Course title	Mathematics 1						
Course instructor(s)	Ivana Marušić, lecturer						
Programme(s) of study	Undergraduate professional programme of study in Mechatronics Undergraduate professional programme of study in Computer Science						
Course status	Compulsory						
Year	1st	Semester	1st	ECTS	7		
Contact hours (L+PS+S)	30+45+0	L	PS		S		
			APS	LPS			
		30	45	0	0		

Course objectives

Familiarising students with terms related to mathematical analysis and linear algebra, as well as the fundamentals of differential calculus that are necessary for attending, understanding and application in general and professional courses.

Expected learning outcomes

Upon completion of the course students will be able to:

- O1: describe sets of natural numbers, integers, rational numbers, irrational numbers, real numbers and complex numbers, and apply arithmetic operations in the aforementioned sets of numbers,
- O2: explain the terms matrix and determinant, name their properties and use them in the matrix and determinant calculus,
- O3: distinguish between the methods of solving linear equation systems and apply a suitable method in solving a concrete system,
- O4: reproduce basic terms of vector algebra and analytical geometry and apply them while solving tasks,
- O5: analyse and solve a task of medium difficulty in the area of mathematical analysis,
- O6: Define and name the properties of arithmetic and geometric sequence and calculate limits of sequences and functions,
- O7: apply the fundamentals of differential calculus in simple tasks.

Course content

1. Sets (Outcome O1)
The notion of set. Subset. Equality of sets. Cardinal number of a set. Operations with sets.

2. Real and complex numbers (Outcome O1)
Set $\mathbb{N}$. Set $\mathbb{Z}$. Properties of sets $\mathbb{N}$ and $\mathbb{Z}$. Binomial theorem. Set $\mathbb{Q}$. Set $\mathbb{I}$. Properties of sets $\mathbb{Q}$ and $\mathbb{I}$. Set $\mathbb{R}$. Properties of set $\mathbb{R}$. Intervals of real numbers. Absolute value of a real number. Set $\mathbb{C}$. Basic operations with complex numbers. Real and imaginary part of complex numbers. Conjugation of complex numbers. Module of complex numbers. Algebraic form of complex numbers. Equality of complex numbers. Trigonometric form of complex numbers. Exponentiation, inverse exponentiation and division of complex numbers. Equations in the set of complex numbers. Complex plane. System of equations in the set of complex numbers.

3. Linear algebra (Outcome O2, O3)
Definition and special forms of matrices. Basic operations with matrices. Matrix polynomial. Multiplication of matrices. Commutative property of matrices. System of linear equations. Rank of a matrix. Determinant of a matrix. Properties of a determinant. Rule of Sarrus. Laplace expansion of the determinant of the n-th order. Invertible matrix. Computing inverse matrices by applying the Gauss-Jordan method. Computing inverse matrices by means of a determinant. Cramer's rule. Matrix equation.

4. Vector algebra and analytical geometry (Outcome O4)
Coordinate system in space. Scalar product. Vector product. Linear combination of vectors. Area and height of triangle. Parallelogram area. Mixed product. Volume and height of parallelepiped. Volume of tetrahedron. Plane equation. Line equation. Intersection of a line and a plane. Intersection of two lines. Orthogonal projection of a point to the line. Orthogonal projection of a line to the plane. Distance between points. Distance between a line and a plane. Distance between a point and a line. Distance between parallel lines. Distance between skew lines.

5. Functions (Outcome O5)

The notion of function. Equality of functions. Manners of specifying a function. Function properties. Elementary functions and their properties. Function composition. Inverse function and domain. Inverse trigonometric functions.

6. Sequences and limit of sequence (Outcome O6)

The notion of sequence. Arithmetic sequence. Geometric sequence. Properties of sequences. Limit of sequence.

7. Limes and continuity of a function (Outcome O6)

Limes of a function. Continuity of a function. Asymptotes.

8. Derivative (Outcome O7)

The issue of speed. The notion of derivative. Derivatives of some elementary functions. Basic rules of derivation.

Required reading

- Tomić, Milorad: Matematika 1, Technical College in Bjelovar, Bjelovar, 2009
- Tomić, Milorad: Matematika 2, Technical College in Bjelovar, Bjelovar, 2009
- Marušić, Ivana: "Presentation for lectures and practical sessions – Mathematics 1", Bjelovar University of Applied Sciences, Bjelovar, 2017, available on: http://vub.hr/1-godina-matematika-predavanja-vjezbe/predavanja/

Further reading

- Pavlovič Demidović, Boris, et al.: "Zadaci i riješeni primjeri iz Matematičke analize za tehničke fakultete", Golden marketing, Tehnička knjiga, Zagreb, 2003

Course title	Fundamentals of Engineering Calculus						
Course instructor(s)	Ivana Marušić, lecturer						
Programme(s) of study	Undergraduate professional programme of study in Mechatronics Undergraduate professional programme of study in Computer Science						
Course status	Compulsory						
Year	1st	Semester	1st	ECTS	2		
Contact hours (L+PS+S)	0+30+0	L	PS		S		
			APS	LPS			
		0	30	0	0		

Course objectives

Familiarising students with the latest concepts of mathematical analysis for the purpose of upgrading the knowledge of concepts from Mathematics 1.

Expected learning outcomes

Upon completion of the course students will be able to:

- O1: apply mathematical methods and physical laws that have application in the engineering profession,
- O2: recognise physical quantities,
- O3: convert base and derived physical units,
- O4: interpret and graph in kinematics.

Course content

1. Power, algebraic expressions, root (Outcome O1)

Power application. Application of algebraic expressions. Equations. Functions. Polynomials and rational functions. Roots. **2. Arrangement in the set of real numbers (Outcome O1)**

Intervals. Inequalities. Absolute value of a real number.

3. Coordinate system in a plane (Outcome O1)

Distance between points in a plane. Midpoint. System of linear equations. Line. Graph of a function.

4. Geometry (Outcome O1, O2)

Points, lines and planes. Angle. Triangle. Trapezium. Isosceles trapezium. Parallelogram. Rhombus. Rectangle. Square. Circle. Disk. Perimeter and area.

5. Trigonometry (Outcome O1, O2, O3)

Right-angled triangle trigonometry. Definitions of trigonometric functions. Properties of trigonometric functions. Application of trigonometric functions.

6. Physical quantities and units (Outcome O1, O2, O3)

SI base units. SI additional units. SI derived units with a special name. Allowed units outside the SI. Prefixes of SI units.

7. Graphs in physics (Outcome O1, O4)

Interpretations of graphs in kinematics.

Required reading

- Pisačić, Katarina: "Osnove inženjerskog proračuna", University North, Varaždin,2014, available on: http://unin.hr/~kpisacic/PA1_vjezbe.pdf
- Marušić, Ivana: "Osnove inženjerskih proračuna", Bjelovar University of Applied Sciences, Bjelovar, 2017, available on: http://vtsbj.hr/osnove-inzenjerskog-proracuna/

Further reading

- Bronštejn, Ilja Nikolajevič; Semendjajev, Konstantin Adolfovič: "Matematički priručnik za inženjere i studente", Tehnička knjiga, Zagreb, 1964

Course title	Fundamentals of Electrical Engineering and Electronics						
Course instructor(s)	Robert Herčeki, lecturer						
Programme(s) of study	Undergraduate professional programme of study in Computer Science						
Course status	Compulsory						
Year	1st	Semester	1st	ECTS	6		
Contact hours (L+PS+S)	30 + 30 + 0	L	PS		S		
			APS	LPS			
		30	30	0	0		

Course objectives

Familiarising students with basic knowledge and problem solving in the area of electrical engineering. Familiarising students with the operation of fundamental electronic components and assemblies.

Expected learning outcomes

Upon completion of the course students will be able to:

- O1: define the basic terms of electrostatics,
- O2: define the basic terms of electric circuits,
- O3: analyse DC electric circuits using Ohm's law and Kirchoff's laws,
- O4: describe the operation principle and application of semiconductor diodes,
- O5: describe the operation of transistor switches.

Course content

1. Electric charge and electric field (Outcome O1)
Coulomb's law. Electric field of the point charge. Potential and voltage in electric field.
2. Basic terms related to electric circuits (Outcome O1)
Electric current, current density, resistance and conductance.
3. Electric circuits of direct current (Outcome O2, O3)
Charge flow. Ohm's law. Temperature dependence of resistance. Connecting resistors, connection in series, connection in parallel and mixed connection. Kirchhoff's laws. Current and voltage real sources. Power and energy of electric current.
4. Electric capacitance (Outcome O3)
Fundamentals of electric capacitance. Types of capacitors. Connection in series, connection in parallel and mixed connection of capacitors.
5. Basic properties of semiconductors (Outcome O4)
Electrical properties of semiconductors. Types of carriers. Types of semiconductors. Current conduction in semiconductors. Generation and recombination. Concentrations of carriers in semiconductors.
6. P-N diode (Outcome O4)
P-N junction in equilibrium. P-N junction under voltage. Static characteristics of diodes. Breaking the P-N barrier. Classification of diodes. Semiconductor diode as a switch. Diode circuits in analogue and pulse electronics. Light Emitting Diodes.
7. Bipolar transistors (Outcome O5)
Operation principles. Areas of operation. Static characteristics. Choosing a location of an operating point. Transistor as a switch. Pulse circuits.
8. Unipolar transistors (Outcome O5)
Operation principle. MOSFET transistor as a switch. Comparison between unipolar and bipolar transistors.

Required reading

- Kuzmanović, Branislav: Osnove elektrotehnike 1, Zagreb, Element, 2006
- Kuzmanović, Branislav: Osnove elektrotehnike 2, Zagreb, Element, 2006
- Butković, Željko; Divković-Pukšec, Julijana; Barić, Adrijan: Elektronika 1 – class material, Faculty of Electrical Engineering and Computing, Zagreb, 2010

Further reading

- Robert L. Boylestad: INTRODUCTORY CIRCUIT ANALYSIS, ISBN: 0-13-173044-4

<table>
<tr><td colspan="2">Course title</td><td colspan="6">Introduction to Informatics</td></tr>
<tr><td colspan="2">Course instructor(s)</td><td colspan="6">Dario Vidić, lecturer
Ivan Sekovanić, lecturer</td></tr>
<tr><td colspan="2">Programme(s) of study</td><td colspan="6">Undergraduate professional programme of study in Mechatronics
Undergraduate professional programme of study in Computer Science</td></tr>
<tr><td colspan="2">Course status</td><td colspan="6">Compulsory</td></tr>
<tr><td>Year</td><td>1st</td><td>Semester</td><td>1st</td><td>ECTS</td><td colspan="3">4</td></tr>
<tr><td rowspan="2">Contact hours (L+PS+S)</td><td rowspan="2" colspan="2">15 + 30 + 0</td><td>L</td><td colspan="2">PS</td><td rowspan="2">S</td></tr>
<tr><td>15</td><td>APS</td><td>LPS
30</td></tr>
<tr><td colspan="7">Course objectives</td></tr>
<tr><td colspan="7">Familiarising students with information technologies and their application.</td></tr>
<tr><td colspan="7">Expected learning outcomes</td></tr>
<tr><td colspan="7">Upon completion of the course students will be able to:
O1 analyse the historical development and current significance of computer systems,
O 2 describe and recognise the main components of a computer system, their functions and the role of numeral systems in computer operation,
O 3 describe the role of computer operating systems,
O 4 create a simple database,
O 5 edit and format texts,
O 6 process data in table calculations,
O 7 describe adjustment and maintenance techniques related to the Windows operating system and its security settings.</td></tr>
<tr><td colspan="7">Course content</td></tr>
<tr><td colspan="7">1. The notion of computer literacy (I1)
Innovations that caused the development of information technologies. Numeral systems.
2. Assemblies and architecture of personal computers (I2)
Von Neumman‘s model of a digital computer. CPU processor. Memory (RAM, ROM-BIOS, CASHE). Input units (hard disk, CD, DVD, modem, USB, keyboard, mouse, touchpad, scanner, web-camera, digital camera). Output units (monitor, projector, printers, plotters).
3. Operating systems (I3)
Early beginnings and development. Operating systems Windows and Linux – characteristics and installation. Operating environment in operating systems – GUI, Kernel.
4. Maintenance and security of the operating systems Windows (I7)
Administration tools. Antivirus protection. Firewall.
5. MS Office tools (I4, I5, I6)
Word. Excel. Access.</td></tr>
<tr><td colspan="7">Required reading</td></tr>
<tr><td colspan="7">1. Dario Vidić, Ivan Sekovanić: Presentations for lectures and practical sessions in the course “Introduction to Informatics”, Bjelovar University of Applied Sciences
2. Šimović, Maletić, Afrić: Osnove informatike, Zagreb, 2010</td></tr>
<tr><td colspan="7">Further reading</td></tr>
<tr><td colspan="7"></td></tr>
</table>

Course title	**Introduction to Programming**						
Course instructor(s)	Ivan Sekovanić, lecturer						
Programme(s) of study	Undergraduate professional programme of study in Computer Science						
Course status	Compulsory						
Year	1st	**Semester**	1st	**ECTS**	6		
Contact hours (L+PS+S)	30 + 30 + 0	L	PS		S		
			APS	LPS			
		30		30			

Course objectives

1. Developing the ability of algorithmic approach to problem solving.
2. Acquiring basic knowledge of programming in *Python*.
3. Developing logical thought processes in the problem-solving process.

