BOOK OF COURSES

UNDERGRADUATE STUDIES

Study Program:
Computer Science
**Study Programme:** Computer science

<table>
<thead>
<tr>
<th>Course title: INTERNET AND WEB TECHNOLOGIES</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Teacher:</strong> Cvetković M. Dragan, Stojmenović I. Miloš</td>
</tr>
<tr>
<td><strong>Course Status:</strong> Required</td>
</tr>
<tr>
<td><strong>ECTS credits:</strong> 8</td>
</tr>
<tr>
<td><strong>Requirements</strong> No requirements</td>
</tr>
</tbody>
</table>

**Course objective**
The primary objective of this course is to familiarize students with basic principles of functioning of the Internet, graphic and web design. The secondary objective of this course is to gain experience in the implementation and implementation of various technologies and programming languages for the needs of creating web sites for a real market environment. As the areas of web design and web programming are extremely topical and with a high volume of fluctuations in the technological and practical sense, one of the additional goals is to train students for independent research in this field as well as to monitor and adapt to current world trends.

**Course outcome**
The general competencies students will acquire are the analysis of existing technologies and programming languages, the synthesis of existing technologies, and the prediction of new trends in web programming. In keeping with the current issue of web development, students should acquire practical knowledge and skills for applying the acquired knowledge in practice.

The professional outcome of this course includes the following five components: 1. Understanding the basic concepts of the Internet - the concept of web browser, web server and web search engine, the concept of protocol, http request / response, client and server communication; 2. Learning about the language HTML - the concept and purpose of client / server languages, learning markap languages, training to create arbitrary web content based on HTML 5; 3. Getting to know the concepts of styling the content of a web page: CSS - understanding different ways of organizing web page elements and their arbitrary styling based on CSS 2 and the latest CSS 3 elements; 4. Analyzing SEO Parameters of Websites - a set of techniques that improve the quality and rating of the site and better serve them as web browsers; 5. Understanding client programming languages and JavaScript implementations - improving Java language elements and its specialized jQuery library with the creation of interactive static web sites.

**Course content**

**Theoretical teaching**
Understanding the way information is transmitted on the Internet, with particular emphasis on the needs of websites, is a key element for the optimal planning and design of quality websites. In the course of theoretical lessons students will gain insight into the principles and technologies of graphic and web design through the following teaching units: topics such as web sites, types of web users, Internet communications, domain names, hosting, HTML purposes (language elements: tags, attributes, entities), work with multimedia, web page content organization, web site styling, work with forms, advanced HTML 5 and CSS 3 applications, JavaScript basics (variables, apps, loops, work with DOM, functions, arrays, objects and methods), basic principles, techniques, and SEO implementations.

**Practical teaching**
Creating a web page, setting for displaying Serbian letters, working with text, images, links, lists, tables and forms. Image processing and image customization for web site needs. Working with multimedia files in a web page. Working with selectors, properties and positioning of elements. JavaScript variables, flow controls, loops, functions, and arrays. Managing DOM: Methods, Properties, and Objects of the JS. Working with events. jQuery effects, animation and transition.

**Literature**

**Number of active classes 5**

| Theoretical classes: 3 | Practical classes: 2 |

**Teaching methods:** Lectures, practices, consultations, homework, midterms, final exam.

**Final grade (maximum number of points: 100)**

<table>
<thead>
<tr>
<th>Pre-exam requirements</th>
<th>Points</th>
<th>Final exam</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attendance</td>
<td>10</td>
<td>Exam</td>
<td>30</td>
</tr>
<tr>
<td>Midterm 1</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Midterm 2</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total:</td>
<td>70</td>
<td>Total:</td>
<td>30</td>
</tr>
</tbody>
</table>
Study Programme: Computer science

Course title: MATHEMATICS

Teacher: Živković S. Dejan, Radičić M. Biljana

Course Status: Mandatory

ECTS credits: 8

Requirement: No requirements

Course Objective:
The main goals of the systematization of the existing mathematical knowledge and learning new mathematical concepts that are important for the subjects in the field of computer science.

Course Outcome:
The student's ability to successfully monitor the mathematical objects in subsequent semesters on the basis of acquired knowledge in the field of matrix algebra, testing and drawing of the graph of a function, application of differential and integral calculus.

Course Content:

Theoretical teaching
Matrices, determinants, characteristics, counting and implementation. Equation system solving.

Practical teaching
Practical teaching follows the theoretical contents by solving problems

References

Number of active teaching units: 6
Lectures: 3
Practice: 3

Teaching Methods
Lectures, practice. Compulsory usage of one mathematical software.

Final grade (maximum number of points: 100)

<table>
<thead>
<tr>
<th>Pre-examination requirements</th>
<th>Points</th>
<th>Final exam</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attendance</td>
<td>10</td>
<td>Exam</td>
<td>30</td>
</tr>
<tr>
<td>Midterm 1</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Midterm 2</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total:</td>
<td>70</td>
<td>Total:</td>
<td>30</td>
</tr>
</tbody>
</table>
Study Programme: Computer science

Course title: PROGRAMMING FUNDAMENTALS

Teacher: Živković Z. Miodrag, Bačanin-Džakula V. Nebojša

Course Status: Mandatory

ECTS credits: 8

Requirements: No requirements

Course objective
The main goal of this course is to introduce students to the basic elements of programming, and fundamental principles of programming in C++ programming language. The secondary course objective is to gain experience in implementation of basic programs in C++ programming language through practical work and to develop a solid basis for a further upgrade on later years of studies. Another goal of this course is that student gains the necessary skills for performing research activities in this dynamic area by exploitation of existing C++ environments and tools.

Course outcome
General competencies which students will gain are analysis, synthesis, and prediction of results and effects, research methods and processes, as well as knowledge application in practice.

The practical outcome of this subject consists of 1. Understanding of fundamental concepts of programming by using programming language C++, basic characteristics of C++ and object oriented languages; 2. Getting students ready for individual application of fundamental programming concepts in C++ programming language and usage of standard C++ libraries; 3. Basic understanding of object-oriented paradigm, classes and objects concepts, functions, inheritance; 4. Ability to develop simple programs in C++ programming language, ability to use NetBeans and Visual Studio development environments.

Course content
Theoretical classes
C++ programming has become one of the most important sectors of the IT industry, as C++ is one of the most popular programming languages. During theoretical classes, students will gain fundamental theoretical knowledge about computer programming, and practical skills for writing programs in C++ programming language through following topics: fundamental elements of programming, introduction to C++ programming language, basic elements of C++, expressions, statement blocks, decision statements, loop statements, functions, classes and objects, arrays, introduction to class inheritance, working with files.

Practical classes in computer laboratories
Practical classes follow theoretical classes, and they are held in computer classrooms on computers with installed C++ IDE (Visual Studio and NetBeans). During practical classes, students will get insight into environment setup, usage of basic C++ libraries, development of simple C++ programs and practical application of principles learned during theoretical classes.

Literature
3.

Number of active classes 2
Theoretical classes: 3
Practical classes: 5

Teaching methods: Lectures, practices, consultations, homework, midterms, final exam.

Final grade (maximum number of points: 100)

<table>
<thead>
<tr>
<th>Pre-exam requirements</th>
<th>Points</th>
<th>Final exam</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attendance</td>
<td>10</td>
<td>Exam</td>
<td>30</td>
</tr>
<tr>
<td>Midterm 1</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Midterm 2</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>70</td>
<td>Total:</td>
<td>30</td>
</tr>
</tbody>
</table>
Study Programme: Computer science

Course title: COMPUTER SYSTEMS AND APPLICATIONS

Teacher: Tomašević R. Violeta, Bačanin-Džakula V. Nebojša, Sarac K. Marko

Course Status: Mandatory

ECTS credits: 8

Requirements: No requirements

Course objective
Introducing students to the basic concepts of computer systems. The course represents an introductory hardware course that serves as a basis for all other courses. The application software is studied within the practical part of the course.

Course outcome
The general competences that students will acquire are the analysis, synthesis and forecasting of solutions and consequences, as well as mastering the methods, procedures and processes of research along with the application of knowledge in practice.

At the end of the course, it is expected that a successful student is able, through the practical demonstration, to show a detailed understanding of all aspects of the hardware of a computer system, as well as knowledge of advanced computer systems architecture. Also, at the end of the course it is expected that the student will adopt an advanced level of use of standard applications that are used in everyday work on computers.

Course content
Theoretical classes
Pre-history up to 1946, Computer hardware history, software, networking, pioneers of computing, definition of computers, digitization, positional number systems, bits, bytes and words, representation of numbers and positioning number systems, fixed and floating point numbers, negative numbers with sign and complete complement, Presentation of non-numerical data (character encoding, graphic data), Presentation of syllables and sequences, Basic organization of Von Neumann's computer, Control unit, instruction creation, decoding and execution, Set of instructions and types (Data Manipulation, Control, Input / Output), Programming in Machine and Assembly languages, The concept of Virtual Machine, Virtual Machine Hierarchy, Intermediary Languages, Interpreter and Compiler Comparison, History of Programming languages, Computer System components.

Practical classes in computer laboratories
Practicing and understanding the positioning number systems, binary numbers, complete complement, and exponential floating-point number representation in IEEE754 format. Introducing computer system components. Hand-on experience with application software - text processors: creation and maintenance of large and complex projects; complex typography, formatting and layout, including tables, forms and graphics; macros; mail integration, work tables: more complicated analysis; more detailed and more complex reports; advanced editing, functions and analysis tools; macros, database systems: more complicated reports with deeper analysis; advanced functions in queries, forms and reports; macros; import, export and linking of the data, presentations: more complicated and effective presentations; advanced scheduling and display features; adding more complex multimedia elements; using powerful link tools with other applications.

References

Number of active classes 4
Theoretical classes: 2
Practical classes: 2

Teaching methods: Lectures, practices, consultations, homework, midterms, final exam.

Final grade (maximum number of points: 100)

<table>
<thead>
<tr>
<th>Pre-exam requirements</th>
<th>Points</th>
<th>Final exam</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attendance</td>
<td>10</td>
<td>Exam</td>
<td>30</td>
</tr>
<tr>
<td>Midterm 1</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Midterm 2</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total:</td>
<td>70</td>
<td>Total:</td>
<td>30</td>
</tr>
</tbody>
</table>
Study Programme: Computer science

Course title: DISCRETE MATHEMATICS

Teacher: Tuba J. Milan, Radičić M. Biljana, Živković S. Dejan

ECTS credits: 8

Course Status: Mandatory

Course Objective:
This course is designed to introduce students to the principles of discrete mathematics and all the ways of choosing appropriate methods depending on the problems to be solved in the field of computer science.

Course Outcome:
Students will be able to apply mastered methods of discrete mathematics in professional subjects in subsequent semesters of teaching.

Course Content
Theoretical classes

References

Number of active teaching units: 5
Lectures: 3
Practice: 2

Teaching Methods
Lectures, computer practice, midterm exams, office hours, seminar papers, final exam.

Final grade (maximum number of points: 100)

<table>
<thead>
<tr>
<th>Pre-examination requirements</th>
<th>Points</th>
<th>Final exam</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attendance</td>
<td>10</td>
<td>Exam</td>
</tr>
<tr>
<td>Midterm 1</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>Midterm 2</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>70</td>
<td>Total: 30</td>
</tr>
</tbody>
</table>

Points
Study Programme: Computer science

Course title: COMPUTER ORGANIZATION AND ARCHITECTURE

Teacher: Tomašević R. Violeta, Bačanin-Džakula V. Nebojša

Course Status: Mandatory

ECTS credits: 8

Requirements: Computer Systems and Applications

Course objective
This course represents second hardware course, where the topics touched in the previous course Computer Systems and Applications are examined in more detail. In the first part of the course, in order to understand the organization and architecture of computer systems, students are introduced to the basics of digital logics, where essential teaching units are processed: digital logic and digital systems, history and overview, combinatorial logic circuits, modular design of combinator circuits, memory elements, modeling and simulation. In the second part of the course, the structure of the processor, memory system and the entire computer system is studied.

