, Kostolac, 2011

Ž. Adamović, Lj. Josimović, Metodologija i tehnologija naučnog istraživanja, Društvo za tehničku dijagnostiku Srbije, Beograd, 2007

Active teaching classes: 105	**Theoretical teaching: 60**	**Practical teaching: 45**

Teaching methods:

Monologic, dialogic, interactive.

Assessment (maximum number of points 100)

Pre-exam obligations	**Points**	**Course exam**	**Points**
Active participation in classes	10	Written exam	50
Practical work		Oral exam	
Interim exam/s	40		
Seminar paper/s			

Back to top

Study programme: Information technologies

Course title: CAD/CAE/CAM and CIM systems

Teacher: Vladimir Blanuša

Course status: Elective

Number of ECTS: 8

Prerequisite course: None

Course objective

Acquisition of basic knowledge in the field of CAD/CAE/CAM and CIM systems. Computer-aided design of parts, computer-aided manufacturing using CAM software, computer-aided analysis using CAE systems and computer-integrated manufacturing.

Learning outcomes

Knowledge and application of CAD/CAE/CAM and CIM systems.

Course contents

Theoretical teaching

Introduction to Computer Aided Design (CAD). Principle of CAD technology. Model and modelling, drawing and modelling, model forms, types of geometric models, geometric model forms. Modelling of parts and modelling of assemblies. General technical documentation (creation of workshop and assembly drawings). Design of machine elements using a special-purpose software system. Design and creation of technical documentation. Modelling of axles, shafts, gears. Creation of subassemblies, assemblies and assembly drawings. Automated programming of machines and complex systems using CAM software systems. Programming of metalworking machines by removing chips (milling machines and lathes). Application of CAE software systems in computer analysis of static behavior of specific parts modelled within the CAD module.

Introduction to Computer Integrated Manufacturing (CIM) and its subsystems. Automatic flexible technological systems, machines and other structures within CIM. Automated programming of machines and complex systems within CIM. Procedures and standards for connecting subsystems within CIM. Methodology for connecting CIM components into a single whole and software solutions. Characteristic models of CIM.

Practical teaching

Computer exercises – Modelling of prismatic parts. Modelling of cylindrical parts. Modelling of complex parts. Preparation of technical documentation. Computer-aided design. Programming of milling machines and lathes using specific programming systems. Computer analyses of static behavior.

Literature

Zeljković, M., Tabaković, S., Živković, A., Živanović, S., Mlađenović, C., Knežev, M.: Osnove CAD/CAE/CAM tehnologija, Univerzitet u Novom Sadu, Fakultet tehničkih nauka, 2018

Zeljković M., Tabaković, S., Antić, A., Živković, A: Programiranje numerički upravljanih obradnih sistema, Fakultet tehničkih nauka, Novi Sad, 2015

Blanuša, V., Čabrilo, A.: Mašine alatke (elektronski materijal), VTŠSS-Novi Sad, 2021

Active teaching classes: 105	**Theoretical teaching: 60**	**Practical teaching: 45**

Teaching methods:

Monologic, dialogic, interactive.

Assessment (maximum number of points 100)

Pre-exam obligations	**Points**	**Course exam**	**Points**
Active participation in classes		Written exam	50
Practical work		Oral exam	
Interim exam/s	50		
Seminar paper/s			

Back to top

Study programme: Information technologies

Course title: Professional master practice 2

Teacher: Head of the study programme

Course status: Compulsory

Number of ECTS: 4

Prerequisite course: None

Course objective

Gaining new and applying acquired advanced knowledge and skills related to the study programme while in the selected host company, focused on the preparation of the Preliminary master thesis and the Master thesis.

Learning outcomes

Practical experience in the application and use of knowledge and skills acquired during the master professional studies. The student is able to identify, analyse and solve a specific problem in the field of the study programme. Data/results that have been collected can serve as a basis for preparing the practical part of the Preliminary master thesis and the Master thesis.

Course contents

Professional master practice 2 is carried out in the 2nd year of master studies, lasting 90 hours. It is performed in companies dealing with production, service and other activities, according to general and individual programme contents, agreed between the co-mentor from the host company, the mentor in charge of practice from the School and the student, with the purpose of collecting data/results and acquiring skills that can serve as a basis for preparing the practical part of the Preliminary master thesis and the Master thesis. The student can also perform Professional master practice 2 at the Higher Education Technical School of Professional Studies in Novi Sad, according to a predetermined and approved plan.

Forms related to Professional master practice 2 are available to students registered for the course.

The student in Professional master practice 2 solves special tasks, which are defined by the co-mentor from the company and the mentor.

The mentor and co-mentor guide the student and precisely define the work tasks and obligations in order to enable the student to gain appropriate data/results/skills necessary for the preparation of the practical part of the Preliminary master thesis and the Master thesis. These are, for example: planning and conducting a survey and/or preliminary experiments, considering possible solutions to a specific problem, selecting a methodology and detailed planning for performing the main analyses, selecting technologies when developing a practical project. The selected assignments provide adequate opportunities for the student to demonstrate abilities in several categories: application of professional knowledge; practical skills; analytical skills, the ability to solve concrete problems; innovation and originality; development skills; time management; written and oral expression, all with the aim of training for the future profession. Then the student performs planned activities – data analysis and/or surveys, experiments, numerical simulations, statistical data processing, development of algorithms and programs, applies selected technologies during the development of the planned practical work. A co-mentor from the company regularly cooperates with the student, instructs them and monitors their work. During the practice, the student keeps the internship diary.

After completing the Professional master practice 2, the student submits a report that corresponds in content and form to the instructions defined at the beginning of the internship.