Expected learning outcomes

Upon completion of the course students will be able to:

O1: distinguish between simple types of data and subject them to programming operations,
O2: apply consecutive, selection commands and loops within a programming code,
O3: use complex types of data and implement more complex programming operations,
O4: write programming functions with or without return value that performs various operations,
O5: use programming commands for writing and reading data from a textual file.

Course content

1. Programming languages and programming (O1)
The history of programming languages. Potential applications of programming. Program development. Thinking like a programmer.
2. Programming language *Python* (O1)
Introduction to *Python*. Setting up *Python* on *Windows*. Programming a "Hello World" program. *Pythona* syntax.
3. Handling simple data (O1)
Standard data types. Variables. Display of numbers. Arithmetic operators. Operators of comparison. Assignment operators. Logical operators. Bitwise operators. Priority of operators. Data type conversions. Mathematical functions. Simple input-output functions.
4. Control during program execution (O2)
Conditional execution. Conditional commands with one sentence and several sentences. Nesting of conditional commands. Programming loop. While loop. For loop. Break and continue commands. Pass command.
5. Strings (O3)
String data. Special signs in strings. Formatting strings. Operations on strings.
6. Collections of objects (O4)
Consecutive collections. Lists. Operations on lists. Matrices. N-tuples. Operations on n-tuples. Associative collections. Dictionaries. Operations on dictionaries. Sets. Operations on sets.
7. Functions (O4)
Definition of a function. Calling a function. Function arguments. Return values of functions. Variable scope.
8. Files (O5)
Operations with files. Reading and writing textual and binary files.

Required reading

1. Alan Mutka, Ivan Sekovanić: Presentations for lectures and practical sessions in the course "Introduction to Programming", Bjelovar University of Applied Sciences
2. Zoran Kalafatić, Antonio Pošćić, Siniša Šegvić, Julijan Šribar: Python za znatiželjne, Element, Zagreb, 2016

Further reading

1. Michael Dawson: Python Programming for the Absolute Beginner, 3rd Edition, Course Technology, Boston, 2010

Course title	Communication Skills							
Course instructor(s)	Tatjana Badrov, MSc, senior lecturer							
Programme(s) of study	Undergraduate professional programme of study in Mechatronics Undergraduate professional programme of study in Computer Science							
Course status	Compulsory							
Year	1st	Semester	1st	ECTS	3			
Contact hours (L+PS+S)	15+30+0	L	PS		S			
			APS	LPS				
		15	30					

Course objectives

Improving students' communication skills.

Expected learning outcomes

Upon completion of the course students will be able to:

- O1: explain basic concepts, types and difficulties in the field of communication,
- O2: distinguish between the techniques of active listening,
- O3: apply information collection techniques and a balanced feedback,
- O4: identify and compare basic communication styles and apply an assertive I-message,
- O5: distinguish between three types of complaints and complaint resolving techniques,
- O6: prepare and deliver a presentation on a given topic,
- O7: describe and demonstrate the basic elements of the negotiation process,
- O8: explain debate principles and participate in a debate on a given topic.

Course content

1. Introduction to communication (Outcome 1)
The notion of communication. Levels of communication phenomena (intrapersonal, interpersonal, group, public and mass communication). Objectives, principles and types of communication.
2. Verbal communication (Outcome 1)
Factors of efficiency and suitability of verbal communication. Prejudices about communication. Aspects of messages. Criteria of successful verbal communication. Connotative and denotative level of importance of verbal communication.
3. Non-verbal and paraverbal communication (Outcome 1)
Types, characteristics and functions of paraverbal communication. Types and functions of non-verbal messages. Communication skills in business communication.
4. Problems in communication (Outcome 1)
The communication process. Noisiness, noise, barriers to communication. Types of noise, external and internal barriers to communication.
5. Techniques and skills of active listening (Outcome 2)
Listening as a physical and mental activity. Types of (not) listening. Principles of active listening.
6. Skills of information gathering (Outcome 3)
Techniques and skills of asking questions. Types of questions according to the objective of communication.
7. Techniques of giving feedback (Outcome 3)
The concept and purpose of feedback. The five main categories of feedback. A balanced feedback. Receiving and giving praise.
8. Communication styles (Outcome 4)
Aggressive, submissive-aggressive, passive, assertive communication style. The link between communication styles and the outcome of communication. Notion and meaning of assertiveness. Principles of assertive communication. Structure and effect of I-messages in relation to the YOU-message.
9. Recognizing and resolving complaints (Outcome 5)
Notion of complaints. Types of complaints. General rules for resolving complaints. Resolving complaints with regard to the type of complaint.
10. Self-introducing and management of impressions (Outcome 6)

Skills of impressions management. Five main strategies of self-introducing.

11. Presentation techniques and skills (Outcome 6)

Preparation and design of presentations. Delivery structure. Verbal and non-verbal elements of delivery in front of an audience. Answering questions.

12. Negotiation (Outcome 7)

Definition of negotiation and negotiation situations. Characteristics of successful negotiators. Preparing for negotiations. Negotiating strategies. Tactics and techniques for the initial, central and final phase of negotiations. Unethical techniques / tactics in negotiations.

13. Debate (Outcome 8)

Notion of debate. Participants in a debate. Debate parts. Debate in the development of critical, logical and creative thinking.

Required reading

- Reardon, K. (1998): Interpersonalna komunikacija – Gdje se misli susreću, Alineja, Zagreb
- Course materials available in the e-learning system Merlin

Further reading

- Fox, R. (2006): Poslovna komunikacija, Hrvatska sveučilišna naklada, Zagreb

Course title	**Technical English 1**							
Course instructor(s)	Ivana Jurković, senior lecturer							
Programme(s) of study	Undergraduate professional programme of study in Computer Science							
Course status	Compulsory							
Year	1st	**Semester**	1st	**ECTS**	2			
Contact hours (L+PS+S)	15 + 30 + 0	L	PS		S			
			APS	LPS				
		15	30	0	0			

Course objectives

Developing students' ability to use the English language related to specific technical fields.

Expected learning outcomes

Upon completion of the course students will be able to use the English language to:

- O1: describe technical functions and applications of products and the manner in which products function,
- O2: describe the properties and application of materials used in various fields of engineering,
- O3: describe the shape and features of components and assemblies as well as joining and fixing procedures,
- O4: demonstrate mastery of simple grammatical structures.

Course content

1. Technology applications (Outcome O1)
Description of technical functions and applications.
Emphasising technical advantages.
Simplifying technical explanations.
2. Technical materials (Outcome O2)
Describing technical materials.
Naming and describing the properties of technical materials.
Comparing technical materials.
Describing the application of technical materials.
3. Components and assemblies (Outcome O3)
Describing the shape and properties of components.
Describing the procedures of connecting components into assemblies or complex units.
Describing joining and fixing procedures.
Describing the position of elements in assemblies.
4. Grammar (Outcome O4)
Present, past and future tenses.
Adjectives.
Modal verbs.
Types of questions and forming questions.
Prepositions.
Relative clauses.

Required reading

- Course materials

Further reading

- Murphy, Raymond: English Grammar in Use, Cambridge University Press, Cambridge, 2004

<table>
<tr><td colspan="2">Course title</td><td colspan="6">Physical Education 1</td></tr>
<tr><td colspan="2">Course instructor(s)</td><td colspan="6">Damir Lauš, lecturer</td></tr>
<tr><td colspan="2">Programme(s) of study</td><td colspan="6">Undergraduate professional programme of study in Computer Science</td></tr>
<tr><td colspan="2">Course status</td><td colspan="6"></td></tr>
<tr><td>Year</td><td>1st</td><td>Semester</td><td>1st</td><td>ECTS</td><td colspan="3">0</td></tr>
<tr><td colspan="2" rowspan="2">Contact hours (L+PS+S)</td><td rowspan="2">0+30+0</td><td rowspan="2">L</td><td colspan="2">PS</td><td rowspan="2">S</td></tr>
<tr><td>APS</td><td>LPS</td></tr>
<tr><td colspan="3"></td><td></td><td></td><td>30</td><td></td></tr>
</table>

Course objectives

1. Acquisition of new and improving on fundamental theoretical and practical motor skills.
2. Prevention of premature reduction of skills resulting from insufficient physical activity.
3. Training students towards individual physical exercise and rational and meaningful ways of spending their free time.
4. Promotion of physical exercise and sports, and enhancing the quality of life in youth, adulthood and old age.

Expected learning outcomes

Upon completion of this course students will be able to:

I1: demonstrate mastery of practical motor skills and exercise independently,

I2: understand the importance of everyday physical exercise for the shaping of anthropological features and achieving success in one's studies and future professional life

Course content

The course is based around a set of kinesiological activities and can be divided into the main and special programme. Students opt for them based on their interests, level of proficiency in motor skills, ability levels, health status and the material conditions available.

The main programme includes kinesiological activities such as athletics, basketball, football, volleyball, dance structures, handball, table tennis…, and the special one is focused on activities less present in primary and secondary school curricula (fitness, aerobics, taekwondo, karate, squash).

Required reading

Further reading

Course title	Mathematics 2						
Course instructor(s)	Ivana Marušić, lecturer						
Programme(s) of study	Undergraduate professional programme of study in Mechatronics Undergraduate professional programme of study in Computer Science						
Course status	Compulsory						
Year	1st	Semester	2nd	ECTS	7		
Contact hours (L+PS+S)	30+45+0	L	PS		S		
			APS	LPS			
		30	45	0	0		

Course objectives

Familiarising students with the latest concepts of mathematical analysis for the purpose of upgrading the knowledge of concepts from Mathematics 1.

Expected learning outcomes

Upon completion of this course students will be able to:

- O1: apply the rules of differential calculus and analyse the obtained results,
- O2: apply the basics of differential calculus in simple problems tasks,
- O3: apply differential calculus in analysing a function graph and optimisation of the real function of a real variable,
- O4: correctly apply the basic methods for solving indefinite integrals and analyse the results of integration,
- O5: analyse the problems of calculating areas and apply integration in solving such calculations,
- O6: apply the methods of partial derivative of a function of two variables to finding the extremes of a function of two variables.

Course content

1. **Derivative** (Outcome O1, O2, O3)

Deriving function composition. Deriving an inverse function. Logarithmic differentiation. Implicit function derivation. Higher order derivatives. Differential of a function. Parametric function derivation. Continuity and derivability of functions. Equation of tangents and normals. Indefinite forms. Rise and fall of a function. Extreme points. Concavity and convexity. Inflection points. Function flow.

2. **Indefinite integral** (Outcome O4)

Definition and basic features. Table integrals. Substitution method. Partial integration method. Integration of rational functions. Integration of trigonometric functions. Integration of irrational functions.

3. **Definite integral** (Outcome O5)

Definition and features of definite integrals. Newton-Leibnitz formula. Substitution method in a definite integral. Partial integration method in a definite integral. Improper integrals. Area of a plane shape. Arc length of a plane curve. Volume of solids of rotation. Surface area of a solid of rotation.

4. **Functions of several variables** (Outcome O6)

Definition area of a function. Partial derivatives of the first order. Partial derivatives of the second order. Partial differential equations. Total differential of the first order. Local extremes of functions of two variables.