Course outcome
At the end of the course, it is expected that a successful student learns how to organize computer system by using basic components, and how to design certain computer system parts. Also, it is expected that a successful student at the end of the course will fully understand the inner workings of the processor, as well as the methods of programming at the machine and assembly level from the outside.

Course content

*Theoretical classes*
Overview and history of computer architecture, Basic building elements (logic circuits, flip-flops, counters, registers, PLAs), Logical expressions, minimization, Sum of product summaries, Notation for register transfers, Physical parameters (gate delay, input capacity, output capacity ), Reasons for studying digital logic, Areas: logical circuits, memory, registers, digital systems, combinatory logic, sequential circuits, basic logic circuits (AND, OR, NOT, NAND, NOR, XOR), Realization of functions using logic circuits, Design medium-sized combinations logic modules, multiplexers, demultiplexers, decoders, encoders, comparators, arithmetic functions (collectors, sequencers, prediction of transmission), Multiplexers, Demultiplexers, Hierarchical design of combinatorial circuits using logic modules, Level-sensitive or changeable devices, master-slave devices, Basic flip flops (SR, D, JK, T), Instructions formats, Address modes, Subroutine call and return mechanism, Input / output and interruptions, Data storage and their technologies, Encryption, data compression and data integrity, Memory hierarchy, Organization and operation of main memory, Latency, cycle time, bandwidth and interlace, Cache memory (address mapping, block size and making and replacement policies), Virtual Memory (page table, TLB), processing errors and reliability, Input / output basics: handshake, buffering, programmed I / O, interrupted I / O, interrupt system structures: vector and priority, interrupt confirmation, external memory, physical organization and drives, masters: protocols, arbitration, direct access to memory DMA), Introduction to networks, Multimedia support, RAID architecture, Implementation of a simple bus, Control unit: wired realization according to microprogrammed, Instructional pipelines, Introduction to Parallelism at the level of instruction (ILP), Introduction to SIMD, MIMD, VLIW, EPIC, Systolic architecture, Interconnection networks (hypercube, mixed replacement, mesh, crossbar), Shared memory systems, Cohesion cache, Memory models and memory consistency.

*Practical classes in computer laboratories*
Electronics, switches, simple processor model, practical experiments with standard logic circuits, simulation software. Designing individual components, working in machine language, work on assembler. Developing a simple processor and its language.

References

Number of active classes 6
Theoretical classes: 3
Practical classes: 3

Teaching methods: Lectures, practices, consultations, homework, midterms, final exam.

Final grade (maximum number of points: 100)

<table>
<thead>
<tr>
<th>Pre-exam requirements</th>
<th>Points</th>
<th>Final exam</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attendance</td>
<td>10</td>
<td>Exam</td>
<td>30</td>
</tr>
<tr>
<td>Midterm 1</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Midterm 2</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total:</td>
<td>70</td>
<td>Total:</td>
<td>30</td>
</tr>
</tbody>
</table>
**Study Programme:** Computer science

**Course title:** OBJECT-ORIENTED PROGRAMMING

**Teacher:** Bačanin-Džakula V. Nebojša, Živković Z. Miodrag

**Course Status:** Mandatory

**ECTS credits:** 8

**Requirements:** Programming Fundamentals

### Course objective

The main goal of this subject is to introduce students to the object-oriented paradigm in Java programming language. The secondary course objective is to gain experience in implementation of object-oriented programs in Java programming language through practical work and to develop a solid basis for a further upgrade on later years of studies. Another goal of this subject is that student gain necessary skills for performing research activities in this dynamic area by exploitation of existing Java environments and tools, such as Eclipse and IntelliJ.

### Course outcome

General competencies which students will gain are analysis, synthesis, and prediction of results and effects, research methods and processes, as well as knowledge application in practice.

The practical outcome of this subject consists of
1. Understanding of object-oriented programming in Java programming language;
2. Getting students ready for individual application of fundamental concepts of object-oriented programming in Java programming language, and usage of standard Java libraries;
3. Advanced knowledge of object-oriented paradigm, concepts of classes and objects, inheritance, encapsulation, polymorphism; 4. Understanding Java design patterns and their application in programming; 5. Ability to individually develop complex Java programs, and the ability to use Eclipse and IntelliJ development environments.

### Course content

#### Theoretical classes

Java programming has become one of the most important sectors of the IT industry, as Java is one of the most popular programming languages, and popular platforms Spring and Android are based on Java. During theoretical classes, students will gain fundamental theoretical knowledge about object-oriented programming, and practical skills for writing object-oriented programs in Java programming language through following topics: objects and classes, encapsulation, inheritance, abstraction, polymorphism, abstract classes, interfaces, exceptions, generics, lambda expressions, object serialization, working with binary files.

#### Practical classes in computer laboratories

Practical classes follow theoretical classes, and they are held in computer classrooms on computers with installed Java IDE (Eclipse, IntelliJ and NetBeans). During practical classes, students will get insight into environment setup, usage of basic Java libraries, development of advanced Java programs and practical application of principles learned during theoretical classes.

### Literature


### Number of active classes

| Theoretical classes: 3 | Practical classes: 3 |

### Teaching methods:

Lectures, practices, consultations, homework, midterms, final exam.

### Final grade (maximum number of points: 100)

<table>
<thead>
<tr>
<th>Pre-exam requirements</th>
<th>Points</th>
<th>Final exam</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attendance</td>
<td>10</td>
<td>Exam</td>
<td>30</td>
</tr>
<tr>
<td>Midterm 1</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Midterm 2</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total:</td>
<td>70</td>
<td>Total:</td>
<td>30</td>
</tr>
</tbody>
</table>
Study Programme: Computer Science
Course Title: ENGLISH LANGUAGE 1
Teacher: Petrović D. Jasna
Course Status: Compulsory
ECST credits: 6
Prerequisite: /

Course Goals and Objectives

The Course aims at improving students’ skills and competences as regards both receptive and productive language skills – while, at the same time, levelling their knowledge at B2 CEFR level. In this way, the Course intends to enable students to communicate in different language contexts and for different purposes.

Course Outcomes

Upon completing the Course, students are capable of understanding and transmitting the intended messages in English. In that communication process – students exhibit the knowledge of language belonging to B2 CEFR level – in terms of writing, speaking, listening and writing. Moreover, they are pragmatically aware of specific needs in different language contexts.

Course Content

Grammar-related Units

Present Simple and Continuous, Static verbs, Articles, Present Perfect Simple and Present Perfect Progressive, Countable and uncountable nouns; Quantifiers, Past Simple and Past Progressive, Used to, Would, Past Perfect Simple and Past Perfect Progressive, Would, was/were going to, Future forms, Time clauses, Modal Verbs, Passive Voice

Lexical Units

Word building; adjective suffixes, noun suffixes; Colour idioms, verbs starting with re-, Sports idioms, Phrasal verbs related to travel, Compound adjectives describing character traits, Idioms with parts of the body, Compound adjectives describing character traits, idioms with parts of the body, phrases and collocations related to education and learning; ESP: Technology in the Modern World, Hardware, Software, the Development of PC

Primary and Secondary Sources Selection

3. Murphy, R. English Grammar in Use (Book with answers and Interactive ebook), CUP, 2015

Active Teaching Hours: 3

<table>
<thead>
<tr>
<th>Theory</th>
<th>Practice</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

Applicable Teaching Methods:
Interactive lectures – gamified and collaborative teaching/learning, humanistic and personalized methods, language skills’ assessment, tests, midterm exams, written and oral exam

Final grade (maximum number of points: 100)

<table>
<thead>
<tr>
<th>Pre-exam Requirements</th>
<th>Points</th>
<th>Final Exam</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attendance and in-class Activity</td>
<td>10</td>
<td>Written part</td>
<td>30</td>
</tr>
<tr>
<td>Practice</td>
<td></td>
<td>Oral Part</td>
<td></td>
</tr>
<tr>
<td>Mid-term Exams/Tests</td>
<td>60</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Other Assessment Items</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>70</td>
<td>Total: 30</td>
<td></td>
</tr>
</tbody>
</table>
Study Programme: Computer Science

Course title: DATA STRUCTURES AND ALGORITHMS

Teacher/Teachers: Živković S. Dejan, Tuba J. Milan

Course Status: Mandatory

ECTS credits: 8

Prerequisites: Programming Fundamentals, Object-oriented Programming

Course Objective

The course Data Structures and Algorithms as the third programming course in this study program, relies on the knowledge acquired under the courses Programming Basics and Object-oriented Programming. After mastering the basic programming principles and concepts in the previous two courses, the aim of this course is to introduce complex data structures and algorithms, along with the complexity of algorithms.

Course Outcome

The general competences that the students will acquire are analysis, synthesis and forecasting of solutions and consequences, mastering the methods, procedures and processes of research, as well as the application of adopted knowledge in practice. The professional outcome of this course relates primarily to enabling the students for stand-alone utilization of advanced programming techniques, taking into account the notion of algorithm’s complexity, and training the students to independently implement advanced data structures, taking into account previous experience with all standard data structures.

Course content

Theoretical realization


Practical realization

Stand-alone implementation of complex and abstract data structures (stacks, singly and doubly-linked lists, heaps, queues, hash tables, etc.) and algorithms in Java and C++ programming languages. Stand-alone implementation of algorithms (bubble sort, insertion sort, linear and binary search algorithms, etc.) in the Java programming language. The utilization of prepacked and preprogrammed algorithms and data structures from the Java Development Kit (JDK).

References


Number of active classes 5  Theoretical classes 3  Practical classes 2

Teaching methods: Lectures, computer exercises, consultation hours, practical project-oriented

Final grade (maximum number of points: 100)

<table>
<thead>
<tr>
<th>Pre-exam obligations</th>
<th>Theoretical classes</th>
<th>Practical classes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attendance</td>
<td>10</td>
<td>Oral exam 30</td>
</tr>
<tr>
<td>Mid-term Exam 1</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>Mid-term Exam 2 (Project)</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>Total:</td>
<td>70</td>
<td>Total: 30</td>
</tr>
</tbody>
</table>
Study Programme: Computer science
Course title: PROBABILITY AND STATISTICS
Teacher: Radičić M. Biljana, Živković S. Dejan
Course Status: Mandatory
ECTS credits: 6
Requirement: -
Course Objective:
The main goal is to use the knowledge gained from probability and statistics, as applied mathematics, to solve the problem of the profession and real life. Those students develop a way of thinking related to mass phenomena using the probability and statistics model.

Course Outcome:
Students will be able to apply a probability decision-making model, model real problems using appropriate random variables, set hypotheses and make decisions by applying the appropriate test.

Course Content
Theoretical classes

Practical teaching
Practical teaching follows the theoretical contents by solving problems.

References

Number of active teaching units: 4
Lectures: 2
Practice: 2
Teaching Methods
Lectures, computer practice, midterm exams, office hours, seminar papers, final exam.

Final grade (maximum number of points: 100)

<table>
<thead>
<tr>
<th>Pre-examination requirements</th>
<th>Points</th>
<th>Final exam</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attendance</td>
<td>10</td>
<td>Written exam / Oral exam</td>
<td>30</td>
</tr>
<tr>
<td>First midterm</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Second midterm</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Seminar paper (s)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total:</td>
<td>70</td>
<td>Total:</td>
<td>30</td>
</tr>
</tbody>
</table>
Study Programme: Computer Science
Course Title: OPERATING SYSTEMS
Teacher/teachers: Jevremović D. Aleksandar, Šarac K. Marko
Course Status: Mandatory
ECTS credits: 8
Requirement: No requirement

Course Objective
The main objective of this course is to familiarize students with the principles of modern operating systems - processes and memory management, device management, file systems, virtualization, then operating systems for embedded devices, real-time operating systems, and error-resistance. Additionally, the goal is to link the adopted principles to real UNIX-like operating systems, and to enable students to use them for practical use in current scenarios - personal computers, smart phones, private data centers and cloud computing.