Literature

Technical documentation from the company and other relevant literature agreed between the student, the co-mentor from the company and the mentor.

Active teaching classes: 90

Teaching methods: Practical, laboratory, individual work.

Assessment (maximum number of points 100)

Pre-exam obligations	**Points**	**Course exam**	**Points**
Active participation in classes		Written exam	
Practical work	50	Oral exam	
Interim exam/s		Practical exam	50
Seminar paper/s			

Back to top

Study programme: Information technologies

Course title: Preliminary master thesis

Teacher: Head of the study programme/Mentor

Course status: Compulsory

Number of ECTS: 16

Prerequisite course: None

Course objective

Developing the ability to independently plan a project that will solve a specific practical problem from the selected field of the study programme. Developing the ability to identify and define problems and communicate with the mentor. Introduction to practical methodologies that can be used in the selected field. Integrating, upgrading, and practically applying the knowledge acquired throughout the studies. Preparation for writing one's master thesis.

Learning outcomes

Through independent and mentored work, the student has defined the outline topic of the master thesis and elaborated it, up to the written proposal and oral defence (viva) of the detailed plan for the master thesis. Thus, the student has developed the ability to conduct analyses and identify problems within the given topic, and is ready to write the master thesis.

Course contents

The course is taught through independent applied research and mentoring, during which the student defines the outline topic of the master thesis, then elaborates it in detail, up to the written proposal and oral defence of the detailed plan for the development of the master thesis.

After receiving the topic and assignment, along with regular consultation and reporting, the student studies the appropriate professional literature, previous master theses and other works in a similar field and performs preliminary analyses in order to better define the assigned problem.

In the second phase, the student studies the nature, structure and complexity of the problem in more detail. He actively searches for published knowledge from the broader topic of the assigned work, plans and conducts preliminary research or programme/idea solutions, in order to better quantify the task and topic. In this way, the student draws conclusions about possible solutions, selects a methodology/technology and plans in detail the organization and execution of the main analyses or applications of the selected technology when developing a practical project, with the aim of producing the master thesis. During this process, the student regularly consults with the mentor, and if necessary, with other teachers of the School. The goal of the student's activities during this phase of research is to gain the necessary experience to solve more complex problems and to identify opportunities for applying previously acquired knowledge in practice. At the end of the research work, the student writes the Preliminary master thesis.

The quality of the written Preliminary master thesis is assessed by the mentor/head of the study programme. The Preliminary master thesis is defended orally.

Literature

Textbooks in the selected field, papers from professional journals, technical documentation from the company, software tutorials, previous master theses, etc.

Active teaching classes: 300

Teaching methods

Practical, research, mentoring.

Assessment (maximum number of points 100)

Pre-exam obligations	**Points**	**Course exam**	**Points**
Active participation in classes		Written exam	
Practical work		Oral exam	
Interim exam/s		Preliminary master thesis (viva)	50
Seminar paper/s			
Preliminary master thesis (preparation)	50		

Back to top

Study programme: Information technologies

Course title: Master thesis

Teacher: Mentor

Course status: Compulsory

Number of ECTS: 16

Prerequisite course: Passed all courses from the study programme

Course objective

Application of theoretical and practical knowledge and skills acquired in the study programme with the aim of enabling students to adopt advanced methods for engineering decision-making and complex problem-solving. Documenting the completed work in written form and public defence (viva) of the project results.

Learning outcomes

Capability to comprehend and solve complex practical problems and apply advanced methods in engineering decision-making. Skills in both written and verbal presentation of work results.

Course contents

Upon passing all exams, the student proceeds to the preparation of the master thesis. The master thesis is a research, methodological, and practical work of the student, aimed at familiarizing them with solving complex practical problems and the methodology of developmental and practical research within one of the fields of the master study programme. Procedures and forms related to the master thesis are available on the School's website.

During the preparation of the master thesis, the student actively consults with their mentor and may also consult other teachers from the School as needed (subject area professor, mentor from Professional master practice 2, etc.). In addition to a detailed review of relevant contemporary literature and/or legal and technical regulations in the chosen field, the master thesis should include at least one of the following elements: analytical, computational, design, developmental, software programming, or experimental aspects.

The master thesis is conducted independently, and it is preferable for it to be linked with the specific knowledge and skills acquired during the professional master practice.

The student writes the master thesis based on the detailed execution plan previously presented in the Preliminary master thesis. In doing so, the student finalizes the preliminary experiments, analyses, or software solutions obtained during the research for the preliminary paper to achieve final results or programs. Then, the student prepares the master thesis in the prescribed format, which includes the following chapters: introduction, objective of the work, subject elaboration, conclusion, and references.

After completing the master thesis, the student submits the written version of the work. Following the verification of compliance with the conditions according to the procedure specified by the Rules on Topic Selection, Preparation, and Defence of Final Work for Bachelor, Specialist, and Master Studies of the Higher Education Technical School of Professional Studies in Novi Sad, the student proceeds to the oral presentation and defence (viva) of the master thesis. The defence is public and takes place before a committee.

Literature

Textbooks in the chosen field, papers from professional journals, technical documentation from the company, software tutorials, previous master theses, etc.

Number of classes: 60

Teaching methods

Mentoring, practical, laboratory, individual work.

Assessment (maximum number of points 100)

Pre-exam obligations	**Points**	**Course exam**	**Points**
Active participation in classes		Written exam	
Practical work		Oral exam	
Interim exam/s		Master thesis (viva)	**50**
Seminar paper/s			
Master work (preparation)	**50**		

Back to top