Required reading

- Tomić, Milorad: Matematika 1, Technical College in Bjelovar, Bjelovar, 2009
- Tomić, Milorad: Matematika 2, Technical College in Bjelovar, Bjelovar, 2009
- Marušić, Ivana: "Presentations for lectures and practical sessions – Mathematics 2", Bjelovar University of Applied Sciences, 2017, available on: http://vub.hr/1-godina-matematika-predavanja-vjezbe/predavanja/

Further reading

- Pavlovič Demidović, Boris, et al.: "Zadaci i riješeni primjeri iz Matematičke analize za tehničke fakultete", Golden marketing, Tehnička knjiga, Zagreb, 2003

Course title	**Application of Mathematical Software Tools**						
Course instructor(s)	Alan Mutka, PhD, lecturer						
Programme(s) of study	Undergraduate professional programme of study in Mechatronics Undergraduate professional programme of study in Computer Science						
Course status	Compulsory						
Year	1st	**Semester**	2nd	**ECTS**	2		
Contact hours (L+PS+S)	0+30+0	L	PS		S		
			APS	LPS			
		0	0	30	0		

Course objectives

Acquiring basic knowledge and skills in working with the mathematical tool Matlab and Simulink.

Expected learning outcomes

Upon completion of this course students will be able to:

- O1: initialise various types of variables in Matlab and execute basic relation and logical operations on them,
- O2: use programming loops and flow control in Matlab,
- O3: use basic mathematical functions, functions for processing sign sequences and functions for working with polynomials in Matlab,
- O4: draw a graph of a given mathematical function using graphical functions in Matlab,
- O5: use basic functions of the symbol package,
- O6: use Simulink to simulate system response.

Course content

1. Introduction to Matlab (Outcome O1)
About *Matlab*. Starting *Matlab*. Organisation of *Matlab* and data structures.
2. Variables (Outcome O1)
Internal variables. External variables. Vectors. Matrices. Complex numbers. Structures. Erasing variables.
3. Operations in Matlab (Outcome O1, O2)
Arithmetic operators. Relational operators. Logical operators. Conditional statements. Loops.
4. Functions (Outcome O3)
Elementary mathematical functions. Functions for vectors and matrices. Functions for working with polynomials. M-functions. M-scripts.
5. Graphical functions of Matlab (Outcome O4)
Functions for drawing 2D graphs. Functions for drawing 3D graphs. Functions for drawing surfaces.
6. Symbolic mathematical expressions (Outcome O5)
Basic functions of Symbolic Toolbox. Conversion of variables. Simplifying symbolic expressions. Functions for solving equations. Functions for solving equation systems. Functions for derivatives and integrals. Drawing graphs.
7. Simulink (Outcome O6)
Basic actions in Simulink. Examples of applying Simulink for the simulation of system behaviour.

Required reading

- Ban, Željko; Matuško, Jadranko; Petrović, Ivan; Primjena programskog sustava MATLAB za rješavanje tehničkih problema, Graphis, Zagreb, 2010

Further reading

- MathWorks: MATLAB ProductHelp, TheMathWorksInc., Natick, 2013

Course title	**Introduction to Computer Networks**						
Course instructor(s)	Ivan Sekovanić, lecturer						
Programme(s) of study	Undergraduate professional programme of study in Computer Science						
Course status	Compulsory						
Year	1st	**Semester**	2nd	**ECTS**	6		
Contact hours (L+PS+S)	30 + 30 + 0	L	PS		S		
			APS	LPS			
		30		30			

Course objectives

1. Understanding the functioning of computer networks and standard communication protocols.
2. Acquiring the basic knowledge of connecting computer networks.

Expected learning outcomes

Upon completion of this course students will be able to:

- O1: describe basic concepts and terms related to computer networks,
- O2: distinguish between the purposes of various network devices and explain the procedure of connecting several computers into a network,
- O3: describe the operation of a standard TCP/IP protocol,
- O4: name and describe the operation of the most important protocols of the application layer.

Course content

1. **Introduction to computer networks (O1)**
   History. Organisation of computer networks. Parameters and classification of networks according to various criteria. Network standards.
2. **Network architectures: OSI model and TCP/IP model (O1)**
   Architectures and concept of computer networks. ISO/OSI standard. TCP/IP architectures.
3. **Physical layer (O2)**
   Transmission media (wired and wireless). Signal modulation. Local area networks (LAN – Ethernet / IEEE 802.3). MAC address. Structure of the Ethernet frame. Structured cabling. Connecting local networks.
4. **Network layer (O3)**
   Network layer protocols. IP protocol (IPv4). IP address. Addressing in the network. Structure of the IP package. IP fragmentation. Network routing. IP protocol IPv6. Comparison between IPV4 and IPV6.
5. **Transmission layer (O3)**
   TCP protocol. Setting up and ending TCP connection. UDP protocol. Flow management.
6. **Application layer (O4)**
   Display of the application layer protocol: HTTP, FTP, Telnet, SMTP. DNS system.
7. **Connecting a local area network to the Internet. Access technologies (ADSL). Related protocols (NAT, DHCP) (O1)**

Required reading

1. A. Bažant et al.: Osnove arhitekture mreža, Element, Zagreb, 2014
2. Ivan Sekovanić: "Presentations for lectures and practical sessions – Introduction to Computer Networks", Bjelovar University of Applied Sciences

Further reading

1. A. S. Tanenbaum, D. J. Wetherall: Computer Networks, 5th Ed., Prentice Hall, 2011

Course title	**Introduction to Linux**						
Course instructor(s)	Tomislav Adamović, lecturer						
Programme(s) of study	Undergraduate professional programme of study in Computer Science						
Course status	Compulsory						
Year	1st	**Semester**	2nd	**ECTS**	5		
Contact hours (L+PS+S)	60	L	PS		S		
			APS	LPS			
		30		30			

Course objectives

1. Understanding the fundamentals of working in the operating system Linux.
2. Understanding the advantages of an open code system.

Expected learning outcomes

Upon completion of this course students will be able to:

- O1: install Linux on a virtual machine, explain the role and significance of the Linux operating system in the current IT system,
- O2: explain the structure and role of a directory and work with directories in shell prompt,
- O3: work with file sin Linux,
- O4: use and explain authorisation and groups in Linux,
- O5: explain the work with processes in Linux,
- O6: describe installation and configuration of basic Linux servers,
- O7: code basic shell scripts.

Course content

1. Introduction to Linux (Outcome O1)
History of Unix. Advantages of the open code system Linux. Installing Linux on a virtual machine.
2. Getting familiarised with Unix/Linux components (Outcome O2)
Kernel. System libraries. Shell. Linux commands. File System. Path. Meta signs. Rerouting. (Outcome O2)
3. Working in the operating system Linux (Outcome O3, O4, O5)
Basic Linux commands. Working with file content. Searching for files. Shell scripts. Discs and partitions. RAM. Administering the Linux system. Archiving and compressing / decompressing data. (Outcome O3)
Authorisation. (Outcome O4)
Processes. (Outcome O5)
4. Installation and configuration of a Linux server (Outcome O6)
WEB server, samba share server, svn server, ssh server, ftp server
5. Shell scripts (Outcome O7)
Fundamentals of shell scripts. Branch commands. Loops. Input parameters. Exit status. Functions. Log in.

Required reading

- Hrvoje Horvat: Uvod u Linux, Open Source Osijek, online edition (https://www.opensourceosijek.org/dokuwiki/wiki:knjige:uvod_u_linux).
- Tomislav Adamović: "Presentations for lectures and practical sessions – Introduction to Linux", Bjelovar University of Applied Sciences

Further reading

- Brian Ward: Kako radi Linux, Dobar plan, Zagreb, 2016

Course title	**Programming in C**						
Course instructor(s)	Krunoslav Husak, lecturer						
Programme(s) of study	Undergraduate professional programme of study in Computer Science						
Course status	Compulsory						
Year	1st	**Semester**	2nd	**ECTS**	6		
Contact hours (L+PS+S)	30 + 45 + 0	L	PS		S		
			APS	LPS			
		30	0	45	0		

Course objectives

Learning how to use the development environment for the development of computer programs using the structured and procedural programming language C.

Expected learning outcomes

Upon completion of this course students will be able to:

- O1: use basic and logical types of data,
- O2: apply the structures for the program flow control,
- O3: apply fields in programming solutions,
- O4: explain the execution and apply strings,
- O5: develop own functions and correctly declare arguments depending on the type of transfer,
- O6: interpret the role of a pointer in programs and use them for working with fields and functions,
- O7: interpret the structures and ways of recording data in files,
- O8: code according to a given specification in the programming language C.

Course content

1. Programming languages and programming (Outcome O1)
History of programming languages. Potential application of programming. Code development.
2. Programming language C (Outcome O1)
Writing a code in C. Development environment Visual Studio. Pre-processor commands. Input-output flow commands.
3. Types of data and arithmetic operators (Outcome O1, Outcome O8)
Types of data. Declaration of variables. Arithmetic operations. Assignment operators.
4. Logical types of data and operators (Outcome O1, Outcome O8)
Logical types of data. Comparison operators. Bitwise operators.
5. Command blocks and conditional code execution (Outcome O2, Outcome O8)
Code execution order. Command blocks. If command block. Switch-case command block.
6. Programming loops (Outcome O2, Outcome O8)
For loop. While loop. Do-while loop. Commands break and continue.
7. Fields (Outcome O3, Outcome O8)
One-dimensional fields. Two-dimensional and multi-dimensional fields.
8. Strings (Outcome O4, Outcome O8)
Characters. Strings. Functions for working with strings.
9. Functions (Outcome O5, Outcome O8)
Declaration of a function. Definition of a function. Standard headers. Header. Calling a function.
10. Pointers (Outcome O6, Outcome O8)
Pointers for working with variables, functions and fields.
11. Structures of data and files (Outcome O7, Outcome O8)
Data structures. Loading and writing of formatted and non-formatted files. Entering complex data into a file.

Required reading

- Domagoj Kusalić: Napredno programiranje i algoritmi u C-u i C++-u, 5th edition, Element, Zagreb, 2014
- Krunoslav Husak: "Presentations for lectures and practical sessions – Programming in C", Bjelovar University of Applied Sciences

Further reading

- D. M. Ritchie, B. W. Kernighan (translated by: Ante Denić): Programski jezik C, 2nd edition (https://www.scribd.com/doc/47734390/Programski-jezik-C)
- J. Šribar, B. Motik: Demistificirani C++, 3rd edition, Element, Zagreb, 2010
- Learn C programming, http://www.tutorialspoint.com/cprogramming/ (available on: 21.2.2016)
- C Tutorials, http://www.codingunit.com/category/c-tutorials (available on: 21.2.2016)

Course title	**Technical English 2**							
Course instructor(s)	Ivana Jurković, senior lecturer							
Programme(s) of study	Undergraduate professional programme of study in Computer Science							
Course status	Compulsory							
Year	1st	**Semester**	2nd	**ECTS**	2			
Contact hours (L+PS+S)	15 + 30 + 0			L	PS		S	
					APS	LPS		
				15	30	0	0	

Course objectives

Developing students' ability to use the English language related to specific technical fields.

Expected learning outcomes

Upon completion of the course students will be able to use the English language to:

- O1: describe the procedure of developing an engineering project,
- O2: describe technical problems and malfunctions as well as their causes and possible solutions,
- O3: discuss about technical requirements and describe project feasibility, improvements and redesigns,
- O4: demonstrate mastery of simple grammatical structures.

Course content

1. Developing an engineering project (Outcome O1)
Describing project design.
Defining a project, precision and tolerance.
Mathematical expressions.
Describing development procedures in an engineering project.
2. Repair and maintenance (Outcome O2)
Describing technical malfunctions.
Assessment of faults.
Describing the causes of malfunctions.
Discussing repair and maintenance.
3. Technical development (Outcome O3)
Describing technical requirements.
Suggesting ideas and solutions.
Feasibility study.
Describing improvements and redesigns.
4. Grammar (Outcome O4)
Numbers.
Conjunctions.
Pronouns.
Present, past and future tenses.

Required reading

- Course materials

Further reading

- Murphy, Raymond: English Grammar in Use, Cambridge University Press, Cambridge, 2004

Course title	**Physical Education 2**						
Course instructor(s)	Damir Lauš, lecturer						
Programme(s) of study	Undergraduate professional programme of study in Computer Science						
Course status							
Year	1st	**Semester**	2nd	**ECTS**	0		
Contact hours (L+PS+S)	0+30+0	L	PS		S		
			APS	LPS			
				30			

Course objectives

1. Acquisition of new and improvement of fundamental theoretical and practical motor skills.
2. Prevention of premature reduction of skills resulting from insufficient physical activity,
3. Training students towards individual physical exercise and rational and meaningful ways of spending their free time,
4. Promotion of physical exercise and sports, and enhancing the quality of life in youth, adulthood and old age.