Course Outcome
The general competencies students will acquire on this subject include an analytical and systematic approach, as well as the application of appropriate methodologies and technologies in solving complex problems. The foreseen professional outcome of this course is understanding the role of the operating system in the abstracting of computer resources and accessing them, then understanding the competitive utilization of computer resources, process scheduling, memory management, virtualization concept, file system, fault tolerance and system performance evaluations. In addition, students are expected to independently predict the optimal performance of the computer system, install the operating system and boot loader, configure system and network parameters, optimize the operating system kernel, and use the BASH language to interact with the core.

Course content
Lectures

Labs
Preparation of a physical or virtual computer system for operating system installation, installation and initial operating system configuration, installation and configuration of the boot loader, compiling and installation of the kernel, advanced application of BASH language for interactive and script mode, network subsystem setup.

Literature

Number of active teaching 5
Lectures: 3
Practice: 2
Teaching methods: Lectures, computer exercises, consultations, practical work in the form of project activities

Final grade (maximum number of points: 100)

<table>
<thead>
<tr>
<th>Pre-examination requirements</th>
<th>Points</th>
<th>Final Exam</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activities during lecture</td>
<td>10</td>
<td>Project Task</td>
<td>30</td>
</tr>
<tr>
<td>Assessment 1</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Assessment 2</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total:</td>
<td>70</td>
<td>Total:</td>
<td>30</td>
</tr>
</tbody>
</table>
**Study Programme:** Computer science

**Course title:** DATABASES

**Teacher/Teachers:** Stojmenović I. Miloš, Veinović Đ. Mladen

**Course Status:** Required

**ECTS credits:** 8

**Previous condition:** No conditions

**Course objective**
The primary objective of this course is to acquire theoretical and practical knowledge for working with databases, mechanisms for storing structured data and ways of their processing in order to obtain useful information. After adopting basic concepts, next goal is to understand the modeling techniques and to overcome the SQL language in order to work with relational databases. Special significance for students is being able to construct and execute multiple queries, according to the given logical conditions. In the end, the goal is to master the practical knowledge of administration and database maintenance.

**Course outcome**
Students will acquire knowledge necessary for modeling relational databases, their implementation in the selected software for database management system. They will be qualified to apply SQL language over relational databases (syntax and usage rules), especially to implement complex queries over multiple tables, in order to obtain the desired information. Students will understand transaction execution procedures, as well as the ability to apply databases in modern multilayer Internet applications.

**Course content**

*Theoretical teaching*
Through the introduction of basic concepts of databases, the weaknesses of classical file systems are examined and the concepts of relational databases are mastered. Through conceptual modeling (entity-relationship model), the modeling of problems from the real world is studied, recognizing entities, characteristic relationships, cardinalities, special relationships. Relational algebra as a basis for queries over relational databases is also being studied. The basics of ANSI SQL languages, definition commands, queries over one or more tables, queries with subqueries, results filtering, etc. are being studied. Detecting poor structure relations, their normalization and de-normalization techniques. Transaction mechanisms. Work with databases in the presentation, application, and data layers. Administration and database recovery in case of failure. Introduction to more complex systems with databases: XML and databases, Datawarehousing, basics of the NoSQL databases.

*Practical teaching*
It is held in computer laboratories. Installation MySQL server through the XAMPP software package. Workbench Installation. Using the Navicat software package. Creating and working with relationships, working with keys (primary, foreign, candidate), familiarizing with indexes (types, applications, purpose), using SQL language (queries, query results, filters, clauses, group by, ordering, aggregation, ...), working with transactions, functions, procedures, triggers (purpose, creating and interrupting), using relational databases in applications (examples).

**References**

**Number of active teaching 6**

<table>
<thead>
<tr>
<th>Lectures</th>
<th>Practice</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

**Teaching methods:** Lectures, computer exercises, consultations, practical work in the form of project activities

**Final grade (maximum number of points: 100)**

<table>
<thead>
<tr>
<th>Pre-examination requirements</th>
<th>Points</th>
<th>Final exam</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activities during lecture</td>
<td>10</td>
<td>Practical project</td>
<td>30</td>
</tr>
<tr>
<td>Assessment 1 (Computer exercises)</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Assessment 2 (Computer exercises)</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Total:</strong></td>
<td><strong>70</strong></td>
<td></td>
<td><strong>30</strong></td>
</tr>
</tbody>
</table>
Study Programme: Computer science

Course title: CRYPTOLOGY 1

Teacher/Teachers: Veinović Đ. Mladen, Adamović Ž. Saša

Course Status: Required

ECTS credits: 8

Condition: No conditions

Course objective
The objective of this course is to enable students to understand the necessary cryptology knowledge and mathematical apparatus necessary for modern ciphering systems and its analysis and synthesis. Introduction into basic information security services: secrecy, integrity, authentication, and non-repudiation. Understanding the concepts and basic properties of symmetric and asymmetric ciphering systems. Understanding the secrecy of a system that is based on the secrecy of keys and on complex mathematical problems.

Course outcome
Students will be able to estimate quality of given ciphering systems and to understand their place and role in the modern integrated computer environment based on mathematical and information theoretical knowledge. Theoretical and practical knowledge for the implementation of security services will be acquired. They will be able to realize absolutely secret and computational safety secret ciphering systems, to understand digital signing techniques and the proper use of digital certificates. The acquired knowledge will enable students to master complex systems for authentication in distributed computer systems based on Internet technologies.

Course content

Theoretical teaching
In the first few classes fundamental cryptological concepts are being presented, as the science of secret communication. Based on Shenon's information theory, an ideal communication channel is being defined and concepts such as: encryption, decryption, algorithm and secret key are being introducing. The most famous historical cryptographic classical systems are considered. Generators of random and pseudo-random sequences are also introduced. Absolute secret ciphering systems are being considered. After that, symmetric encryption systems (sequential and block algorithms) are being studied as well as the most famous algorithms (DES, 3DES, AES). Then, asymmetric ciphering systems, the Diffie-Helman protocol, and the RSA algorithm are considered. The most famous hash algorithms are studied. Then digital signing techniques, a system of certificates and trust chains. Finally, the basics of access control are given: authorization and authentication in complex authentication protocols.

Practical teaching
It is held in computer laboratories. Cryptool software is being used, which includes given prepared modules for implementing security systems. The software enables the construction of complex security systems as the block scheme, without the need for using a specific programming language. In Cryptool software various encryption schemes are being demonstrated from classical to modern symmetric and asymmetric ciphering systems. The correct use of algorithms, key operation, application of hash functions is shown, and the effects of proper encryption and decryption are monitored via plaintext and ciphertext on the created communication channel.

References

Number of active teaching 6

<table>
<thead>
<tr>
<th>Lectures</th>
<th>Practice</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

Teaching methods: Lectures, computer exercises, consultations, practical work in the form of project activities

Final grade (maximum number of points: 100)

<table>
<thead>
<tr>
<th>Pre-examination requirements</th>
<th>Points</th>
<th>Final exam</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activities during lecture</td>
<td>10</td>
<td>Computer exercises</td>
<td>30</td>
</tr>
<tr>
<td>Assessment 1 (practical project)</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Assessment 2 (practical project)</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total:</td>
<td>70</td>
<td>Total:</td>
<td>30</td>
</tr>
</tbody>
</table>
**Study Programme:** Computer science

**Course title:** PROGRAMMING LANGUAGES

**Teacher:** Živković Ž. Miodrag, Jovanović A. Mlađan

**Course Status:** Mandatory

**ECTS credits:** 8

**Requirements:** Programming Fundamentals, Object-oriented Programming

**Course objective**
The main goal of this subject is to introduce students to the basic techniques of service-oriented programming, and to other advanced programming techniques in Java programming language and Spring framework. The secondary course objective is to gain experience in implementation of advanced programs in Java programming language and Spring framework through practical work and to develop a solid basis for a further upgrade on later years of studies. Another goal of this subject is that student gain necessary skills for performing research activities in this dynamic area by exploitation of existing Java environments and tools, such as Eclipse and IntelliJ, and Spring framework.

**Course outcome**
General competencies which students will gain are analysis, synthesis, and prediction of results and effects, research methods and processes, as well as knowledge application in practice. The practical outcome of this subject consists of 1. Understanding of basic concepts of network programming and multi-threaded programming in Java programming language; 2. Understanding of basic concepts of service-oriented programming in Java with Spring framework; 3. Getting students ready for individual application of fundamental concepts of service-oriented programming in Java programming language, and usage of Spring framework; 4. Basic understanding and implementation of RESTful web services; 5. Ability to individually develop Java web services, and ability to use Eclipse, IntelliJ, and Spring Boot development environments.

**Course content**

*Theoretical classes*
Java programming has become one of the most important sectors of the IT industry, as Java is one of the most popular programming languages, and popular platform Spring is based on Java. During theoretical classes, students will gain fundamental theoretical knowledge about network, multi-threaded and service oriented programming, and practical skills for writing service-oriented programs in Java Spring environment through following topics: fundamentals of network communication, sockets, multi-threaded programming, introduction to Spring, beans, wiring, autowiring, XML based and annotation based configuration, dependency injection, aspect-oriented programming, fundamentals of service-oriented programming, development of web services, working with database from Java Spring application, RESTful web services in Spring environment.

*Practical classes in computer laboratories*
Practical classes follow theoretical classes, and they are held in computer classrooms on computers with installed Java IDE (Eclipse, IntelliJ and NetBeans). During practical classes students will get insight into environment setup, Spring framework setup, development of Java web services and RESTful applications, and practical application of principles learned during theoretical classes.

**Literature**

**Number of active classes** 6  
**Theoretical classes:** 3  
**Practical classes:** 3

**Teaching methods:** Lectures, exercises, consultations, homework, projects, midterms, final exam.

**Final grade (maximum number of points: 100)**

<table>
<thead>
<tr>
<th>Pre-exam requirements</th>
<th>Points</th>
<th>Final exam</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attendance</td>
<td>10</td>
<td>Exam</td>
<td>30</td>
</tr>
<tr>
<td>Midterm 1</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Midterm 2</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total:</td>
<td>70</td>
<td>Total:</td>
<td>30</td>
</tr>
</tbody>
</table>
**Study Programme:** Computer Science

**Course Title:** COMPUTER NETWORKS

**Teacher/teachers:** Bačanin-Džakula V. Nebojša, Jevremović D. Aleksandar, Šarac K. Marko

**Course Status:** Mandatory

**ECTS credits:** 8

**Requirement:** No requirements

### Course Objectives

The main objective of this course is to familiarize students with the principles of modern computer telecommunications, network applications, reliable data transfer, routing and packet forwarding concepts, allocation of network resources and mobile telecommunications concepts. Additionally, the goal is to link the adopted principles with practical work in simulation environments and real network equipment.

### Course Outcome

The general competencies students will acquire on this subject include an analytical and systematic approach, as well as the application of appropriate methodologies and technologies in solving complex problems.

The foreseen professional outcome of this course is the understanding of modern computer telecommunications, the technologies on which they are based, the roles they have in modern distributed computing systems and environments, and the current trends. It is expected to adopt appropriate terminology, understanding the organization of the Internet network, commutation techniques, the principles of layers and the role of different layers. Then, the understanding of symbolic and logical addresses, the addressing schemes, the locating of network resources and client-server applications based on the use of sockets, the reliable data transfer protocol, the factors affecting performance and reliability, the function of internetworking and IP addressing, Ethernet technology, principles of wireless telecommunications networks and mobility.

### Course content

**Lectures**


**Labs**


### Literature


### Number of active classes

| Theoretical classes: 3 | Practical classes: 2 |

### Teaching methods:

Lectures, exercises, consultations, homework, projects, midterms, final exam.

### Final grade (maximum number of points: 100)

<table>
<thead>
<tr>
<th>Pre-exam requirements</th>
<th>Points</th>
<th>Final exam</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attendance</td>
<td>10</td>
<td>Practical task</td>
<td>30</td>
</tr>
<tr>
<td>Midterm exam 1</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Midterm exam 2</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total:</td>
<td>70</td>
<td>Total:</td>
<td>30</td>
</tr>
</tbody>
</table>
**Study Programme:** Computer Science

**Course Title:** ENGLISH LANGUAGE 2

**Teacher:** Petrović D. Jasna

**Course Status:** Compulsory

**ECST credits:** 6

**Prerequisite:** /

### Course Goals and Objectives

English Language 2 is a Course that follows the English Language 1 course – i.e. its form and content. The Course aims at enhancing students’ skills and competences regarding all the language skills – while, at the same time, levelling their knowledge at B2 CEFR level. In this way, the Course intends to enable students to communicate in different language contexts and for different purposes, as well as to make them more confident in the process.