Expected learning outcomes

Upon completion of this course students will be able to:

I1: demonstrate mastery of practical motor skills and exercise independently,

I2: understand the importance of everyday physical exercise for the shaping of anthropological features and achieving success in one's studies and future professional life.

Course content

The course is based around a set of kinesiological activities and can be divided into the main and special programme. Students opt for these based on their interests, level of proficiency in motor skills, ability levels, health status and the material conditions available.

The main programme includes kinesiological activities such as athletics, basketball, football, volleyball, dance structures, handball, table tennis…, and the special one is focused on activities less present in primary and secondary school curricula (fitness, aerobics, taekwondo, karate, squash).

Required reading

Further reading

Course title	**Digital Techniques**						
Course instructor(s)	Dario Vidić, lecturer						
Programme(s) of study	Undergraduate professional programme of study in Mechatronics Undergraduate professional programme of study in Computer Science						
Course status	Compulsory						
Year	2nd	**Semester**	3rd	**ECTS**	6		
Contact hours (L+PS+S)	30 + 30 + 0	L	PS		S		
			APS	LPS			
		30	16	14			

Course objectives

To acquire basic knowledge of digital techniques.

Expected learning outcomes

Upon completion of this course students will be able to:

O1: use different number systems and codes,
O2: design units for detecting and correcting data transfer errors,
O3: minimise and realise complex logical functions using basic logic circuits,
O4: explain the operation of combinational and arithmetic units,
O5: explain the operation of basic types of bistables and counters,
O6: design sequential circuits,
O7: explain the operation of AD and DA conversion circuits.

Course content

1. Number systems and codes (Outcome O1, O2)
Number systems (decimal, binary, hexadecimal etc.). (Outcome O1)
Conversion of numbers between number systems. (Outcome O1)
Operations with binary numbers. (Outcome O1)
Characteristic binary codes. (Outcome O1)
Binary encoding. (Outcome O2)
2. Logic circuits (Outcome O3)
Propositional logic. The basic principles of Boolean algebra. AND, OR, NOT, NAND, NOR gates. Complex logical operations. Minterms and maxterms. Methods of minimisation (K-map, the Quine-McCluskey method).
Design of logic circuits in semiconductor technology: TTL technology. CMOS technology.
3. Complex combinational modules (Outcome O4)
Adders. Digital comparator. Parity circuit. Encoder and decoder. Multiplexer and demultiplexer.
4. Bistables (Outcome O5)
Operation and types of bistables. Bistable design using logic integrated circuits.
5. Registers and sequential circuits (Outcome O5 and O6)
Design and implementation of registers. Design of counters. Asynchronous and synchronous counters. Decade counters. Sequential machines.
6. D/A and A/D conversion (Outcome O7)

Required reading

- Vrhovski, Zoran; Šumiga Ivan: Digitalna tehnika – problem book, Technical College in Bjelovar, Bjelovar, 2015
- Dario Vidić: "Presentations for lectures and practical sessions – Digital Techniques", Bjelovar University of Applied Sciences.

Further reading

- Peruško, Uroš: Digitalna elektronika, Školska knjiga, Zagreb, 1996
- Čupić, Marko: Digitalna elektronika i digitalna logika, problem book, Kigen d.o.o., Zagreb, 2006

Course title	**Web Programming 1**							
Course instructor(s)	Tomislav Adamović, lecturer							
Programme(s) of study	Undergraduate professional programme of study in Computer Science							
Course status	Compulsory							
Year	2nd	**Semester**	3rd	**ECTS**	5			
Contact hours (L+PS+S)	60	L	PS		S			
			APS	LPS				
		30		30				

Course objectives

1. Familiarising students with the fundamentals of front-end web programming.
2. Mastering the webpage programming tools: HTML, CSS, *JavaScript* and *jQuery*, Ajax and React.

Expected learning outcomes

Upon completion of this course students will be able to:

- O1: Explan the client-server concept and fundamental communication protocols
- O2: Create a static HTML document, use HTML5 for animation effects
- O3: Use CSS. Create a separate CSS file. Use CSS framework
- O4: Program in *JavaScript* and inclusion in a static HTML document
- O5: Program in *jQuery*, use asynchronous calls, Ajax
- O6: Include React in an existing webpage

Course content

1. Introduction to web programming (Outcome O1)
Fundamental concepts. Communication protocols. The client-server model. The first web page.
2. HTML (Outcome O2)
Statement syntax. Language elements. Lists. Charts. Styles. Links. Working with images. Tables. Forms. HTML5.
3. CSS (Outcome O3)
Introduction to CSS. Structure and syntax of CSS. Selectors and declarations. Working with style classes. Style inheritance. Object management using CSS. Implementation of CSS.
4. *JavaScript* (Outcome O4)
Inclusion in a HTML document. Variables. Operators. Loops. Flow control. Arrays. Functions. Events. Errors.
5. Introduction to *jQuery* and Ajax (Outcome O5)
Inclusion in a HTML document. Syntax. Selectors. Events. Effects. Animations. Asynchronous calls, integration into an existing webpage.
6. Introduction to React (Outcome O6)
Concepts and tools of React. Integration of React into an existing webpage.

Required reading

- Tomislav Adamović: "Presentations for lectures and practical sessions – Web Programming 1", Bjelovar University of Applied Sciences
- http://www.w3schools.com (online)

Further reading

- Shelley Powers: Naučite JavaScript, Dobar plan, Zagreb, 2010
- Gilberto Crespo, Responzivni Web dizajn uz jQuery, Zagreb, Dobar plan, 2015
- Karol Krol, WordPress kompletan priručnik, Zagreb, Dobar plan, 2017
- Peter Gasston, Knjiga za CSS3, Zagreb : Dobar plan, 2013
- Mark Pilgrim, HTML5, Zagreb, Dobar plan, 2010

Course title	**Object-Oriented Programming**						
Course instructor(s)	Alan Mutka, PhD, lecturer						
Programme(s) of study	Undergraduate professional programme of study in Computer Science						
Course status	Compulsory						
Year	2nd	**Semester**	3rd	**ECTS**	6		
Contact hours (L+PS+S)	30+30+0	L	PS		S		
			APS	LPS			
		30		30			

Course objectives

To learn how to use C++ for creating object-oriented applications.

Expected learning outcomes

Upon completion of this course students will be able to:

- O1: Use fundamental language elements in creating a program in the programming language C++
- O2: Correctly define classes and objects
- O3: Correctly define inheritance and polymorphism
- O4: Correctly define templates
- O5: Use a C++ standard library for program development
- O6: Use the object-oriented programming language C++ and its libraries for developing a program according to a given specification

Course content

1. Object-oriented programming languages (Outcome O1)
An overview of programming techniques. An introduction to object-oriented programming.
2. Elements of C++ (Outcome O1, O6)
Variables and constants. Expressions and statements. Functions. Loops. Flow control. Pointers. References. Input and output streams.
3. The basics of classes (Outcome O2, Outcome O6)
Class. Class members. Accessing class members. Visibility of data in a class. Friend class. The *this* keyword. Constructors and destructors. Creating and deleting of objects. The *Const* functions. *Volatile*. Static class members. Pointers and classes. Operators.
4. Inheritance (Outcome O3, Outcome O6)
Hierarchical inheritance. The constructor and destructor. Hiding of functions in base class. Virtual functions.
5. Polymorphism (Outcome O3, Outcome O6)
Virtual inheritance. Abstract data types. Advanced inheritance.
6. Templates (Outcome O4**,** Outcome O6**)**
Function templates. Class templates.
7. Standard libraries (Outcome O5**,** Outcome O6**)**

Required reading

- Julijan Šribar, Boris Motik: Demistificirani C++, 4th edition, Element, Zagreb, 2014
- Alan Mutka: "Presentations for lectures and practical sessions – Object-Oriented Programming", Bjelovar University of Applied Sciences.

Further reading

- Domagoj Kusalić: Napredno programiranje i algoritmi u C-u i C++-u, 5th edition, Element, Zagreb, 2014

Course title	**Data Structures and Algorithms**					
Course instructor(s)	Krunoslav Husak, lecturer					
Programme(s) of study	Undergraduate professional programme of study in Computer Science					
Course status	Compulsory					
Year	2nd	**Semester**	3rd	**ECTS**	5	
Contact hours (L+PS+S)	30 + 30 + 0	L	PS		S	
			APS	LPS		
		30	0	30	0	

Course objectives

To learn the fundamental characteristics of standard data structures (list, stack, queue, binary tree) and algorithms for working with these data structures in the modern, procedural programming language C++.

Expected learning outcomes

Upon completion of this course students will be able to:

- O1: explain the complexity of operations and algorithms,
- O2: use recursive algorithms,
- O3: create solutions based on simple data structures (list, stack, queue),
- O4: create solutions based on complex data structures (tree, heap, queue, dictionary),
- O5: describe and use sorting, search and compression algorithms,
- O6: use general techniques for algorithm construction,
- O7: program in the programming language C++ according to the given specification.

Course content

1. Introduction, analysis and complexity of algorithms (Outcome O1)
Fundamental concepts. Structured and non-structured data. Algorithm definitions, history, conventions and writing algorithms. Analysing algorithms.
2. Recursion (Outcome O2, Outcome O7)
Term, analysis of well-known recursive algorithms, various recursive procedures.
3. Data structures (Outcome O3, Outcome O7)
Basic terms of data structures, lists, data structure – classes, functions for working with objects, class example analysis.
4. Simple data structures (Outcome O3, Outcome O7)
List and general lists. Stack. Data input and output, searching, printing.
5. Complex data structures (Outcome O4, Outcome O7)
Queue. Binary tree and binary search tree. K-tree. Set. Dictionary. Priority queue. Copying. Heap. Data input and output, searching, printing.
6. Implementation of the aforementioned structures in complex algorithms (Outcome O5, Outcome O7)
Data array sorting, searching and compression.
7. General algorithm design techniques (Outcome O6, Outcome O7)
“Divide and conquer”, dynamic programming, the “greedy method”, backtracking.

Required reading

- Domagoj Kusalić: Napredno programiranje i algoritmi u C-u i C++-u, 5th edition, Element, Zagreb, 2014
- Krunoslav Husak: "Presentations for lectures and practical sessions – Data Structures and Algorithms", Bjelovar University of Applied Sciences.

Further reading

- Robert Manger: Strukture podataka i algoritmi, Element, Zagreb, 2014

Course title	**Database Systems**							
Course instructor(s)	Tomislav Adamović, lecturer							
Programme(s) of study	Undergraduate professional programme of study in Computer Science							
Course status	Compulsory							
Year	2nd	**Semester**	3rd	**ECTS**	6			
Contact hours (L+PS+S)	60	L	PS		S			
			APS	LPS				
		30		30				

Course objectives

1. To understand database management systems.
2. To understand the design of relational databases, the entity-relationship model and relational algebra.
3. To learn the SQL query language.

Expected learning outcomes

Upon completion of this course students will be able to:

- O1: create and change tables in the relational data model
- O2: select data from tables
- O3: write CRUD commands for changing data in tables
- O4: explain basic types of parameters and objects in PL/SQL
- O5: create and test the RESTfull service on a database
- O6: work with fundamental commands and programming concepts in PL/SQL
- O7: work with exceptions, debug a code and perform unit tests

Course content

1. Introduction to databases (Outcome O1)
Defining the fundamental concepts. Database architecture. Database management systems.
2. Relational data model (Outcome O1, O2)
Relational schema. Operations in the relational data model. Relational algebra. (Outcome O1)
Introduction to SQL. (Outcome O2)
3. Data selection (Outcome O2)
Simple queries. Expression. Defining conditions. Aggregate functions. Joined relations. SQL statements containing conditions with subqueries. Grouping results. Specifying conditions for a group of records. Order of results. Storing results in a temporary relation. Determining union of relations using the *select* statement.
4. Changing data (Outcome O3)
Ways of using the *INSERT* statement. *INSERT* statement. *DELETE* statement. *UPDATE* statement.
5. Fundamentals of SQL language use. (Outcome O4, O5, O6, O7)
Introduction. Fundamental objects in SQL language. Types of data. (Outcome O4)
Basic SQL statements for data management. (Outcome O5).
Statement format. Data and relation definition statements. Statements for transfer of data to and from an operating system file. (Outcome O6)
Exceptions, debugging and unit tests (Outcome O7)

Required reading

- Tomislav Adamović: "Presentations for lectures and practical sessions – Database Systems", Bjelovar University of Applied Sciences

Further reading

- C. J. Date: An Introduction to Database Systems, 8th edition, Addison Wesley, Boston, 2006
- J. D. Ullman, J. Widom: A First Course in Database Systems, Prentice-Hall, 2008
- Robert Manger: Baze podataka, Zagreb, Element, 2014

Course title	Technical English 3						
Course instructor(s)	Ivana Jurković, senior lecturer						
Programme(s) of study	Undergraduate professional programme of study in Computer Science						
Course status	Compulsory						
Year	2nd	Semester	3rd	ECTS	2		
Contact hours (L+PS+S)	15 + 30 + 0	L	PS		S		
			APS	LPS			
		15	30	0	0		

Course objectives

To develop students' ability to use the English language related to specific technical fields.