### Course Outcomes

Upon completing the English Language 2 Course, students are capable of understanding and transmitting the intended messages in English in a confident manner. In that communication process – students exhibit the knowledge of language belonging to B2 CEFR level – in terms of writing, speaking, listening and writing. Moreover, they are pragmatically aware of specific needs in different language contexts.

### Course Content

**Grammar-related Units**

- Relative Clauses, Participle Clauses, Conditional Sentences Types 2,3, Unreal Past, Infinitives and ing-forms, Causative form, Reported Speech, Clauses of purpose, result and consession, Comparisons, Inversion,
- All/both/neither/none/either, Double conjunctions

**Lexical Units**

- Word Formation – foreign prefixes, suffixes, Diminutives, Gender Counterparts, Phrasal Verbs – valency, Verb and Adjective collocations; Food issues, Critical Thinking, Around the Globe – Interculturality, Gender Equality; ESP: Telecommunications, Social Networks, the Internet, Safety on the Internet

### Primary and Secondary Sources Selection

3. Murphy, R. English Grammar in Use (Book with answers and Interactive ebook), CUP, 2015

### Active Teaching Hours: 3

<table>
<thead>
<tr>
<th>Theory: 2</th>
<th>Practice: 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>30</td>
</tr>
</tbody>
</table>

### Applicable Teaching Methods:

Interactive lectures – gamified and collaborative teaching/learning, humanistic and personalized methods, language skills’ assessment, tests, midterm exams, written and oral exam

### Final grade (maximum number of points: 100)

<table>
<thead>
<tr>
<th>Pre-exam Requirements</th>
<th>Points</th>
<th>Final Exam</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attendance and in-class Activity</td>
<td>10</td>
<td>Written part</td>
<td>30</td>
</tr>
<tr>
<td>Practice</td>
<td></td>
<td>Oral Part</td>
<td></td>
</tr>
<tr>
<td>Mid-term Exams/Tests</td>
<td>60</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Other Assessment Items</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total:</td>
<td>70</td>
<td>Total:</td>
<td>30</td>
</tr>
</tbody>
</table>
Study Programme: Computer Science

Course title: SOFTWARE-DEFINED NETWORKS

Teacher / teachers: Sarac K. Marko

Course Status: Mandatory

ECTS credits: 8

Prerequisites: Computer Networks

Course objective
The objective of this course is that students master the necessary knowledge of the application layer used in modern computer networks. Getting acquainted with the basic and advanced services of computer network application services: the role of modern servers, application servers, redundancy, recovery after disaster. Mastering the concepts and basic properties of the correct choice of solution.

Course outcome
The general competences that students will acquire are the analysis, synthesis and forecasting of solutions and consequences, mastering methods, procedures and processes of research as well as the application of knowledge in practice.

Students become qualified to use the appropriate solution of the modern network operating system to independently evaluate and implement the correct implementation in a modern network computing environment. Students will gain theoretical and practical knowledge to implement application roles and network service functions. They will be able to implement services of business computer network of medium and large enterprises. It will meet the implementation of redundant Internet links, the basics of distributed computing, load management, backup management, system failure and replication to systems based on Internet technologies.

Course content
At the introductory classes, students are introduced to the concept of software-defined networks, centralized management, abstraction and automation of the modern data center. Software-defined networks developed from the need for automation, scalability and optimization of network resources. Students master terminology and become familiar with the elements of modern advanced data center solutions that will be implemented in a virtual environment of private distributed computing. After the basics, students master the techniques of automatic network configurations, increased reliability, centralized topology management, group policy management, network elements and infrastructure.

Practical realization
It is realized in computer laboratories. Students start exercises through individual work that introduces students to the network environment and roles of network operating systems. Students learn about the roles of Linux and Microsoft network operating systems and their functionalities in modern systems. In later stages of realization, students associate environments that were individually prepared in order to compare functionalities and acquire practical knowledge. The experiments and solutions that are implemented are based on case studies from the real network environment of business systems. Exercises are followed by lectures with different vendor solutions and demonstrations.

Literature

Number of active classes 6

| Methods of teaching: Lectures, exercises in a computer laboratory, consultations |
|---|---|---|

Final grade (maximum number of points: 100)

<table>
<thead>
<tr>
<th>Pre-exam obligations</th>
<th>Points</th>
<th>Final exam</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Presence</td>
<td>10</td>
<td>Written exam (electronic)</td>
<td>30</td>
</tr>
<tr>
<td>Colloquium 1 (theory + task)</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Colloquium 2 (theory + task)</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total:</td>
<td>70</td>
<td>Total:</td>
<td>30</td>
</tr>
</tbody>
</table>
Study Programme: Computer Science
Course Title: CRYPTOLOGY 2
Teacher: Milosavljević M. Milan, Adamović Z. Saša
Course Status: mandatory
ECST credits: 8
Prerequisite: Cryptology 1

Course Goals and Objectives
Getting acquainted with information theoretical concepts of security, methods of measurement and evaluation. Understanding the concepts of modern authentication protocols, identity management, secret sharing and zero-knowledge protocols. Getting acquainted with mechanisms for generating and distributing cryptographical keys based on biometric data. Mastering physical layer security techniques based on computer communication channels, using the principles of common randomness and their coding. Introduction to the basic elements of modern cryptanalysis. Getting acquainted with basic settings and operational techniques of modern cryptanalysis.

Course Outcomes
Obtaining theoretical and practical knowledge in the domain of digital signature, public key infrastructure, secret sharing, authentication, authorization, security of operating system, secure development of application software and cryptanalysis.

Course Content
Theory

Practice
Implementation and understanding of basic cryptographic protocols. Design of cryptobiometric systems based on iris biometrics. Implementation and understanding systems for generating and distributing cryptographic keys based on common randomness. Implementation of Mauer’s satellite protocol. Implementation and understanding methods for the synthesis of wire-tap codes based on LDPC and polar codes. Implementation of basic cryptanalytic methods for cyphertext and pseudo-random generator analysis.

Primary and Secondary Sources Selection

Active Teaching Hours: 5
Theory: 3
Practice: 2

Applicable Teaching Methods:
Lectures, practice, seminar papers, midterms, final exam

Final grade (maximum number of points: 100)

<table>
<thead>
<tr>
<th>Pre-exam Requirements</th>
<th>Points</th>
<th>Final Exam</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attendance</td>
<td>10</td>
<td>Written exam</td>
<td>30</td>
</tr>
<tr>
<td>First midterm</td>
<td>30</td>
<td>Oral exam</td>
<td></td>
</tr>
<tr>
<td>Second midterm</td>
<td>30</td>
<td>Project presentation</td>
<td></td>
</tr>
<tr>
<td>Other Assessment Items</td>
<td>70</td>
<td>Total:</td>
<td>30</td>
</tr>
</tbody>
</table>
**Study Programme:** Computer Science

**Course:** USER INTERFACE PROGRAMMING AND DESIGN

**Lecturer:** Jevremović D. Aleksandar, Jovanović A. Mlađan

**Course Status:** Mandatory

**ECTS credits:** 6

**Pre-requisite:** Programming Fundamentals

**Course objectives**

Learning principles and techniques for the development of usable user interfaces (visual design principles such as consistency, visibility, simplicity and efficiency).

Learning techniques for creating usable user interfaces, including prototypes of various levels of fidelity, implementation, input and output elements, model-view-controller architecture, and layout of user interface elements.

Learning methods and techniques for measuring user interface usability (heuristic evaluation).

**Course outcomes**

Based on the objectives, expected outcomes concerning **common competencies** include:

1) visual design principles for efficient organization and presentation of information in user interfaces;
2) methods and techniques for creating usable user interfaces;
3) methods and techniques for evaluating usability of user interfaces.

**Specific competencies** include fundamental knowledge of using technologies for user interface prototyping, technologies for implementing user interfaces, and tools for measuring user interface usability.

**Course content**

*Theoretical teaching*

Theoretical teaching starts by introducing user interface software architecture. It continues with the topics regarding principles of user-centred design and visual design. Then the teaching proceeds with the input and output elements of user interface giving some specific, common examples such as symbols, forms, menus, buttons and navigation. Next, it describes the principles of user interface design. The principles include the layout of user interface elements, colors, typography, localization and prototyping. What follows are the topics about evaluating user interface usability including usability attributes, heuristic evaluation, and testing with end users. Theoretical teaching concludes with user interface design patterns.

*Practical teaching*

Practical teaching applies the topics above by using Web technologies for user interface design and implementation (HTML, CSS and Angular).

**Literature**


**Total number of teaching hours:** 4  **Theoretical hours:** 2  **Practical hours:** 2

**Lecturing methods:** Frontal lectures, midterm exams, labs

**Final grade (maximum number of points: 100)**

<table>
<thead>
<tr>
<th>Pre-exam activities</th>
<th>Points</th>
<th>Final exam</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class presence and activity</td>
<td>10</td>
<td>Project defence</td>
<td>30</td>
</tr>
<tr>
<td>Midterm 1</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Midterm 2</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total:</td>
<td>70</td>
<td>Total:</td>
<td>30</td>
</tr>
</tbody>
</table>
Study Programme: Computer science

Course name: DIGITAL FORENSICS

Teacher: Jevremović D. Aleksandar, Adamović Ž. Saša

Course Status: Elective

ECTS credits: 8

Requirement:

Course objectives

The objective of the course is to introduce students to the fundamental principles of digital forensics as a hybrid science in a modern world whose importance is increasing today. In addition to theoretical background, students will be trained in various practical activities defined by the selected scenario and computer incident in modern digital forensic tools. Due to the rapid evolution of the software and hardware, there are big changes and bigger challenges on the side of forensic tools. For this reason, students should also be familiar with the methods and algorithms used by these tools and be trained to develop new ones.

Course outcome

The general competences that students will acquire are the analysis, synthesis and forecasting of solutions and consequences, mastering the methods, procedures and processes of research as well as the application of knowledge in practice.

The professional outcome of this course includes theoretical knowledge and practical experience in various forensic scenarios in the modern world. Students will acquire general competencies for various types of digital evidence, ways of detecting illegal or unauthorized activities, and recovering files on deliberately damaged media using forensic tools and techniques. This course provides the acquisition of professional competencies in the form of the ability to work in standardized software that is accepted and approved for use in detecting a computer incident and obtaining evidence relevant to the court.

Course content

Theoretical teaching

Introduction to digital forensics, computer incidents and ways of conducting a forensic investigation. Identify different sources for information collection (operating systems, files, network traffic, applications, combined sources). Recovering damaged or deleted data. Anti-forensics, the use of cryptography and steganography. Standardized forensic tools, computer incidents, legal issues and ethics.

Practical classes in labs

Includes the use of open-source Autopsy (https://www.autopsy.com/) an advanced forensic tool that allows most scenarios related to forensics of combined sources, starting with automatic or manual analysis of a large list of forensic modules (http://sleuthkit.org/autopsy/docs/user-docs/4.8.0/+).

Literature


Number of active classes 5  
Theory teaching: 3  
Practical teaching: 2

Methods of teaching: Lectures, computer exercises, consultations, practical work in the form of project activities.