Expected learning outcomes

Upon completion of the course students will be able to use the English language to:

- O1: describe regulations, standards procedures and measures related to occupational health and safety
- O2: describe automated systems, measurable parameters, readings and approximate values
- O3: explain testing procedures, conduction of experiments and describe the predicted outcomes of testing
- O4: demonstrate mastery of more complex grammatical structures

Course content

1. Occupational health and safety (Outcome O1)
Description of occupational health and safety measures.
Protective equipment.
Regulations and standards.
Written instructions.
2. Automatic control (Outcome O2)
Describing automated systems.
Describing measurable parameters.
Giving approximate values.
Describing graphs.
3. Tests and experiments (Outcome O3)
Describing tests and experiments.
Comparison of results and expectations.
Assumptions.
4. Grammar (Outcome O4)
Imperative.
Conditional clauses.
Passive Voice.
Nouns.
Compounds.
Articles.

Required reading

- Course materials

Further reading

- Murphy, Raymond: English Grammar in Use, Cambridge University Press, Cambridge, 2004

Course title	Microcomputers							
Course instructor(s)	Zoran Vrhovski, senior lecturer							
Programme(s) of study	Undergraduate professional programme of study in Mechatronics Undergraduate professional programme of study in Computer Science							
Course status	Compulsory							
Year	2nd	Semester	4th	ECTS	4			
Contact hours (L+PS+S)	15 + 30 + 0	L	PS		S			
			APS	LPS				
		15		30				

Course objectives

To familiarise students with the application of microcomputers and their programming and usage in the design and production of more complex electronic devices.

Expected learning outcomes

Upon completion of this course students will be able to:

- O1: choose a microcomputer that is optimal for a given purpose from the aspect of price, features and availability
- O2: configure microcomputer operation using registers in the development software environment
- O3: create a microcontroller control program for a given purpose in the development software environment
- O4: use microcontroller interrupts when it is required by the functionality of an electronic device koristiti
- O5: connect electronic devices to a microcontroller taking into account the purpose of an individual pin of a microcontroller

Course content

1. Fundamental concepts and microcomputer architecture (Outcome O1)
Application of microcomputers. Historical development of microcomputers. Differences between microcomputers, microcontrollers and microprocessors. Microcomputer structure: CPU, buses. Microcomputer architecture: (CISC, RISC). Execution of microcomputer instructions.
2. Microcontrollers (Outcome O1, O2, O4)
Features. Architecture. CPU. Clock rate. Instruction execution. Set of instructions. Memory. Input/output registers. Reset. Watchdog. Interrupts. Digital inputs and outputs. Counters and timers. PWM. Digital-to-analog converter. Analog-to-digital converter and analog inputs. USART (serial communication). I2C communication. SPI communication. External interrupts. Power supply. Producers of microcontrollers. Features of the Atmel ATMega16 microcontroller.
3. Programming a microcontroller (Outcome O2, O3, O4)
Programming a microcontroller. Instruction set of the Atmel AVR microcontroller family. Machine code. Main program and infinite loops. Interrupt routines. Functions. Microcontroller programming environments. Fuse bits. Lock bits. In-System Programming.
4. Connecting electronic devices to a microcontroller (Outcome O4, O5)
Connecting the following devices to a microcontroller: push buttons, LED diodes, LCD display, potentiometer, NTC resistor, numeric display, optocoupler, transistor as a switch, relay, buzzer, analog and digital temperature sensors, Bluetooth module, graphic display, GSM module, matrix keyboard, servo motor, ultrasound sensor, real time clock modula, H bridge, communication modules. Microcomputer system management using a computer or smartphone application.

Required reading

- Vrhovski, Zoran; Miletić, Marko: Mikroračunala - Programiranje mikrokontrolera porodice Atmel u programskom okruženju Atmel Studio 6, Technical College in Bjelovar, Bjelovar, 2014
- Vrhovski, Zoran: Presentations of lectures in Microcomputors, Bjelovar University of Applied Sciences, available on: https://vub.hr/mikroracunala/
- ATMEL: 8-bit AVR Microcontroller with 16K Bytes In-System Programmable Flash – ATMEGA16,

http://ww1.microchip.com/downloads/en/DeviceDoc/doc2466.pdf (available on: 19.7.2018)

Further reading

- F. Barrett, Steven; Pack, Daniel; Thornton, Mitchell: Atmel AVR microcontroller primer: programming and interfacing, Morgan &Claypool Publishers, Thornton, 2007

Course title	**Software Engineering**						
Course instructor(s)	Tomislav Adamović, lecturer						
Programme(s) of study	Undergraduate professional programme of study in Computer Science						
Course status	Compulsory						
Year	2nd	**Semester**	4th	**ECTS**	4		
Contact hours (L+PS+S)	45	L	PS		S		
			APS	LPS			
		30	15				

Course objectives

1. Understanding steps in software development.
2. Acquisition of basic knowledge on a systematic, disciplined and measurable approach to software development, implementation and maintenance.

Expected learning outcomes

Upon completion of this course students will be able to:

- O1: describe software life cycle elements and models
- O2: draw diagrams with process elements in software development
- O3: describe best practices in project management and planning
- O4: assess the complexity and quality of a program code, programmer's productivity and risks
- O5: analyse various user requirement models
- O6: design software based on requirements
- O7: describe the principles and tools used for program testing
- O8: apply the basic concepts of object-oriented development

Course content

1. Introduction to software engineering (Outcome O1)
Software engineering. Organisation in IT. Software life cycle. Software life cycle models.

2. Software processes and other models (Outcome O2)
Software process model. Data flow diagram. Petri net. Object model. Sequence diagram. Hierarchy diagram. Control flow graph. State diagram. UML diagrams.

3. Software project planning and management (Outcome O3)
Approaches to management. Team approach. Critical practice. Maturity model. Personal software process. Earned value analysis. Defect tracking. Post-mortem analysis. Project planning. Assigning responsibilities. The program evaluation and review technique. Software cost estimation.

4. Software measurement and risks (Outcome O4)
Measurement. Software measurement theory. Software metrics. Process metrics. Risk assessment. Risk exposure. Risk reduction. Risk management plan. Formal inspection and technical inspection. Software reliability. IEEE standards for software quality assurance.

5. Requirements. (Outcome O5)
Data flow modelling. Behaviour modelling. Dictionary. System diagram. IEEE requirements specification standard.

6. Software design (Outcome O6)
Stages in the design process. Design concepts. Measuring cohesion. Measuring coupling. Requirements traceability.

7. Software testing (Outcome O7)
Fundamentals of software testing. Test coverage criteria. Data flow testing. Random testing. Boundary testing.

8. Object-oriented development, metrics and testing (Outcome O8)
Inheritance and polymorphism. Object identification. Association identification. Identification of multiplicity. Object-oriented design metrics. MOOD metrics. MM testing. Function pair coverage.

Required reading

- Tomislav Adamović: "Presentations for lectures and practical sessions – Software Engineering", Bjelovar University of Applied Sciences
- David A. Gustafson: Schaum's Outlines of Software Engineering, McGraw-Hill, 2002

Further reading

- Roger S. Pressman: Software Engineering: A Practitioner's approach, McGraw-Hill, 2014
- Ian Sommerville: Software Engineering, Addison-Wesley, 2011

Course title	**C# Programming**							
Course instructor(s)	Krunoslav Husak, lecturer							
Programme(s) of study	Undergraduate professional programme of study in Computer Science							
Course status	Compulsory							
Year	2nd	**Semester**	4th	**ECTS**	5			
Contact hours (L+PS+S)	30 + 30 + 0			L	PS		S	
					APS	LPS		
				30	0	30	0	

Course objectives

To acquire knowledge and skills required for efficient usage of the contemporary object-oriented programming language C#.

Expected learning outcomes

Upon completion of this course students will be able to:

- O1: recognise and use the syntax of the programming language C#,
- O2: use and apply the basic concepts of the object-oriented paradigm in C#,
- O3: use and apply advanced concepts of the object-oriented paradigm and collections in C#,
- O4: apply multithreading for the purpose of developing responsive C# computer programs,
- O5: develop and design programs with a graphical user interface,
- O6: develop and design programs for communicating with the external environment.

Course content

1. Uvod (Outcome O1)
Programming language C#. Introduction to .NET, CLR and C#. Programming tools. The *Visual Studio* runtime environment. Working in the command line.
2. Fundamentals of C# (Outcome O1)
Program syntax and structure. Value types and reference types. Variables. Operators. Loops and flow control. Sequences. String and Char types. Formatting data for display. Working with a text. Character encoding. Commenting and code readability.
3. Abstraction using classes (Outcome O2)
Abstraction and encapsulation. Defining class and objects. Constructors. Defining methods. Visibility modifiers. Inheritance. Polymorphism. Abstract classes. Interface.
4. Advanced concepts of the object-oriented paradigm, exceptions and collections (Outcome O3)
Generics. Inner and anonymous classes. Delegates and events. Exceptions and processing of exceptions. Structures. List<T>. Collections and polymorphism. Sets, lists and maps.
5. Multithreading (Outcome O4)
Multithreading and multithreaded applications. Fundamentals of working with multithreaded applications.
6. Creating applications with a graphical user interface (Outcome O5)
Creating applications with a graphical user interface using Windows Forms. Basic concepts. Windows. Working with controls. Visual arrangement of controls. Events. Event processing. Support to controls. Simple graphical controls. Menus. User interface and multithreading. Creating a simple WPF user interface.
7. Files and data flows (Outcome O6)
Working with directories and files. Writing and reading text files. Data flows.
8. Communication with external environment (Outcome O6)
Fundamentals of XML and JSON. Creating, searching and serialisation of XML and JSON. Various ways of working with networks. Network protocols. Operation of the server-client application. Basic concepts of working with a database. Queries. Object context.

Required reading

- Krunoslav Husak: "Presentations for lectures and practical sessions – C# Programming", Bjelovar University of Applied Sciences
- Ian Griffiths, Matthew Adams, Jesse Liberty: Programiranje C# 4.0, O'Reilly Media, 2011

Further reading

Course title	**Fundamentals of Programming in JAVA**						
Course instructor(s)	Alan Mutka, PhD, lecturer						
Programme(s) of study	Undergraduate professional programme of study in Computer Science						
Course status	Compulsory						
Year	2nd	**Semester**	4th	**ECTS**	5		
Contact hours (L+PS+S)	30+30+0	L	PS		S		
			APS	LPS			
		30		30			

Course objectives

To master the usage of the programming language Java in creating object-oriented applications.

Expected learning outcomes

Upon completion of this course students will be able to:

- O1: Design a solution to a given problem according to a given specification and implement it using objects and classes
- O2: Apply the inheritance principle in the programming language Java
- O3: Apply error management in the programming language Java
- O4: Apply interfaces and abstract classes in the programming language Java
- O5: Apply collections and generic structures in the programming language Java
- O6: Apply input and output streams in Java
- O7: Apply Swing classes in creating GUI applications
- O8: Design a multithreaded solution to a given problem according to the specifications

Course content

1. Fundamental structures of the Java programming language (Outcome O1)
Historical development. Development of Java programming. Integrated development environment. Creating the *HelloWorld* project. Variables and expressions. Loops and flow control. Functions. Operators. Conversions. One-dimensional arrays. Multidimensional arrays. Operations on arrays. Methods for working with sequences of characters.
2. Object-oriented programming in Java (Outcome O1, O2, O3, O4, O5, O6)
Classes and objects. Class members. Access to class members. Visibility of data in a class. Constructors. Destructors. Inheritance. Polymorphism. Error management. Interface and abstract classes. Collections and generic structures. Input and output streams.
3. Advanced programming in Java (Outcome O7, O8)
Creating GUI applications. Multithreaded programming.