Final grade (maximum number of points: 100)

<table>
<thead>
<tr>
<th>Pre-exam obligations</th>
<th>Points</th>
<th>Final exam</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attendance</td>
<td>10</td>
<td>Project presentation</td>
<td>30</td>
</tr>
<tr>
<td>First midterm</td>
<td>30</td>
<td>Oral exam</td>
<td></td>
</tr>
<tr>
<td>Second midterm</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total:</td>
<td>70</td>
<td>Total:</td>
<td>30</td>
</tr>
</tbody>
</table>
**Study Programme:** Computer Science

**Course title:** FUNDAMENTALS OF CRYPTANALYSIS

**Teacher / teachers:** Milosavljević M. Milan, Tomašević R. Violeta

**Course Status:** Mandatory

**ECST credits:** 8

**Prerequisites:** Programming languages, Mathematics, Statistics

**Course objectives**
The aim of this course is to introduce the basics of cryptanalysis, its historical development, methods and practice. By gaining knowledge from this domain, a more fruitful understanding of all security mechanisms of modern information protection in cyber space is achieved. Getting to know the mechanisms of leaving digital footprints, as a prerequisite for any cryptanalysis, is the basis for other modern areas of computer science, such as data engineering, machine learning, data science, etc. Students will be encouraged on this subject to generate generic cryptoanalytic attacks based on heterogeneous data sources available on the Internet, integrating knowledge and practices from the domain of statistical decision making, machine learning, algebraic analysis, and efficient methods of discrete optimization.

**The outcome of the course**
The general competences that students will acquire are the analysis and synthesis of generic cryptoanalytic attacks, structured in their complexity based on the a priori information available about the target system of interest. In addition, students will master the concrete methods, procedures and implementation of generic cryptoanalytic attacks. The professional outcome of this course includes the following components: understanding the basic concepts of collecting digital footprints that leave modern cryptological mechanisms of information protection, descriptive summarization, cleaning, dimensional reduction and discretization, both for the compression or for creation more convenient symbolic descriptions; training for the independent application of basic cryptanalysis algorithms; cyphertext based attacks, selected open text attack, or set of pairs of open text – cyphertext; understanding the basic settings of the problem of side channels and their treatment with machine learning methods, as a unique process for automating the acquisition of relevant information from the side channel footprints; getting familiar with the typical programming environment for these needs in the Jupiter Notebook, Python Library NumPy, SciPy, Pandas, Matplotlib, Scikit-learn, PyTorch, TensorFlow, Keras.

**Course Content**

*Theory*

*Practice*
It is performed in a computer laboratory in the Anaconda environment, Jupiter Notebook and Python programming language with the corresponding library from the cryptanalysis domain, the method of solving the system of linear and nonlinear algebraic equations in the Galois field, data science and machine learning. Particular attention will be given to the processing and analysis of information sources subject to the protection mechanism in order to identify, distinguish and reconstruct (partial or complete) open text information from the cypher text. The emphasis will be on practicing the analysis and synthesis of genetic cryptoanalytic attacks.

**Primary and Secondary Sources Selection**

**Active Teaching Hours:** 5  
**Theory:** 3  
**Practice:** 2

**Methods of teaching:** Lectures, practice, seminar papers, midterms, final exam

**Final grade (maximum number of points: 100)**

<table>
<thead>
<tr>
<th>Pre-exam Requirements</th>
<th>Points</th>
<th>Final Exam</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attendance and in-class Activity</td>
<td>10</td>
<td>Written part</td>
<td>30</td>
</tr>
<tr>
<td>First Mid-term Exams/Tests</td>
<td>30</td>
<td>Oral Part</td>
<td></td>
</tr>
<tr>
<td>Second Mid-term Exams/Tests</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>70</td>
<td></td>
<td>30</td>
</tr>
</tbody>
</table>
**Study Programme:** Computer Science  
**Course Title:** ENGLISH LANGUAGE 3  
**Teacher:** Petrović D. Jasna  
**Course Status:** Compulsory  
**ECST credits:** 6  
**Prerequisite:** /  

### Course Goals and Objectives

English Language 3 is a Course that follows the English Language 2 course – i.e. its form and content. The Course aims at enhancing students’ skills and competences regarding all the language skills – and placing them within the borders of C1 CEFR level. In this way, the Course intends to enable students to communicate in different language contexts and for different purposes – general, specific and academic.

### Course Outcomes

Upon completing the English Language 3 Course, students are capable of understanding and transmitting the intended messages in English in a confident and effective manner. In that communication process – students exhibit the knowledge of language belonging to C1 CEFR level – i.e. they are capable of performing rather demanding language tasks related to study and work endeavours. Moreover, students are pragmatically aware of specific needs in different language contexts.

### Course Content

**Grammar-related Units**

- Revising and shedding some new light on – Present Simple & Progressive, Present Perfect Simple & Progressive, Future forms, Past tenses, Passive voice, Relative clauses, Participle clauses, Adjectives/Adverbs, Gradability, Comparisons, Articles, Determiners

**Lexical Units**

- Common Lexical Mistakes, Easily Confused Words, Compound Formations, Fixed Expressions, E-commerce, IT Literacy, Socialisation, Generation and their Traits; ESP – Artificial Intelligence, Robots, Stylometry

### Primary and Secondary Sources Selection

2. Murphy, R. English Grammar in Use (Book with answers and Interactive ebook), CUP, 2015  

### Active Teaching Hours:

<table>
<thead>
<tr>
<th>Theory: 2</th>
<th>Practice: 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

### Applicable Teaching Methods:

Interactive lectures – gamified and collaborative teaching/learning, humanistic and personalized methods, language skills’ assessment, tests, midterm exams, written and oral exam

### Final grade (maximum number of points: 100)

<table>
<thead>
<tr>
<th>Pre-exam Requirements</th>
<th>Points</th>
<th>Final Exam</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attendance and in-class Activity</td>
<td>10</td>
<td>Written part</td>
<td>30</td>
</tr>
<tr>
<td>Practice</td>
<td></td>
<td>Oral Part</td>
<td></td>
</tr>
<tr>
<td>Mid-term Exams/Tests</td>
<td>60</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Other Assessment Items</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total:</td>
<td>70</td>
<td>Total: 30</td>
<td></td>
</tr>
</tbody>
</table>
Study Programme: Computer Science

Course title: CLOUD COMPUTING

Teacher/Teachers: Bačanin-Džakula V. Nebojša, Jovanović A. Mlađan

Course Status: Mandatory

ECTS credits: 8

Prerequisites: Programming Languages

Course Objective
The primary goal of this course is to introduce students to the basic principles of Cloud Computing and the essential technologies that enable it - virtualization technology and hyper-converged infrastructure. The secondary goal of this course is to gain experience in implementing cloud services through practical projects using existing public cloud tools. Among other things, another objective of the course the acquisition of necessary skills by the students to carry out research activities in this dynamic field by exploiting existing environments and simulators, such as the CloudSim software package.

Course Outcome
The general competences that students will acquire are the analysis, synthesis and forecasting of solutions and consequences, mastering the methods, procedures and processes of research as well as the application of knowledge in practice. The professional outcome of this course includes the following five components: 1. Understanding the basic concepts of cloud computing paradigm - why and how this paradigm has emerged, its basic characteristics, advantages and challenges of various cloud computing service and delivery models; 2. Training the students for independent implementation of the basic concepts of clustered infrastructure and understanding trade-off between the energy utilization, efficiency of performing cloud operations and the costs, exploring ways to manage cloud computing data center in order to build and implement cloud applications that are resilient, elastic and cost-effective; 3. Introducing the concepts of system, network and data warehouse virtualization, and hyper-converged infrastructure technology, as well as with the role that these technologies have in providing cloud computing services; 4. Introducing the cloud platforms and services from global vendors such are Amazon Web Services (AWS), Microsoft Azure, and Google Compute Engine (GCE).

Course content

Theoretical realization
Cloud computing has transformed the IT industry in recent decades by opening the ability to use "unlimited" and elastic computing resources and delivering enterprise grade applications in the form of software as a service. During theoretical lessons, the students will gain insight into the principles and technologies of cloud computing through the following lesson units (topics): converged and hyper-converged infrastructure, virtualization technology, cloud computing service (SaaS, PaaS, IaaS, XaaS) and delivery (public, private, community, hybrid) models, migrating computing resources to the cloud, scaling physical with the virtual cloud resources, types and categories of cloud storages, disaster recovery and business continuity in the cloud, basic principles and goals of cloud security and utilized technologies for satisfying these goals.

Practical realization
Implementing and configuring virtual machines (instances) in environments of Type 1 (VMware Sphere) and Type 2 hypervisors (Oracle VirtualBox, VMware Workstation), implementing and configuring virtual resources in AWS EC3 (Elastic Computer Cloud) and Microsoft Azure environments, creating and linking applications through secured APIs to Amazon and Microsoft cloud computing platforms, implementation of applications based on micro - services and RESTful Web services using Spring, Spring Boot, Spring Cloud, and hosting them on private or public cloud services.

References

Number of active classes 6
Theoretical classes: 3
Practical classes: 3

Teaching methods: Lectures, computer exercises, consultation hours, practical project-oriented

Final grade (maximum number of points: 100)

<table>
<thead>
<tr>
<th>Pre-exam obligations</th>
<th>Points</th>
<th>Final exam</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Presence</td>
<td>10</td>
<td>Oral exam</td>
<td>30</td>
</tr>
<tr>
<td>Mid-term Exam 1</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mid-term Exam 2 (Project)</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total:</td>
<td>70</td>
<td>Total:</td>
<td>30</td>
</tr>
</tbody>
</table>
**Study Programme:** Computer Science

**Course Title:** WEB APPLICATIONS DEVELOPMENT

**Teacher/teachers:** Jovanović A. Mladen

**Course Status:** Mandatory

**ECTS credits:** 8

**Prerequisites:**

**Course Objective**
The main objective of this course is to familiarize students with the principles of web services, as well as to enable independent development of Web applications and their publishing on the Web using appropriate production environments. In addition, students are expected to understand factors that affect the performance of Web applications, identify points where congestion occurs, and optimize performance.

**Course Outcome**
The general competencies students will acquire on this subject include analytical and systematic approach, as well as the application of appropriate methodologies and technologies in solving complex problems and software development.
The foreseen professional outcome of this course is the ability to develop a three-layer web application that uses HTML5 / CSS3 / JavaScript technologies on the client side (interface), the PHP programming language on the server side, and uses data from the relational database (MariaDB). Optimal selection of components of development and production environment, with emphasis on scalability and understanding of factors that affect performance.

**Course content**

**Lectures**
Internet networks, protocols, services, addressing systems. DNS addressing, domain registration and migration, geographic adaptation of the response. Web service, history and current technologies and protocols. Development and production environments, shared hosting, virtual private servers, content delivery networks, scalability. Usability of Web applications, adaptive and customizable design. Optimization of performance, caching, graphic and multimedia content.

**Labs**

**Literature**

**Number of active teaching hours:** 5 **Lectures:** 3 **Exercises:** 2

**Methods of teaching:** lectures, exercises, project, consultations

**Final grade (maximum number of points: 100)**

<table>
<thead>
<tr>
<th>Pre-exam obligations</th>
<th>Points</th>
<th>Final exam</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attendance/Activity during the course</td>
<td>10</td>
<td>Project defense/Oral exam</td>
<td>30</td>
</tr>
<tr>
<td>Midterm exam 1</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Midterm exam 2</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total:</td>
<td>70</td>
<td>Total:</td>
<td>30</td>
</tr>
</tbody>
</table>
**Study Programme**: Computer science

**Course title**: SOFTWARE DEVELOPMENT

**Teacher/teachers**: Obradović J. Đorđe, Živković Ž. Miodrag

**Course Status**: Elective

**ECTS credits**: 8

**Prerequisite**:

**Aim of the course**

Introduces students to the concepts of software engineering. Enforces an engineering approach to solution of complex problems using problem analysis and solution synthesis. Considers the complexity of software development through phases of software lifecycle. Emphasizes the different aspects which must be accounted during software development, the problems that can be encountered and ways how they can be solved. Mastering of analyzing the user specifications, design, implementation, testing, delivery and software maintenance.

**Outcome of the course**

Students will obtain theoretical knowledge of software engineering, traditional and agile software development procedures. They will be acquainted with all development phases, their roles and mutual connection in obtaining a software product. They will learn how to produce quality software in a systematic, well organized and controlled manner. They will apply acquainted theoretical knowledge working on a specific practical project. Cooperation in teams will enable them to gain experience, not only in application of some methods and techniques they learned, but also to establish proper interpersonal relations which is very important for the final success of the project.

**Contents of the course**

The introduction analyses splitting the process of software development into phases in general, the problems that arise at that time, and then exposes selected traditional and agile development methods. After that, there follows a detailed acquaintance with all phases in software development. In a requirement analysis phase, all types of requests are considered as well as the procedures of their collection, methods of solving conflicts, behavioral modelling using ER diagrams, event traces and finite automata leading to final request specification along with validation and verification. The design phase enforces modularity on different levels using UML language, and several design strategies (client-server architecture, layered architecture, pipes and filters, object-oriented approach, etc.). There follows an introduction to techniques of producing quality programs and internal and external documentation. In the testing phase the stress is put on distinguishing between errors and faults. Three testing techniques are considered into details: unit, integration and system testing. There were analyzed “white” and “black” box methods within unit testing, different integration techniques within integration testing (bottom up and top down, “sandwich” integration, etc.), and functional testing, performance and acceptance testing, and installation testing within system testing. Finally, software delivery and maintenance are considered. Different types of maintenance are analyzed (corrective, preventive, adaptive, and maintenance aimed at system enhancement), as well as problems and expenses during maintenance.

**Exercises**

Taking place in the computer lab. Students work on a project during which they are supposed to pass all the software development phases (according to conditions) and to finally producing working piece of software. They work in small groups and develop team spirit. Different development environments (netBeans, VisualStudio Code) and technologies (PHP, HTML5, CSS, JS, BOOTSTRAP, MariaDB) are used depending on affinities or previous programming skills.

**Literature**


<table>
<thead>
<tr>
<th>Number of active teaching hours</th>
<th>Lectures: 3</th>
<th>Exercises: 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Methods of teaching:</td>
<td>lectures, exercises, project, consultations</td>
<td></td>
</tr>
</tbody>
</table>

**Final grade (maximum number of points: 100)**

<table>
<thead>
<tr>
<th>Pre-exam obligations</th>
<th>Points</th>
<th>Final exam</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attendance/Activity during the course</td>
<td>10</td>
<td>Project defense/Oral exam</td>
<td>30</td>
</tr>
<tr>
<td>Midterm exam 1</td>
<td>30</td>
<td>Total:</td>
<td>30</td>
</tr>
<tr>
<td>Midterm exam 2</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total:</td>
<td>70</td>
<td></td>
<td>30</td>
</tr>
</tbody>
</table>
Study Programme: Computer Science

Course Title: ETHICAL HACKING AND PENETRATION TESTING

Teacher/teachers: Milosavljević M. Milan, Kunjadić Dž. Goran

Course Status: Elective

ECTS credits: 8

Prerequisite:

Course objective

The aim of the course is to introduce students with the basics of ethical hacking that is used to test the system with the Penetration Testing Pen-Test. The aim is for students to get acquainted with elements of a hacker attack that information systems could be exposed to. The taxonomy of the attack will be discussed in detail, and in this way students will gain insight into what methods and techniques are used by hackers. In particular, PTest standards will be processed as well as part of the ethics, and in this way, students will be stressed under what conditions the learned techniques are allowed to use. In addition to the theoretical part, great importance will be given to practical exercises.

Course outcome

The general competences that students will acquire are the analysis and anticipation of possible hacker attacks as well as possible ways of defense as well as making conclusions about the level of security of the system.

The professional outcome of this course includes the following components: 1. Understanding basic concepts of taxonomy of attacks on computer systems as well as attack motives; 2. Introduction to the basic elements of ethics in the broader sense, and especially with the ethics of using information resources, both hardware and software; 3. Training for self-execution of attacks on information systems in order to test the level of system security both on theoretical and practical level; 4. Acquiring knowledge and skills on ways of detecting the vulnerability of information systems in such a way that the systems themselves can not detect scanning; 5. Get acquainted with PTest standards and their practical applications in order to be able to perform Pen-Test on their own.

Content of the course

Theoretical classes

Due to the increasing complexity of information systems and their mutual connections, the defense of the system from hacking attacks is increasingly gaining in importance. The application of methods and techniques of hacker attacks has been used very successfully in detecting the vulnerability of the system, whether attacks come from the environment or from the system itself. During theoretical lessons, students will gain knowledge of the basic concepts of hacking, as well as methods of attacking information systems, whether closed or distributed information systems. In particular, ethical issues concerning the use of offensive methods for checking the system's penetration, that is, the possibility of disturbing the integrity of the system will be addressed. In the theoretical part, hacking motives will also be processed to prevent or prevent potential attacks. The basics of the PTstand standard will also be the subject of theoretical study.

Practical classes in labs

Linux Architecture; File system; Users, permissions, services; System files of relevance to the security of the Linux system; Kali Linux OS; Elements of computer networks and TCP/IP protocols; OSI Model; Computer addressing, routing, ARP, DHCP; VPN; Methodology of attacking computer systems; The terms Vulnerability assessment and Penetration testing; Types of attack (on confidentiality, integrity, availability); The procedure for performing Vulnerability estimation and Pentest; Tools for each stage of the attack; Information gathering; Target discovery; Enumerating target; Metasploit; DOS attack.

Literature


Number of active classes: 6  Theoretical teaching: 3  Practical classes: 3

Teaching methods: Lectures, computer exercises, consultations, practical work in the form of project activities.

Final grade (maximum number of points: 100)

<table>
<thead>
<tr>
<th>Pre-exam obligations</th>
<th>Points</th>
<th>Final exam</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Presence</td>
<td>10</td>
<td>Practical task</td>
<td>30</td>
</tr>
<tr>
<td>Midterm 1</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Midterm 2</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total:</td>
<td>70</td>
<td>Total:</td>
<td>30</td>
</tr>
</tbody>
</table>
**Study Programme:** Computer Science  
**Course Title:** COMPUTER NETWORKS SECURITY  
**Teacher/teachers:** Jevremović D. Aleksandar, Kunjadić Dž. Goran, Adamović Ž. Saša  
**Course Status:** Mandatory  
**ECTS credits:** 8  
**Prerequisites:**

---

### Course Objective
The main objective of this course is to introduce students to the principles of security in computer systems and networks, to get acquainted with the theoretical basics and practical implementation of the most common attacks in computer networks, then to understand human factors that affect security, to understand security risks and solutions that exist in the Web environment, and that for the identified types of attacks, they select and implement adequate security mechanisms.

### Course Outcome
The general competencies students will acquire on this subject include an analytical and systematic approach, an understanding of the concept of value, risk, protection, and security. The foreseen professional outcome of this course is understanding the factors that affect the security of computer systems and networks. Understanding security on the physical, network and application level. Understanding human factors that affect security, social engineering, biometric identification, privacy and anonymity. Protection of the principle of confidentiality, credibility and availability, then authentication, authorization. Implement security mechanisms in the Web environment and understand the specific security risks in it.

### Course content

**Lectures**

**Labs**
Simulation of different types of attacks using Kali distribution of Linux and Metasploit platform. Protect Web applications from SQL injection, XSS and CSRF attacks, using the HSTS mechanism. Understanding the HTTPS protocol and using digital certificates.

### Literature

---

**Number of active classes:** 6  
**Theoretical classes:** 3  
**Practical classes:** 3

**Teaching methods:** Lectures, computer exercises, consultation hours, practical project-oriented

**Final grade (maximum number of points: 100)**

<table>
<thead>
<tr>
<th>Pre-exam obligations</th>
<th>Points</th>
<th>Final exam</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Presence</td>
<td>10</td>
<td>Oral exam</td>
<td>30</td>
</tr>
<tr>
<td>Mid-term Exam 1</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mid-term Exam 2 (Project)</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Total:</strong></td>
<td>70</td>
<td><strong>Total:</strong></td>
<td>30</td>
</tr>
</tbody>
</table>
**Study Programme:** Computer Science

**Course title:** PARALLEL AND CLOUD PROGRAMMING

**Teacher/Teachers:** Bačanin-Džakula V. Nebojša, Jovanović A. Mlađan

**Course Status:** Elective

**ECTS credits:** 8

**Prerequisites:** Programming Languages, Cloud Computing

**Course Objective**

The primary objective of this course is to research principles, tools and programming techniques for the various available parallel cloud platforms. Parallel programming is considered to be the only cost-effective method for quickly solving problems that require reporting services for large amounts of operations or the processing of large amounts of data. The secondary objective of this course is acquiring necessary skills for carrying out research activities in this dynamic field by exploiting existing environments and simulators, such as the Amazon Web Services (AWS) Elastic Compute Cloud (EC2) platform and the CloudSim software package.

**Course Outcome**

The general competences that students will acquire are the analysis, synthesis and the possibility of independent problem solving, mastering the methods, procedures and processes for research activities, as well as the application of knowledge in practice.

The professional outcome of this course includes the following components: 1. Understanding and the ability to discuss the advantages and disadvantages of different parallel architectures and paradigms; 2. Stand-alone implementation of the parallel cloud applications using message passing paradigms; 3. Understanding key problems in the domain of parallel programming; 4. Ability to identify an optimal way to solve a specific problem using parallel programming techniques; 4. Acquiring ability to apply advanced concepts of parallel programming using the message passing paradigm.

**Course content**

**Theoretical realization**


**Practical realization**

Multi-threaded programming in Java programming language. Client-server programming in Java. Implementation of the solutions for the concurrent and parallel programming in Java. Implementing parallel application in the Platform as a Service Platform (PaaS) environments. Implementation of Java SOAP and RESTful Web services using Spring, Spring Boot and Spring Cloud technology.

**References**


<table>
<thead>
<tr>
<th>Number of active classes</th>
<th>Theoretical classes</th>
<th>Practical classes</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>3</td>
<td>5</td>
</tr>
</tbody>
</table>

**Teaching methods:** Lectures, computer exercises, consultation hours, practical project-oriented

**Final grade (maximum number of points: 100)**

<table>
<thead>
<tr>
<th>Pre-exam obligations</th>
<th>Points</th>
<th>Final exam</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Presence</td>
<td>10</td>
<td>Oral exam</td>
<td>30</td>
</tr>
<tr>
<td>Mid-term Exam 1</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mid-term Exam 2 (Project)</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total:</td>
<td>70</td>
<td>Total:</td>
<td>30</td>
</tr>
</tbody>
</table>
Study Programme: Computer Science

Course name: RISK MANAGEMENT IN CYBERSPACE

Teacher/teachers: Jevremović D. Aleksandar, Kunjadić Dž. Goran

Course Status: Elective

ECTS credits: 8

Previous condition:

Course objective

The objective of the course is to introduce students to the basics of risk assessment in cyberspace, as well as ways of their reduction and overcoming. Students will be familiar with the family of ISO / IEC 27000 standards that deal with the information security management area - ISMS (Information Security Management System). The goal is for students to acquire the necessary knowledge to gain the right to an internationally recognized ISO / IEC 27001 certificate. Rising awareness of risks in cyberspace is a basic and initial element of defense in cyberspace, which students will learn and accept during the study of the material of this subject.

Course outcome

General competencies that students will acquire are knowledge about the importance of spotting and tracking information risks in cyberspace as well as methods for accessing this issue.

The professional outcome of this course includes the following components: 1. Familiarization with the family of ISO / IEC 27000 standards in order to understand the risks in cyberspace; 2. Introduction to methods of risk assessment in virtual space and their quantification; 3. Getting to know the possibilities and ways of reducing risks and mitigating the consequences of possible compromise of cyber space; 4. Acquiring knowledge and ability to independently or in the team, perform ISO / IEC 27000 certification, or analyze the safety management compliance in cyberspace with ISO / IEC 27000 standards; 5. Acquiring knowledge and the ability to create and use tools in order to be able to independently check the level of security of cyber space.

Contents of the course

Theoretical classes

With the integration of IT resources, global cyberspace has been created, with each user using his part. By creating global cyberspace, there has been the duplication and complexity of risks that threaten the integrity of the work or the whole of the cyber space. Risk management in this complex environment involves primarily their identification, and then the recognition and application of risk mitigation methods. Risk quantification is a particular challenge and students will be presented theoretical bases of the risk quantification methodology in cyberspace. Ensuring the continuity of cyber space functioning in the event of disorders caused by a conscious or unconscious intervention by humans, either natural causes of environmental impact, is a significant item with which students will be introduced.