Required reading

- Alan Mutka: "Presentations for lectures and practical sessions – Fundamentals of Programming in JAVA", Bjelovar University of Applied Sciences
- Yakov Fain: Programiranje Java, Kompjuter biblioteka, Zagreb, 2015

Dopunska literatura

- Big Java: Compatible with Java 5, 6 and 7, author: Cay S. Horstmann

Course title	**Web Programming 2**							
Course instructor(s)	Tomislav Adamović, lecturer							
Programme(s) of study	Undergraduate professional programme of study in Computer Science							
Course status	Compulsory							
Year	2nd	**Semester**	4th	**ECTS**	6			
Contact hours (L+PS+S)	60	L	PS		S			
			APS	LPS				
		30		30				

Course objectives

1. Mastering the PHP scripting language for web programming.
2. Mastering the use of databases within a web page.
3. Familiarising students with the Node.js technology.

Expected learning outcomes

Upon completion of this course students will be able to:

- O1: explain the concepts of backend programming and PHP settings on a server
- O2: use PHP programming elements for generating a HTML page
- O3: use PHP in object-oriented programming
- O4: use the MySQL base
- O5: integrate PHP and MySQL
- O6: explain the Node.js concept and create a web application in Node.js

Course content

1. Introduction to PHP (Outcome O1)
2. Language fundamentals. (Outcome O2)
Lexical structure. Types of data. Variables. Expressions and operators. Flow control statements. Building PHP in web pages.
3. Use of the PHP language (Outcome O2)
Functions. Character sequences. Arrays. Objects. Web techniques. PDF extensions. Security. PHP on different platforms. Web services. Error handling.
4. PHP object-oriented programming (Outcome O3)
Classes, methods and features, Instance, inheritance.
5. Databases (Outcome O4, O5)
Relational databases and SQL. MySQL and object interface. (Outcome O4)
Integration of PHP and MySQL (Outcome O5)
6. Node.js (Outcome O6)
Introduction to Node.js. Asynchronous calls and callbacks. NPM (node package manager). Node.js framework express, socket.io Node.js and databases. Exceptions, debugging and testing.

Required reading

- Tomislav Adamović: "Presentations for lectures and practical sessions – Web Programming 2", Bjelovar University of Applied Sciences
- http://www.w3schools.com (online)

Further reading

- Kevin Tatroe: Programiranje PHP, 3rd edition, Dobar plan, Zagreb, 2015

Course title	**Operating Systems**					
Course instructor(s)	Krunoslav Husak, lecturer					
Programme(s) of study	Undergraduate professional programme of study in Computer Science					
Course status	Compulsory					
Year	2nd	**Semester**	4th	**ECTS**	4	
Contact hours (L+PS+S)	30 + 15 + 0	L	PS		S	
			APS	LPS		
		30	0	15	0	

Course objectives

To learn the main characteristics of contemporary operating systems. To acquire knowledge about processes and threads as well as ways of communication between them. To learn the basic characteristics of a file subsystem. To learn the basic security characteristics of operating systems as well as multiprocessor and embedded systems.

Expected learning outcomes

Upon completion of this course students will be able to:

- O1: interpret a simple computer model,
- O2: analyse the interrupt system operation on a computer,
- O3: explain the processes and inter-process communication,
- O4: explain threads and inter-thread communication,
- O5: explain memory management principles,
- O6: interpret a file subsystem and analyse multiple disk redundant containers,
- O7: analyse operating system security management techniques,
- O8: analyse the main features of multiprocessor and embedded systems,
- O9: develop computer programs and solutions that shall be accomplished using operating system functions.

Course content

1. Introduction (Outcome O1)
History of operating systems. Tasks of operating systems. Structure of operating systems.
2. Simple computer model (Outcome O1, Outcome O9)
The Von Neumann computer model. Description of computer components. Thread.
3. Performing input/output operations, interrupt (Outcome O2, Outcome O9)
Connecting devices to a computer. Busy waiting and processor interrupts. Direct memory access.
4. Processes (Outcome O3, Outcome O9)
Basic terms. Programs and processes. Multitasking. Internal process structure. Initiating and tracking processes. Inter-process communication.
5. Kernel of an operating system, threads and inter-thread communication (Outcome O4, Outcome O9)
Kernel data structure. Kernel functions. Threads. Initialising and the states of threads. Multithreaded task accomplishment. Multithreading model. Mutual exclusion. Mutual exclusion hardware. The producer/consumer problem. Deadlock. The concept of monitor. Dynamic behaviour of an operating system. Basic ways of assigning processors to threads.
6. Memory management (Outcome O5, Outcome O9)
Static and dynamic memory management. Paging.
7. File subsystem and multiple disk redundant containers (Outcome O6, Outcome O9)
Disks. File systems. The role of data buffer. Designing redundancy systems. System dependability and undependability. Ways of redundant disk organisation.
8. Security of operating systems (Outcome O7, Outcome O9)
Basic terms. Threats. Data protection. Authentication. Authorisation. Malware. Cryptography.
9. Multiprocessor and embedded systems, virtualisation (Outcome O8, Outcome O9)
Basic terms. Multiprocessor systems. Kernel in a multiprocessor system. Embedded systems. The main features of embedded systems. The application of embedded systems. Virtualisation. Application of virtualisation. Ways of realisation.

Required reading

- Krunoslav Husak: "Presentations for lectures and practical sessions – Operating Systems", Bjelovar University of Applied Sciences
- Leo Budin, Marin Golub, Domagoj Jakobović, Leonardo Jelenković: Operacijski sustavi, Element, Zagreb, 2010

Further reading

- A. Silberschatz, P. B. Galvin, G. Gagne: Operating System Concepts, John Wiley & Sons, 2003

Course title	**Technical English 4**							
Course instructor(s)	Ivana Jurković, senior lecturer							
Programme(s) of study	Undergraduate professional programme of study in Computer Science							
Course status	Compulsory							
Year	2nd	**Semester**	4th	**ECTS**	2			
Contact hours (L+PS+S)	15 + 30 + 0			L	PS		S	
					APS	LPS		
				15	30	0	0	

Course objectives

To develop students' ability to use the English language related to specific technical fields.

Expected learning outcomes

Upon completion of the course students will be able to use the English language to:

- O1: describe examples of advanced technologies and innovative solutions,
- O2: communicate in via e-mail,
- O3: draft a curriculum vitae and job application in standard international formats,
- O4: prepare and give a 10-minute presentation in English on one of the given topics in the technical area.

Course content

1. Advanced technologies (Outcome 1)
Renewable energy sources. Describing possibility and restrictions. Advanced technology and innovations.
2. Writing formal and informal e-mails (Outcome 2)
Levels of formality. Enquiries. Offers. Exchanging information. Deadlines. Confirming deadlines and arrangements.
3. Curriculum vitae and job application (Outcome 3)
Writing a curriculum vitae and job application in the English language.
4. Presentation skills in the English language (Outcome 4)
Preparing a structured presentation in English on a given topic related to the technical field.

Required reading

- Course materials

Further reading

Course title	**.NET Programming**						
Course instructor(s)	Krunoslav Husak, lecturer						
Programme(s) of study	Undergraduate professional programme of study in Computer Science						
Course status	Compulsory						
Year	3rd	**Semester**	5th	**ECTS**	6		
Contact hours (L+PS+S)	30 + 30 + 0	L	PS		S		
			APS	LPS			
		30	0	30	0		

Course objectives

To acquire knowledge and skills required for efficient development of complex web applications with access to databases using C# and the .NET platform.
To acquire knowledge and skills required for creating mobile applications using the .NET platform.

Expected learning outcomes

Upon completion of this course students will be able to:

O1: create a simple web application using Web Forms,
O2: create a simple web application using Core Pages,
O3: create a simple web application using Core MVC,
O4: create a simple web application using Xamarin.

Course content

1. Introduction (Outcome O1, Outcome O2, Outcome O3, Outcome O4)
.NET framework components. Web application components. Processing static and dynamic web pages. Various ASP.NET components for web application development. ASP.NET and ASP.NET Core. Development of applications for mobile devices using Xamarin.

2. ASP.NET Web Forms (Outcome O1)
Fundamentals of Web Forms. Development of a simple web application using Web Forms. Entering HTML and CSS into Web Forms. Adding and managing the controls. Data validation. Session control and web application status control. Master pages. Connection with a database.

3. ASP.NET Core Pages (Outcome O2)
Dynamic compiling. Development of a simple Core Pages application using the Razor syntax. Project structure. Editing of the design of a web application. Data entry and presentation. Storing data in a database.

4. ASP.NET Core MVC (Outcome O3)
MVC project structure. Application settings. Development of models and repositories. Controller development. View development and editing. Entity Framework. Development and initialisation of the application daatabse. Authentication and authorisation.

5. Xamarin (Outcome O4)
Basic terms. Development of a simple Xamarin.Forms application. Simple data entry. Models and event management. MVVM. Design and development of an Android application. Storing data using SQLite.

Required reading

- Krunoslav Husak: "Presentations for lectures and practical sessions – .NET Programming", Bjelovar University of Applied Sciences
- Ian Griffiths, Matthew Adams, Jesse Liberty: Programiranje C# 4.0, O'Reilly Media, 2011
- Jess Chadwick, Todd Snyder, Hrusikesh Panda: Programiranje ASP.NET MVC 4, O'Reilly Media, 2013

Further reading

Course title	**Development of Computer Games**					
Course instructor(s)	Alan Mutka, PhD, lecturer					
Programme(s) of study	Undergraduate professional programme of study in Computer Science					
Course status	Compulsory					
Year	3rd	**Semester**	5th	**ECTS**	6	
Contact hours (L+PS+S)	30+30+0	L	PS		S	
			APS	LPS		
		30		30		

Course objectives

To familiarise students with the development of computer games.

Expected learning outcomes

Upon completion of this course students will be able to:

O1: describe the basic elements involved in a video game,
O2: explain and create a video game design document,
O3: develop a system for controlling the main objects on the scene,
O4: develop a functional user interface of a video game,
O5: develop systems for the management of graphics, animations and sounds in a video game.

Course content

1. **Introduction** (Outcome 1)
   Basic terms. Tools. Linear algebra. Video game elements.
2. **Video game design** (Outcome 2)
   Software framework for game development. Video game architecture. Design patterns. Data structures. System of components. Game outcomes.
3. **Video game components** (Outcome 3, Outcome 4)
   Video game object. Camera. Scene. User interface. Layers. Levels.
4. **Resources** (Outcome 5)
   Animations. Graphics. Sounds. Templates. Scripts.

Required reading

- Jeremy Gibson Bond: Introduction to Game Design, Prototyping, and Development: From Concept to Playable Game with Unity and C# , 2nd edition, Pearsone Education, New Jersey
- Alan Mutka: "Presentations for lectures and practical sessions – C# Programming", Bjelovar University of Applied Sciences

Further reading

Course title	**Mobile Application Development**						
Course instructor(s)	Alan Mutka, PhD, lecturer						
Programme(s) of study	Undergraduate professional programme of study in Computer Science						
Course status	Compulsory						
Year	3rd	**Semester**	5th	**ECTS**	6		
Contact hours (L+PS+S)	30+30+0	L	PS		S		
			APS	LPS			
		30	0	30	0		

Course objectives

1. Familiarising students with the advantages of developing programming solutions for mobile devices.
2. Familiarising students with tools for the development of Android mobile applications.
3. Mastering the methods of designing Android user interfaces.

Expected learning outcomes

Upon completion of this course students will be able to:

- O1: describe platform architecture and use basic development components for the implementation of the development component for mobile platform solutions,
- O2: develop a basic application using basic elements for the development of a mobile application: dialogues, menus and settings,
- O3: develop and use databases and allow access to the application data,
- O4: use location, telephone, SMS, e-mail and web services.