Practical classes in labs

Using Script languages to create an executable code to test the risks in cyber space; Creating a code for reading system parameters in a cyber space; Creation of a code for monitoring the activities of users in the cyber space in order to determine the potential risks; Familiarizing with the formal way of determining the level of risk and the degree of security of elements of the cyber space in accordance with ISO / IEC 27000 standards; Practical examples of certification and degree of compliance of elements in cyber space in accordance with ISO / IEC 27000 standards.

Literature


Number of active classes: 5

Theoretical teaching: 2

Practical classes: 3

Teaching methods: Lectures, computer exercises, consultations, practical work in the form of project activities.

Final grade (maximum number of points: 100)

<table>
<thead>
<tr>
<th>Pre-exam obligations</th>
<th>Points</th>
<th>Final exam</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Presence</td>
<td>10</td>
<td>Practical task</td>
<td>30</td>
</tr>
<tr>
<td>Midterm 1</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Midterm 2</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>70</td>
<td>Total:</td>
<td>30</td>
</tr>
</tbody>
</table>

30
Study Programme: Computer science

Course subject: PROGRAMMING AND MOBILE APPLICATIONS DEVELOPMENT

Teacher: Živković Z. Miodrag

Course Status: Elective

ESPB credits: 8

Requirements: Programming Fundamentals, Object-oriented programming

Course objective
The main goal of this subject is to introduce students to the mobile applications development for Android in Java programming language. The secondary course objective is to gain experience in the implementation of mobile applications for Android through practical work and practical projects. Another goal of this subject is that student gain necessary skills for performing research activities in this dynamic area by exploitation of existing Android and Java environments and emulators, such as Android Studio.

Course outcome
General competencies which students will gain are analysis, synthesis, and prediction of results and effects, research methods and processes, as well as knowledge application in practice.

The practical outcome of this subject consists of 1. Understanding of hardware and software platform of mobile devices; 2. Understanding of fundamental concepts of mobile applications development – fundamental characteristics, advantages and challenges, understanding of limitations such as limited battery and energy consumption; 3. Getting students ready for individual application of fundamental concepts of mobile applications and understanding of the balance between energy usage and execution efficiency, in order to implement robust, usable and efficient mobile applications; 4. Ability to individually develop applications for Android.

Course content
Theoretical classes
Mobile applications development has become one of the most important sectors of the IT industry. Java is one of the most popular programming languages, and popular platform Android is based on Java. During theoretical classes, students will gain fundamental theoretical knowledge about principles of mobile application development, and practical skills for writing Android applications in Java programming language through following topics: characteristics of mobile devices, overview of hardware platforms and operating systems, development environments and programming languages, mobile applications architecture, user interface development, activity as a basic Android component, explicit and implicit intents, fragments, storing data (SQLite database, Shared Preferences), AsyncTask and connecting to the Internet, using external API for retrieving data, applications based on Content Provider and Broadcast Receiver, services, working with sensors, localization.

Practical classes in computer laboratories
Practical classes follow theoretical classes, and they are held in computer classrooms on computers with installed Java and Android Studio IDE. During practical classes students will get insight into environment setup, emulator setup, usage of fundamental Android libraries, development of mobile applications for Android and practical application of principles learned during theoretical classes.

Literature

Number of active classes 5

<table>
<thead>
<tr>
<th>Theoretical classes: 3</th>
<th>Practical classes: 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Teaching methods: Lectures, exercises, consultations, midterms, practical projects.</td>
<td></td>
</tr>
</tbody>
</table>

Final grade (maximum number of points: 100)

<table>
<thead>
<tr>
<th>Pre-exam requirements</th>
<th>Points</th>
<th>Final exam</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attendance</td>
<td>10</td>
<td>Individual project</td>
<td>30</td>
</tr>
<tr>
<td>Midterm 1</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Midterm 2</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total:</td>
<td>70</td>
<td>Total:</td>
<td>30</td>
</tr>
</tbody>
</table>
Study Programme: Computer science

Course name: WEB SECURITY

Teacher: Jevremović D. Aleksandar, Adamović Ž. Saša

Course Status: Elective

ECTS credits: 8

Requirement: Web and Internet Technologies

Course objectives
The objective of the course is to introduce students to ways to compromise the security of modern Web applications and techniques to mitigate their vulnerability. The basic issue is the security of the Web application on the server side, including the protection of data privacy, but also the protection of client and server communications, as well as various forms of vulnerability of the client side of the Web application associated with the use of the Web browser and web browsing. Through practical examples, students are being trained to develop and test complete solutions to achieve a high level of Web application security.

Course outcome
The general competences that students will acquire are the analysis, synthesis and forecasting of solutions and consequences, mastering the methods, procedures and processes of research as well as the application of knowledge in practice.

The professional outcome of this course includes: practical experience in web programming. They get the skills to critically review Web applications from the point of view of security vulnerabilities. Design and implement solutions for various types of security bugs. Developing secure Web applications.

Course content

Theoretical teaching
During theoretical lessons students will gain more knowledge about the most important security issues when creating Web applications and what they can do to ensure that servers, software and data are protected. They will be introduced to what motivates hackers and their most common methods of attack, and then present the techniques and ways of thinking needed to make a solution to these security challenges on the Web.

Practical classes in labs
During practical classes, students will learn basic principles that include all security measures for protecting Web services and applications according to the recommendations of OWASP (https://www.owasp.org/index.php/Top_10-2017_Top_10). It will also understand the importance of filtering incoming and outgoing data controls, as well as a smart encryption strategy and user authentication.

Literature

Number of active classes 5
Theory teaching: 3
Practical teaching: 2

Methods of teaching: Lectures, computer exercises, consultations, practical work in the form of project activities.

Final grade (maximum number of points: 100)

<table>
<thead>
<tr>
<th>Pre-exam obligations</th>
<th>Points</th>
<th>Final exam</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attendance</td>
<td>10</td>
<td>Written exam</td>
<td>30</td>
</tr>
<tr>
<td>First midterm</td>
<td>30</td>
<td>Oral exam</td>
<td></td>
</tr>
<tr>
<td>Second midterm</td>
<td>30</td>
<td>Project presentation</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>70</td>
<td>Total:</td>
<td>30</td>
</tr>
</tbody>
</table>
**Study Programme:** Computer Science  
**Course Title:** ARTIFICIAL INTELLIGENCE  
**Teacher:** Milosavljević M. Milan, Jovanović A. Mlađan  
**Course Status:** Elective  
**ECST credits:** 6  
**Prerequisite:** /  

**Course Goals and Objectives**  
Introduction to the basic concepts of artificial intelligence and its place within traditional computer science subfields. Introduction to theoretical and practical insights necessary for the synthesis of knowledge-based artificial intelligence systems, as well as systems based on learning from examples or based on genetic information formed in the evolutionary process within a set of potential solutions.

**Course Outcomes**  
Mastering theoretical and practical knowledge from the domain of analysis and synthesis of artificial intelligence systems within the paradigm of automated reasoning, machine learning, evolutionary and genetic programming.

**Course Content**  

**Theory**  
The notion of artificial intelligence and its history, generic architecture: agents and production systems, problem solving as a search, non-informative search strategy, informative and heuristic search: A and A* algorithm, genetic algorithms, simulated annealing, predicate calculus of the first type, automated reasoning based on resolutions, knowledge-based systems, reasoning in unconstrained conditions, the basics of neuroscience, neuronal network architecture and training algorithms, deep neural networks; the application of deep neural networks over audio, visual and textual data.

**Practice**  
Installing Anaconda with Jupiter Notebook and Python programming environment. Implementation of basic search algorithms within a unique object-oriented paradigm. Experimenting with different heuristic functions. Implementation of basic machine learning algorithms and deep neural networks within the Scikit-learn, SciPy, NumPy, TensorFlow, Theano and neurolab libraries. Training on audio, visual, and textual data sources. Getting acquainted with the principles of the system synthesis for content recommendation systems, systems for extracting emotional states from texts, images and speech, personality model reconstruction (eg. OCEAN) from available information from the web, as well as other modern systems of this class.

**Primary and Secondary Sources Selection**  
1. Milan Milosavljevic, Artificial Intelligence, Singidunum University, 2019.  
4.  

**Active Teaching Hours:** 4  
- **Theory:** 2  
- **Practice:** 2

**Applicable Teaching Methods:**  
Lectures, practice, seminar papers, midterms, final exam

**Final grade (maximum number of points: 100)**  

<table>
<thead>
<tr>
<th>Pre-exam Requirements</th>
<th>Points</th>
<th>Final Exam</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attendance</td>
<td>10</td>
<td>Written exam</td>
<td>30</td>
</tr>
<tr>
<td>First midterm</td>
<td>30</td>
<td>Oral exam</td>
<td></td>
</tr>
<tr>
<td>Second midterm</td>
<td>30</td>
<td>Project presentation</td>
<td></td>
</tr>
<tr>
<td>Other Assessment Items</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total:</td>
<td>70</td>
<td>Total:</td>
<td>30</td>
</tr>
</tbody>
</table>
Study Programme: Computer science

Course subject: PRACTICUM – MOBILE AND WEB APPLICATIONS DEVELOPMENT

Teacher: Živković Z. Miodrag

Course Status: Elective

ESPB credits: 8

Requirements: Programming Fundamentals, Object-oriented programming

Course objective

The main goal of this subject is to introduce students to the fundamental principles of web applications and mobile applications development, with the accent on ASP .NET and iOS platforms. The secondary course objective is to gain experience in implementation of web applications and mobile applications for iOS through practical work and practical projects. Another goal of this subject is that student gain necessary skills for performing research activities in this dynamic area by exploitation of existing environments and emulators, such as Visual Studio.

Course outcome

General competencies which students will gain are analysis, synthesis, and prediction of results and effects, research methods and processes, as well as knowledge application in practice.

The practical outcome of this subject consists of 1. Understanding fundamental concepts of web applications; 2. Understanding of basic protocols behind web services; 3. Understanding multi-tier architecture of web applications; 4. Understanding of fundamental concepts of mobile applications development – fundamental characteristics, advantages and challenges, understanding of limitations such as limited battery and energy consumption; 5. Getting students ready for individual development of web applications and applications for Apple phones.

Course content

Development of web applications and mobile phones applications is one of the most important sectors of software development. Programming language C# and ASP.Net framework are one of the most popular solutions in the industry. During theoretical classes, students will gain fundamental theoretical knowledge about the development of web and mobile applications through the following topics: concepts of multi-tier web applications, introduction to C#, object-oriented aspects of C#, classes, objects, inheritance, introduction to .Net framework, web forms, ASP.Net web services, MVC pattern, controller development, implementation of RESTful Web API, data access with ADO.Net, introduction to mobile application development, introduction to iOS application development.

Practical classes computer laboratories

Practical classes follow theoretical classes, and they are held in computer classrooms on computers with installed Visual Studio IDE. During practical classes, students will get insight into environment setup, emulator setup, usage of fundamental iOS libraries, development of mobile applications for Apple, development of web applications in ASP.Net framework and practical application of principles learned during theoretical classes.

Literature


Number of active classes

<table>
<thead>
<tr>
<th>Theoretical classes:</th>
<th>Practical classes:</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>2</td>
</tr>
</tbody>
</table>

Teaching methods: Lectures, practices, consultations, midterms, individual project.

Final grade (maximum number of points: 100)

<table>
<thead>
<tr>
<th>Pre-exam requirements</th>
<th>Points</th>
<th>Final exam</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attendance</td>
<td>10</td>
<td>Individual project</td>
<td>30</td>
</tr>
<tr>
<td>Midterm 1</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Midterm 2</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total:</td>
<td>70</td>
<td>Total: 30</td>
<td></td>
</tr>
</tbody>
</table>
**Study Programme:** Computer science

**Course name:** PRACTICUM - PROTECTION OF INFORMATION SYSTEMS

**Teacher:** Adamović Ž. Saša, Kunjadić Đ. Goran

**Course Status:** Elective

**ECTS credits:** 8

**Requirement:**

**Course objective**

The objective of this course is to introduce students to the standard program libraries for the implementation of cryptographic mechanisms - methods for generating symmetric and asymmetric keys, modern symmetric and asymmetric cipher, key distribution protocols and other authentication services, non-integrity and secrecy services. The implementation of all the above mentioned mechanisms is predominantly planned in the Java programming language, where students will further improve their programming skills and acquire knowledge for the correct implementation of cryptographic mechanisms that are the basis of the development of their own information security solutions.