Course content

1. **Introduction** (Outcome 1)
   Introduction to Android. Historical perspective. Introduction to the Android mobile platform.
2. **Runtime environment** (Outcome 1)
   Installation and familiarisation with the Android Studio development environment. Installation of the Android SDK package. Creating a virtual Android device. Use of emulators and integrated tools. Creating the first Android application. Installation and setup of the application on a mobile device.
3. **Development of an Android mobile application** (Outcome 2)
   Java programming language. Components of the Android operating system. Basic elements. Application life cycle. Creating a new project and window. Designing a simple graphical interface (visual, XML). Basic standard elements of a user interface. Events and actions. Screen orientation change. Layout of elements on the interface. Logs and messages. Creating the menu. Use of application resources. Connecting and activating other windows *(Activity)*.
4. **Data storage** (Outcome 3)
   Data storage in memory (arrays, lists). Linking them to user interface elements. Data storage in the device (files,*SQLite*).
5. **Intent** (Outcome 4)
   General information about intent. Implicit and explicit intent. Applications. SMS, use of cameras, sensors.

Required reading

- Bruno Trstenjak: "Presentations for lectures and practical sessions in Mobile Application Programming", Bjelovar University of Applied Sciences.

Further reading

- Android SDK documentation (http://developer.android.com)
- Online tutorial (http://startandroid.ru/en/lessons.html)

- Marko Gargenta, Naučite Android, O'Reilly/IT Expert, 2011

Course title	**LabVIEW Graphical Programming**					
Course instructor(s)	Alan Mutka, PhD, lecturer					
Programme(s) of study	Undergraduate professional programme of study in Mechatronics Undergraduate professional programme of study in Computer Science					
Course status	Compulsory					
Year	3rd	**Semester**	5th	**ECTS**	5	
Contact hours (L+PS+S)	30+30+0	L	PS		S	
			APS	LPS		
		30	0	30	0	

Course objectives

Familiarising students with the fundamentals of graphical programming and the use of the LabView software

Expected learning outcomes

Upon completion of this course students will be able to:

O1: explain the principle of graphical programming in the LabView environment,
O2: apply loops and flow control in graphical programming,
O3: use relation data in graphical programming,
O4: apply modularity in graphical programming,
O5: use modules for connection with peripheral devices,
O6: apply advanced approaches in graphical programming.

Course content

1. Introduction to the *LabVIEW* programming environment (Outcome O1)
What is *LabVIEW*? Virtual instruments (VI). Front panel. Block diagram. Browsing through controls, virtual instruments and functions. Techniques of error correction in programming. Error processing. Program execution.
2. Graphical programming in the *LabVIEW* environment (Outcome O2, O3, O4)
Implementation of VI. Controls and displays. Programming loops. Time delay functions. Graphical data display. *Case* structures. Modular applications. Use of subVI. Arrays. Clusters. Polymorphism. Working with files and text.
3. Measuring and signal generating device (Outcome O5)
Measurement and generation of analogue signals. Digital inputs and outputs. Analogue inputs and outputs. Virtual measuring instruments. Virtual oscilloscope. Time-frequency analysis of the measured signal. Connecting a sensor and actuator to LabView.
4. Advanced LabView graphical programming (Outcome O6)
Sequential programming. State machines. Data transfer between parallel loops. Variables. Queues. Events.

Required reading

- National Instruments: LabVIEW Core 1Course Manual, National Instruments, 2015
- National Instruments: LabVIEW Core 1 Excercises, National Instruments, 2015
- National Instruments: LabVIEW Core 2 Course Manual, National Instruments, 2015
- National Instruments: LabVIEW Core 2 Excercises, National Instruments, 2015

Further reading

- National Instruments: LearnLabVIEW, available on: http://www.ni.com/academic/students/learnlabview/ (2.3.2016)

Course title	Fundamentals of Entrepreneurship						
Course instructor(s)	Goran Kutnjak, PhD, Full Professor						
Programme(s) of study	Undergraduate professional programme of study in Computer Science						
Course status	Elective						
Year	3rd	Semester	5th	ECTS	4		
Contact hours (L+PS+S)	30+15+0	L	PS		S		
			APS	LPS			
		30	15				

Course objectives

Familiarising students with basic terms related to entrepreneurship, the significance of an enterprise in the economic environment and basic elements of entrepreneurial activity of a company.

Expected learning outcomes

Upon completion of this course students will be able to:

- O1: explain and interpret the notion of a company, entrepreneur and entrepreneurship,
- O2: interpret economic conditions for setting up and running a company,
- O3: name and describe business activity principles,
- O4: analyse the company business policy, differentiate between various business activity factors,
- O5: analyse the types of company assets, determine and conduct the distribution of company results,
- O6: single out the elements of work process economisation,
- O7: analyse and interpret capacity and its use,
- O8: determine optimum stock levels,
- O9: interpret the economics of company functions and explain the fundamentals of cost theory.

Course content

1. **Notion of company, entrepreneur and entrepreneurship (Outcome O1)**
2. **Economic conditions of setting up and running a company (Outcome O2)**
3. **Business activity principles (Outcome O3)**
4. **Company business policy. Business activity factors. Company assets. (Outcome O4)**
5. **Fixed assets. Short-term assets. Determining company results. (Outcome O5)**
6. **Income and expenses. Business result and its distribution. (Outcome O6)**
7. **Economisation of work process elements. (Outcome O6)**
8. **Capacities and its usage. (Outcome O7)**
9. **Stock economisation. (Outcome O8)**
10. **Economics of company functions. (Outcome O9)**

Required reading

- Bobera, D., Hunjet. A., Kozina, G. (2015) Poduzetništvo, Varaždin, University North

Further reading

- Hisrich, R.H., Peters, M.P., Shepherd, D.A. (2012) Poduzetništvo, MATE, Zagreb
- Karić, M.(2001) Ekonomika poduzeća, Faculty of Economics in Osijek, Osijek
- Kolaković, M. (2006) Poduzetništvo u ekonomiji znanja, Sinergija, Zagreb

Course title	Business Finance						
Course instructor(s)	Mira Dimitrić, PhD, Full Professor Ivana Tomas Živković, PhD, Assistant Professor						
Programme(s) of study	Undergraduate professional programme of study in Computer Science						
Course status	Elective						
Year	3rd	Semester	5th	ECTS	4		
Contact hours (L+PS+S)	30+15+0	L	PS		S		
			APS	LPS			
		30	15				

Course objectives

The objective of the course is acquiring fundamental knowledge about the financial analysis of business activities, planning money and commodity flow in a company, determining the optimum relation between sources of funds and the structure of funds, familiarising students with the possibilities of funding business activities with short-term, medium-term and long-term loans, funding the development with own means or with loans, planning liquidity and cash flow analysis, identifying business problems.

Expected learning outcomes

Upon completion of this course students will be able to:

- O1: explain financial policies of companies and their objectives, capital budgeting policy, investments in financial instruments, working capital policy and distribution policy,
- O2: apply various techniques and methods of financial management based on compounding and discounting,
- O3: evaluate capital costs and investment yields,
- O4: analyse and apply the instruments of financial analysis of company business activities based on financial reports and information from the capital market,
- O5: substantiate the company funding rules and the concept of position,
- O6: substantiate the influence of capital structure and dividend policy on the value of a company,
- O7: describe various sources and models of company funding, external and internal, including the loan process,
- O8: explain and apply the basic financial risk concept.

Course content

1. Introduction to business finance (Outcome O1)
2. Basic financial concepts of time value of money and the relation between economic and financial categories (Outcome O1)
3. Long-term financial decisions on investing in real investments (capital budgeting) – economic criteria and operating leverage (Outcome O2)
4. Financial criteria of long-term financial decisions on investing in real investments (Outcome O2)
5. Evaluation of investments in debt instruments and costs of funding with debt instruments (Outcome O?)
6. Financial analysis (Outcome O3)
7. Evaluation of investments in ownership instruments and costs of funding with ownership instruments (Outcome O4)
8. Fundamental financial risk concept (Outcome O5)
10. Capital structure (Outcome O6)
11. Dividend policy (Outcome O6)
12. Financial rules and company funding rules (Outcome O7)
13. Types of company funding with respect to maturity (Outcome O8)

Required reading

- Dimitrić, Mira: Business Finance, course material
- Dimitrić, Mira; Tomas Živković, Ivana: Exercises in Business Finance

Further reading

- Orsag, Silvije: Poslovne financije, Avantis; HUFA, 2016

Course title	**Computer and Robot Vision**						
Course instructor(s)	Zoran Vrhovski, senior lecturer						
Programme(s) of study	Undergraduate professional programme of study in Computer Science						
Course status	Elective						
Year	3rd	**Semester**	5th	**ECTS**	4		
Contact hours (L+PS+S)	30+15+0	L	PS		S		
			APS	LPS			
		30	0	15			

Course objectives

1. To familiarise students with algorithms for image pre-processing and analysis
2. To master the tools for the extraction of useful information from an image on a computer

Expected learning outcomes

Upon completion of this course students will be able to:

- O1: acquire and analyse an image using the LabVIEW software,
- O2: acquire and analyse an image using the OpenCV software,
- O3: develop a program that analyses deep image from stereo cameras.

Course content

1. Introduction (Outcome O1)
Introduction to computer vision. An overview of image analysis software (LabVIEW, OpenCV). An overview of image acquisition hardware. Image display in a computer.
2. Camera and acquisition (Outcome O1)
Types of camera. Communication protocols for image transmission. Image acquisition hardware. Illumination. Image parameters during acquisition. Mathematical model of camera. Camera calibration. Camera acquisition in the LabVIEW environment.
3. Image pre-processing (Outcome O1)
Region of interest. Basic operators on an image. Image rotation. Mirroring. Image warping. Image filtering. Image smoothing.
4. Image analysis (Outcome O1)
Morphological image processing. Edge extraction. Colour-based image retrieval in HSV or RGB colour space. Shape detection. Histograms.
5. OpenCV (Outcome O2)
An overview of the OpenCV library for image pre-processing and analysis. Image acquisition. Image processing and analysis. Practical examples.
6. Advanced application and use thereof (Outcome O3)
Robot vision. Getting a deep image from one camera. Getting a deep image from two cameras (stereo vision). OCR. DICOM. Barcodes.

Required reading

- Zoran Vrhovski, "Presentations for lectures and practical sessions in Computer and Robot Vision", Bjelovar University of Applied Sciences.

Further reading

- Christopher G. Refl, Image Acquisition and Processing with LabVIEW, CRC Press, 2003
- O. Faugeras, Three-dimensional Computer Vision, MIT Press, 1993
- R. Jain et al., Machine Vision, McGraw-Hill, 1995

Course title	**Field Practice 1**				
Course instructor(s)	Zoran Vrhovski, senior lecturer				
Programme(s) of study	Undergraduate professional programme of study in Computer Science				
Course status	Compulsory				
Year	3rd	**Semester**	5th	**ECTS**	3

Contact hours (L+PS+S)	0 + 80 + 10	L	PS		S
			APS	LPS	
				80	10

Course objectives

1. Familiarising oneself with the company, work environment and colleagues.
2. Gaining an insight into the organisation and work in a real work environment.
3. Learning to take on and carry out specific work tasks.

Expected learning outcomes

Upon completion of this course students will be able to:

O1: analyse business processes within a company,
O2: assess the usage proportion of individual technologies in a concrete IT project,
O3: design their own application or a part of a concrete application, write the pseudocode and implement it.

Course content

Required reading

- Guidelines for Computer Science students on doing the field practice.
- Field practice journal.

Further reading

Course title	**Computer and Data Security**							
Course instructor(s)	Dario Vidić, lecturer							
Programme(s) of study	Undergraduate professional programme of study in Computer Science							
Course status	Compulsory							
Year	3rd	**Semester**	6th	**ECTS**	6			
Contact hours (L+PS+S)	30 + 30 + 0			L	PS		S	
					APS	LPS		
				30	0	30	0	

Course objectives

1. Familiarising students with the concept of information system protection and methods of threat prevention and risk recognition/detection.
2. Understanding the risks of networked systems.
3. Understanding the weaknesses of operating systems.
4. Familiarising students with the approaches to enhancing the security of operating systems.

Expected learning outcomes

Upon completion of this course students will be able to:

- O1: describe the basic concepts of security and security threats to operating systems,
- O2: explain weaknesses of operating systems and software, and apply enhancements on a concrete operating system,
- O3: explain security weaknesses of databases and describe the methods of security risk reduction,
- O4: compare and use protocols that allow for the security or networks and mobile devices,
- O5: explain and apply basic cryptography methods while using computer systems.