**Course outcome**

The general competences that students will acquire are the analysis, synthesis and forecasting of solutions and consequences, mastering the methods, procedures and processes of research as well as the application of knowledge in practice.

The professional outcome of this course includes: training students to adopt the fundamental concepts of information systems security at the theoretical and practical level, and to make connections with the criteria essential for the analysis and synthesis of modern security systems, which as a rule work in a distributed, insecure information and communication environment. Knowledge of the application of cryptographic solutions in modern systems of e-business, e-government and other professional systems.

**Course content**

*Theoretical teaching*

Introduction to standardized cryptographic software architectures. Correct procedures for generating the cryptographic parameters necessary for modern symmetric ciphers with corresponding encryption modes. Generating and distributing asymmetric keys. Key management and certificates. Different computer and information protocols for key exchange through public channels. Standardized battery tests for cryptographic random generator quality assessment.

*Practical classes in labs*

The use of Java cryptographic libraries JCA, JCE, and BounceCastle in the Netbeans programming environment. Calling and correctly entering parameters into software solutions for generating symmetric and asymmetric keys, as well as their proper use in modern symmetric and asymmetric ciphers (AES, DES, RSA). Development of own random source generators based on adequate information sources and application of standard NIST battery tests for testing the quality of generated values for application in the domain of cryptographic parameters. Introduction to specific foreign and domestic protection solutions. Designing security scheme for the selected real system. Documentation of the realized solution.

**Literature**

2. Phil Smith, Practical Cryptography in Java, 2015, p. 86.

**Number of active classes 8**

<table>
<thead>
<tr>
<th>Methods of teaching</th>
<th>Theory teaching: 2</th>
<th>Practical teaching: 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Final grade (maximum number of points: 100)</td>
<td>Points</td>
<td>Final exam</td>
</tr>
<tr>
<td>Pre-exam obligations</td>
<td>Attendance</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>First midterm</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>Second midterm</td>
<td>30</td>
</tr>
</tbody>
</table>
Study Programme: Computer science

Course subject: SOFTWARE TESTING

Teacher: Obradović J. Đorđe

Course Status: Elective

ECTS credits: 8

Requirements: Programming Fundamentals, Object-oriented programming.

Course objective
The main goal of this subject is to introduce students to the fundamental principles of software testing, with a focus on programming language Java and JUnit tool which is used for unit testing of Java programs. The secondary course objective is to gain experience in implementation of automated unit tests for Java programs through practical work and practical projects. Another goal of this subject is that student gain necessary skills for performing research activities in this dynamic area by exploitation of existing environments used for testing, such as Selenium for web application testing.

Course outcome
General competencies which students will gain are analysis, synthesis, and prediction of results and effects, research methods and processes, as well as knowledge application in practice. Students will gain theoretical knowledge about different testing methodologies, and their application in software development.

The practical outcome of this subject consists of 1. Understanding different aspects of testing; 2. Understanding fundamental concepts of software testing: black box and white box testing; 3. Understanding different levels of testing – unit, integration, and system testing; 4. Getting students ready for individual application of fundamental software testing concepts, in order to ensure that tested applications are robust, usable and efficient; 5. Ability to develop tests in JUnit and Selenium tools.

Course content
Software testing is one of the most important activities in the software development process. During theoretical classes, students will gain fundamental theoretical knowledge about software testing principles through the following topics: software testing levels, introduction to JUnit, black box techniques (equivalence classes partitioning, boundary values analysis, cause-effect graph, decision table, state model), white box techniques (control flow graph, code coverage, statement coverage, decision coverage, path coverage, cyclomatic complexity of program, data flow testing), integration testing, regression testing, system testing, static testing, role of testing in different software development models, GUI testing, web application testing with Selenium, testing object-oriented programs, defect management.

Practical classes computer laboratories
Practical classes follow theoretical classes, and they are held in computer classrooms on computers with installed Java IDE. During practical classes, students will get insight into environment setup, JUnit and Selenium setup, and practical application of principles learned during theoretical classes.

Literature

Number of active classes 6
Theoretical classes: 3
Practical classes: 3

Teaching methods: Lectures, practices, consultations, midterms, practical project activities.

Final grade (maximum number of points: 100)

<table>
<thead>
<tr>
<th>Pre-exam requirements</th>
<th>Points</th>
<th>Final exam</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attendance</td>
<td>10</td>
<td>Individual project</td>
<td>30</td>
</tr>
<tr>
<td>Midterm 1</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Midterm 2</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total:</td>
<td>70</td>
<td>Total:</td>
<td>30</td>
</tr>
</tbody>
</table>
**Study Programme:** Computer science

**Course name:** SECURE AND ROBUST PROGRAMMING

**Teacher:** Jevremović D. Aleksandar, Kunjadić Dž. Goran, Adamović Ž. Saša

**Course Status:** Elective

**ECTS credits:** 8

**Requirement:** Programming languages

**Course objective**
The objective of the course is to familiarize and understand all security aspects of the software, including design, secure implementation and maintenance. The subject will address the following topics: authentication and authorization, security control, TOCTOU vulnerability, data entry control, memory management, fixing weaknesses, and distributing security patches. The course requires previous experience in programming and working with computer systems.

**Course outcome**
The general competences that students will acquire are the analysis, synthesis and forecasting of solutions and consequences, mastering the methods, procedures and processes of research as well as the application of knowledge in practice.

The professional outcome of this course includes theoretical and practical knowledge necessary for the development of secure software, understand the causes that lead to security failures and the ways in which they are exploited—they are abused, the skill of using the security-oriented software techniques and informing about the current affairs in this domain through the reading of scientific Articles from relevant science journals.

**Course content**

*Theoretical teaching*

Theoretical instruction includes introduction to security aspects and support by programming languages (encapsulation, exception management, secure encryption, various security errors), system programming (data types, data structures, flow control, library, OOP), string security (different string vulnerabilities and exploit), dynamic memory management, intrusion security, formatted outputs, competitive programming, file management, platform security (.Net, JRE) and code quality assessment tools.

*Practical classes in labs*

The practical teaching of secure and robust programming will be in the form of research and identification of code vulnerability. System programming languages will be used, and the focus will also be on security features in other programming languages - runtime environments (Java, JVM). Students work on projects where they can find and fix security vulnerabilities in the application (https://www.sans.org/top25-software-erro/) and / or develop an application that demonstrates the use of strong security mechanisms.

**Literature**


**Number of active classes 6**

<table>
<thead>
<tr>
<th>Theory teaching:</th>
<th>3</th>
<th>Practical teaching:</th>
<th>3</th>
</tr>
</thead>
</table>

**Methods of teaching:** Lectures, computer exercises, consultations, practical work in the form of project activities.

**Final grade (maximum number of points: 100)**

<table>
<thead>
<tr>
<th>Pre-exam obligations</th>
<th>Points</th>
<th>Final exam</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attendance</td>
<td>10</td>
<td>Written exam</td>
<td>30</td>
</tr>
<tr>
<td>First midterm</td>
<td>30</td>
<td>Project presentation</td>
<td></td>
</tr>
<tr>
<td>Second midterm</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total:</td>
<td>70</td>
<td>Total:</td>
<td>30</td>
</tr>
</tbody>
</table>
Study Programme: Computer Science
Course Title: ENGLISH LANGUAGE 4
Teacher: Petrović D. Jasna
Course Status: Compulsory
ECST credits: 6
Prerequisite: /

Course Goals and Objectives

English Language 4 is a Course that follows the English Language 3 course – i.e. its form and content. The Course aims at enhancing students’ skills and competences regarding all the language skills – and placing them within the borders of C1-C1+ CEFR level. In this way, the Course intends to enable students to communicate in different language contexts and for different purposes – general, specific and academic.

Course Outcomes

Upon completing the English Language 4 Course, students are capable of understanding and transmitting the intended messages in English in a confident and effective manner. In that communication process – students exhibit the knowledge of language belonging to C1-C1+ CEFR level – i.e. they are capable of performing rather demanding language tasks related to study and work endeavours. Moreover, students are pragmatically aware of specific needs in different language contexts.

Course Content

Grammar-related Units

Revising and shedding some new light on – Conditional Sentences – III Types, Cleft Sentences, Indirect Speech, Clefting, Infinitives, Participle, Special Case Passive and Causative, Mood in English

Lexical Units

Metaphor and Metonymy, Similarities and Differences in Different Registers, Language of the Media, Back Formation, Conversion; ESP – E-marketing, E-commerce, E-banking, Cryptology

Primary and Secondary Sources Selection

2. Murphy, R. English Grammar in Use (Book with answers and Interactive ebook), CUP, 2015

Active Teaching Hours: 3

Theory: 2

Practice: 1

Applicable Teaching Methods:
Interactive lectures – gamified and collaborative teaching/learning, humanistic and personalized methods, language skills’ assessment, tests, midterm exams, written and oral exam

Final grade (maximum number of points: 100)

<table>
<thead>
<tr>
<th>Pre-exam Requirements</th>
<th>Points</th>
<th>Final Exam</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attendance and in-class Activity</td>
<td>10</td>
<td>Written part</td>
<td>30</td>
</tr>
<tr>
<td>Practice</td>
<td></td>
<td>Oral Part</td>
<td></td>
</tr>
<tr>
<td>Mid-term Exams/Tests</td>
<td>60</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Other Assessment Items</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total:</td>
<td>70</td>
<td>Total: 30</td>
<td></td>
</tr>
</tbody>
</table>
**Study Programme:** Computer Science

**Course Title:** PRACTICAL RESEARCH AND SOFT SKILLS

**Teacher:** Dobrijević M. Gordana, Đorđević-Boljanović Đ. Jelena, Cvetković M. Dragan

**Course Status:** Mandatory

**ECST credits:** 8

**Prerequisite:** No requirements

**Course Goals and Objectives**

The objective of the course is to introduce students with basic concepts and categories in the field of applied skills with a focus on solving specific problems. The course aims at mastering skills through active teaching supplemented by the implementation of practice in a particular business society, followed by writing and defense of an independent project. Training students for independent work in the economy, deepening and application of knowledge in the field of professional development and business skills, and the application of theoretical knowledge in concrete case studies are the focus of this course.

**Course Outcomes**

The student will be able to understand and apply key skills, basic principles for achieving goals, the role of lifelong training in achieving success and positive business results, and the individual aspect of the role of an individual as an employee in the organization. On the basis of the problem that was processed, the student is trained to write and defend the project - the final work on the studies.

**Course Content**

*Theory*

Basic concepts of applicable business skills and theory of success; Personal skills; Introspection skill; Emotional and Social Intelligence; Time management skills; Social skills: negotiation skills, conflict management skills and conflict situation skills, stress management skills and stress reactions; Communication skills: oral, written and non-verbal communication; formal and informal business correspondence; Public speaking skills and oratory skills; Academic skills: academic writing, rules for writing academic papers, the skills of presenting academic papers; Sales skills; Analytical skills.

**Primary and Secondary Sources Selection**


**Active Teaching Hours:** 3  
Theory: 3  
Practice: 0

**Applicable Teaching Methods:**

lectures, practice, seminar and project paper, case analysis, simulation, interactive discussion, midterms, written exam

**Final grade (maximum number of points: 100)**

<table>
<thead>
<tr>
<th>Pre-exam Requirements</th>
<th>Points</th>
<th>Final Exam</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attendance and in-class Activity</td>
<td>10</td>
<td>Written part</td>
<td></td>
</tr>
<tr>
<td>Practice</td>
<td></td>
<td>Oral Part</td>
<td>60</td>
</tr>
<tr>
<td>Mid-term Exams/Tests</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Other Assessment Items</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>40</td>
<td>Total: 60</td>
<td></td>
</tr>
</tbody>
</table>