Course content

1. **Introduction to computer security (O1)**
   Basic concepts, security threats, security objectives.
2. **Operating system security (O2)**
   Access control, authorisation, user authentication. Memory protection. Biometric identification.
3. **Software security (O2)**
   Analysis and defence against a malicious code (viruses, spyware, rootkits). Web security: XSS and XSRF attacks and defences.
4. **Database security (O3)**
   Access control, privacy.
5. **Network security (O4)**
   IPSec, SSL/TSL, DDoS attacks, DNS, network firewalls.
6. **Security of mobile devices (O4)**
7. **Fundamentals of cryptography (O5)**
   Encryption, authentication, random numbers, digital signature.

Required reading

1. Dario Vidić: "Presentations for lectures and practical sessions in Computer and Data Security", Bjelovar University of Applied Sciences.

Further reading

1. Dieter Gollmann: Computer Security, 2nd edition, Wiley, 2005
2. Miroslav Bača: Uvod u računalnu sigurnost, Narodne novine, Zagreb, 2004

Course title	**Field Practice 2**						
Course instructor(s)	Zoran Vrhovski, senior lecturer						
Programme(s) of study	Undergraduate professional programme of study in Computer Science						
Course status	Compulsory						
Year	3rd	**Semester**	6th	**ECTS**	5		
Contact hours (L+PS+S)	0 + 140 + 10	L	PS		S		
			APS	LPS			
				140	10		

Course objectives

1. Familiarising oneself with the company, work environment and colleagues.
2. Gaining an insight into the organisation and work in a real work environment.
3. Learning to take on and carry out specific work tasks.

Expected learning outcomes

Upon completion of this course students will be able to:

O1: analyse business processes within a company,
O2: assess the usage proportion of individual technologies in a concrete IT project,
O3: design their own application or a part of a concrete application, write the pseudocode and implement it.

Course content

Required reading

- Guidelines for Computer Science students on doing the field practice.
- Field practice journal.

Further reading

Course title	Thesis						
Course instructor(s)	Course instructors in the professional programme of study in Computer Science						
Programme(s) of study	Undergraduate professional programme of study in Computer Science						
Course status	Compulsory						
Year	3rd	Semester	6th	ECTS	11		
Contact hours (L+PS+S)	0 + 0 + 150	L	PS		S		
			APS	LPS			
					150		

Course objectives

1. To individually research and cover a selected topic, using the theoretical and practical knowledge acquired during studies.
2. To use the achieved competences in solving problems related to the field of study and to successfully use professional and scientific literature in the written coverage of the topic.

Expected learning outcomes

Upon completion of this course students will be able to:

- O1: cover topics related to the area of computer science using the theoretical and practical knowledge acquired in the course of studies,
- O2: document the researched topic related to the field of computer science,
- O3: present the researched topic related to the field of computer science.

Course content

Course content is based on the selected thesis topic.

Required reading

- Literature to be used for thesis depends on the topic selected and covered. Literature used in a thesis shall be listed in the printed version thereof.

Further reading

Course title	**Network Programming**							
Course instructor(s)	Zoran Vrhovski, senior lecturer							
Programme(s) of study	Undergraduate professional programme of study in Computer Science							
Course status	Elective							
Year	3rd	**Semester**	6th	**ECTS**	4			
Contact hours (L+PS+S)	30 + 15 + 0			L	PS		S	
					APS	LPS		
				30	0	15	0	

Course objectives

1. Rendering students capable of understanding the fundamental concept of network programming
2. Familiarising students with the C# .NET software for network programming
3. Teaching students to implement the basic network applications

Expected learning outcomes

Upon completion of this course students will be able to:

- O1: recognise and explain the basic architecture and network elements required for setting up the communication between network applications,
- O2: explain the possibilities of TCP/IP and UDP protocols for connecting two applications in the network environment,
- O3: design a simple network server-client application in the C#. NET environment,
- O4: explain and select appropriate input-output models for achieving network communication,
- O5: select appropriate protocols and design a simple application for connecting to a web and e-mail server.

Course content

1. **Introduction (O1)**
   Introduction to computer networks. Network architectures. An overview of fundamental concepts: socket, IPv4 address, clients, servers. The C# .NET software for network programming.
2. **Input/output operations in the .NET environment (O2, O3)**
   An overview of C# classes for input/output operations. The *Socket* class. Server functions. Client functions. Programming a UDP client/server. Programming a TCP/IP client/server. An overview of simple classes for network programming: *TcpClient*, *TcpListener*, *UdpClient*.
3. **Synchronous and asynchronous input/output operations (O4)**
   Models of input/output operations. Socket multiplexing. Using threads for working simultaneously with more users.
4. **Communication with web and e-mail servers (O5)**
   HTTP communication with web-servers. Designing an application for communication with a web server. E-mail protocols (POP3, IMAP, SMTP). Designing an application for sending electronic mail. Data protection. Access control.

Required reading

1. Dario Vidić, "Presentations for lectures and practical sessions in Network Programming", Bjelovar University of Applied Sciences.

Further reading

1. Richard Blum, C# Network Programming, (2003), Sybex.
2. David B. Makofske, Michael J. Donahoo, Kenneth L. Calvert, TCP/IP Sockets in C#, (2004), Elsevier Digital Press.

Course title	**Computer Graphics**						
Course instructor(s)	Alan Mutka, PhD, lecturer						
Programme(s) of study	Undergraduate professional programme of study in Computer Science						
Course status	Elective						
Year	3rd	**Semester**	6th	**ECTS**	4		
Contact hours (L+PS+S)	30+15+0	L	PS		S		
			APS	LPS			
		30	0	15	0		

Course objectives

1. Acquisition of basic knowledge of implementing computer graphics.
2. Familiarisation with the OpenGL library.

Expected learning outcomes

Upon completion of this course students will be able to:

- O1: define the features of contemporary graphics hardware,
- O2: develop programs that use graphics primitives and apply geometrical transformations,
- O3: explain the principles of modelling 3D objects and representation of objects,
- O4: apply knowledge in mathematics, physics and programming on graphics applications and solve problem tasks,
- O5: solve the problems of 3D graphics display and the development of graphics applications.

Course content

1. **Introduction** (Outcome 1)
   Graphic flow system. Introduction to OpenGL. The first program. Graphics hardware. Graphics software.
2. **Graphics primitives and transformations** (Outcome 2)
   Two-dimensional (point, line). 2D transformations. Three-dimensional (point, line, plane). 3D transformations. View and projection transformation.
3. **Modelling and representation of objects** (Outcome 3)
4. **Interpolation, curves** (Outcome 4)
   Linear interpolation (bilinear, trilinear). Curves. Curve segment.
5. **Displaying 3D surfaces** (Outcome 5)
   Hidden line and hidden surface removal (procedures in object and projection space). Models and procedures of illumination, shading. Colour in computer graphics. Textures. Fractals.

Required reading

- Alan Mutka, "Presentations for lectures and practical sessions in Computer Graphics", Bjelovar University of Applied Sciences

Further reading

- J. D. Foley, A. van Dam, S. K. Feiner, J. F. Hughes: Computer Graphics - Principles and Practice, 2nd edition in C, Addison-Wesley, 2001
- D. Hearn, M. P. Baker: Computer Graphics with Open GL, 3rd edition, Prentice-Hall, 2003
- Čupić, Mihajlović: Interaktivna računalna grafika kroz primjere u OpenGL-u, course material, Faculty of Electrical Engineering and Computing, University of Zagreb, 2016

Course title	Financial Institutions and Markets						
Course instructor(s)	Zdenko Prohaska, PhD, Full Professor Bojana Olgić Draženović, PhD, Assistant Professor						
Programme(s) of study	Undergraduate professional programme of study in Computer Science						
Course status	Elective						
Year	3rd	Semester	6th	ECTS	4		
Contact hours (L+PS+S)	30+15+0	L	PS		S		
			APS	LPS			
		30	15				

Course objectives

1. Acquisition of knowledge of features and functions of financial institutions and financial markets and the regulation thereof.

Expected learning outcomes

Upon completion of this course students will be able to:

- O1: describe the financial system and financial instruments,
- O2: explain changes taking place on the money market, capital market and foreign exchange market,
- O3: explain the legal framework of the capital market in Croatia,
- O4: describe contemporary trends on financial markets,
- O5: describe trends on the securities market in Croatia,
- O6: explain the international capital market.

Course content

1. **Financial system** (Outcome 1)
   Features and functions of the financial system. Supply and demand on the financial market. Surplus units, deficit units, balanced budget units. Development of the financial market.
2. **Financial assets, receivables and instruments** (Outcome 1)
   Money. Equity. Debt securities. Derivatives. Financial pyramid. Agency ratings. Rating symbols.
3. **Financial institutions** (Outcome 1)
   Central bank. Commercial banks. Active, passive and neutral banking operations. Banks' credit potential. Integration of banks.
4. **Non-banking financial institutions** (Outcome 1)
   Contractual savings institutions. Investment funds. Financial companies. State-owned financial institutions. Investment banks.
5. **Money market. Foreign exchange market** (Outcome 2)
   Money market. Short-term securities market. Spot foreign exchange market. Foreign exchange futures market.
6. **Capital market** (Outcome 2)
   Primary and secondary market. Stock exchange. OTC market. Stock exchange indices. Bull and bear market cycles.
7. **Legal framework and regulation of the capital market** (Outcome 3)
   The Capital Market Act. The Investment Funds Act. The Privatisation Investment Funds Act. Act on the Takeover of Joint Stock Companies. The Leasing Act.
8. **Capital market institutions** (Outcome 3)
   Croatian Financial Services Supervisory Agency. Central Depository and Clearing Company. Intermediaries in securities traffic.
9. **Investors on capital market** (Outcome 3)
   Individual investors. Institutional investors. Investment funds. Pension funds. Insurance companies.
10. **Contemporary trends on the financial market** (Outcome 4)
    Internationalisation of finances. Regulation and deregulation. Asset securitisation. Electronic money transfer. Telebanking. Expert systems.
11. **The role of the state in financial markets** (Outcome 4)
    Regulation. Public debt policy. Deposit insurance system.
12. **Deposit and non-deposit financial institutions in the Republic of Croatia** (Outcome 5)
    Banks. Savings banks. Insurance companies. Pension funds. Investment funds.

13. **Securities market in the Republic of Croatia** (Outcome 5)
    Brokerage companies. Zagreb Stock Exchange. Money market. Institutional investors.
14. **International financial market** (Outcome 5)
    World Bank. International Monetary Fund. European Bank for Reconstruction and Development. European Investment Bank. International capital markets.

Required reading

- Klačmer Čalopa, M., Cingula, M. (2009): Financijske institucije i tržišta kapitala, Faculty of Organisation and Informatics, University of Zagreb

Further reading

- Leko, V. (2012): Financijske institucije i tržišta, Faculty of Economics and Business, University of Zagreb

Course title	Bank Management							
Course instructor(s)	Saša Živković, PhD, Full Professor							
Programme(s) of study	Undergraduate professional programme of study in Computer Science							
Course status	Elective							
Year	3rd	Semester	6th	ECTS	4			
Contact hours (L+PS+S)	30+15+0	L	PS		S			
			APS	LPS				
		30	15		0			

Course objectives

1. Familiarising students with the development, functions and organisation of banks, banking activities and specific relations between banks and their clients.
2. Familiarising students with the foundation and business activities of banks within economic management of the bank as the most important institution in financial mediation.

Expected learning outcomes

Upon completion of this course students will be able to:

- O1: define basic terms in banking,
- O2: describe the principles of banking activities,
- O3: explain the bank capital structure,
- O4: explain basic banking activities,
- O5: compare the payment system in Croatia with the payment systems abroad.

Course content

1. **Notion and types of banks and other financial institutions** (Outcome 1)
2. **Croatian banking system** (Outcome 1)
3. **Foundation and management of a bank** (Outcome 1)
4. **Principles of banking activities** (Outcome 2)
5. **Bank regulation and supervision** (Outcome 2)
6. **Monitoring, control and relations between a bank and the central bank, control organs and other institutions** (Outcome 2)
7. **Capital and bank capital structure: theory and regulation** (Outcome 3)
8. **Bank activities, income and expenses** (Outcome 3)
9. **Measuring the success rate of bank activities** (Outcome 3)
10. **Fundamental banking activities** (Outcome 4, Outcome 5)
    Mobilisation of capital in the country and from abroad. Investments in loan and non-loan placements. Neutral or mediation activities. Own and other banking activities. Payment system in the country and abroad (principles, instruments, techniques).

Required reading

- Rose, S.P., (2003), Menadžment komercijalnih banaka, Mate, Zagreb

Further reading