Faculty of Computer Science and Information Technology

WEST POMERANIAN UNIVERSITY OF TECHNOLOGY
IN SZCZECIN, POLAND

THE OFFER FOR INTERNATIONAL STUDENTS
FOR THE YEAR 2021/2022
FIRST DEGREE

	Course title	Person responsible for the course	Semester (winter/summer)	ECTS points	Hours
1	3D printing and design	Radosław Mantiuk	winter/summer	3	45
2	Artificial Intelligence	Przemysław Klęsk	winter/summer	3	30
3	Artificial Neural Networks	Marcin Pluciński	winter/summer	3	30
4	Audio Signal Processing	Mirosław Łazoryszczak	winter/summer	3	30
5	Bash – Command Language Interpreter for Engineers	Magdalena Szaber-Cybularczyk	winter/summer	2	30
6	Big Data analytics tools and software	Agnieszka Konys	winter/summer	4	60
7	Brain-Computer Interface	Izabela Rejer	winter/summer	4	60
8	Business Intelligence	Przemysław Różewski	winter/summer	4	60
9	C# Programming Language	Marcin Pietrzykowski	winter/summer	3	45
10	C++ programming language	Agnieszka Konys	winter/summer	4	60
11	Compilers	Włodzimierz Bielecki	winter/summer	4	60
12	Computer Games Programming	Radosław Mantiuk	summer	4	75
13	Computer Networks	Remigiusz Olejnik	winter/summer	4	60
14	Computer System Architecture	Mariusz Kapruziak	winter/summer	4	60
15	Computer Vision for Video Surveillance	Adam Nowosielski	winter/summer	3	30
16	Database systems	Przemysław Korytkowski	winter/summer	5	60
17	Data Mining Algorithms	Przemysław Klęsk	winter/summer	4	45
18	Data Warehousing and Big Data	Przemysław Różewski	winter/summer	4	60
19	Deep learning for visual computing	Radosław Mantiuk	winter/summer	5	60
20	Digital Circuits	Mirosław Łazoryszczak	winter/summer	4	60
21	Dynamic documents and front-end Web development	Jarosław Jankowski	winter	3	45
22	E-commerce and online marketing technologies	Jarosław Jankowski	winter	3	45
23	EEG signal analysis in Matlab	Izabela Rejer	winter/summer	4	60
24	Embedded systems	Mirosław Łazoryszczak	winter/summer	4	60
25	Essentials of Fuzzy Logic	Marcin Pluciński	winter/summer	4	60
26	Expert systems	Joanna Kołodziejczyk	winter/summer	3	45
27	F# Programming Language	Marcin Pietrzykowski	winter/summer	3	45
28	FPGA Design and reconfigurable computing	Mariusz Kapruziak	winter/summer	3	45
29	Fundamentals of Error-Correcting Block Codes	Dorota Majorkowska-Mech	winter	3	30

	Course title	Person responsible for the course	Semester (winter/summer)	ECTS points	Hours
30	Graphical User Interface in .NET	Marcin Pietrzykowski	winter/summer	2	30
31	Human-Computer Interaction	Adam Nowosielski	winter/summer	3	30
32	Intelligent Decision Systems	Wojciech Sałabun	winter/summer	4	60
33	Introduction to front-end application development	Bartłomiej Małachowski	winter/summer	3	45
34	Introduction to Natural Language Processing	Joanna Kołodziejczyk	winter/summer	4	60
35	Introduction to the Internet of Things	Remigiusz Olejnik	winter/summer	4	60
36	Intro to Mathematical Programming	Wojciech Sałabun	winter/summer	4	60
37	Intro to Statistic: Making Decisions Based on Data	Wojciech Sałabun	winter/summer	4	60
38	Knowledge Engineering and Ontology Development	Agnieszka Konys	winter/summer	4	60
39	LaTeX	Remigiusz Olejnik	winter/summer	2	30
40	Machine Learning	Przemysław Klęsk	winter/summer	2	30
41	Microprocessor design and soft-processors	Mariusz Kapruziak	winter/summer	3	45
42	Mobile Application Development	Radosław Maciaszczyk	winter/summer	4	45
43	Parallel Programming	Włodzimierz Bielecki	winter/summer	4	60
44	Programmable control devices 1	Sławomir Jaszczak	winter/summer	3	45
45	Programmable control devices 2	Sławomir Jaszczak	summer	3	45
46	Prolog Programming for Artifcial Intelligence	Joanna Kołodziejczyk	winter/summer	3	45
47	Python GUI Programming	Krzysztof Małecki	winter/summer	3	45
48	Python Programming Language	Krzysztof Małecki	winter/summer	4	60
49	Social media and complex network analytics	Jarosław Jankowski	winter	3	45
50	Software engineering	Łukasz Radliński	winter	3	45
51	Software for Engineering Calculations	Marcin Pluciński	winter/summer	2	45
52	Stochastic Optimization	Jan Rodziewicz-Bielewicz	winter/summer	4	60
53	Web application development with Angular framework	Bartłomiej Małachowski	winter/summer	3	45
54	Алгоритмические основы цифровой обработки сигналов и изображений	Aleksandr Cariow	winter/summer	3	30

Course title	3D printing and design		
Level of course	first cycle		
Teaching method	project course / lecture		
Person responsible for the course	Radosław Mantiuk	**E-mail address to the person**	Radoslaw.Mantiuk@zut.edu.pl
Course code (if applicable)	WI-1-3DD	**ECTS points**	3
Semester	winter/summer	**Language of instruction**	english
Hours per week	3	**Hours per semester**	45
Objectives of the course	Transfer of knowledge and gaining skills and competences in the field of CD model preparation and ctual 3D printing.		
Entry requirements	Without any initial requirements		
Course contents	Implementation of a project involving the design of the object in CAD software, preparation of the g-code in CAM software, and printing the object on the FFF printer. 3D printing technologies. FFF printer construction. Introduction to CAD software. Tutorial: designing basic object. CAD techniques. Introduction to CAM software. Object manufacturing.		
Assessment methods	Lecture Workshops Preparation of the CAD model and physical print of this model.		
Recommended readings	1. Ben Redwood, Brian Garret, Filemon Schöffer, and Tony Fadell, The 3D Printing Handbook: Technologies, Design and Applications, Google Book, 2018		
Knowledge	Gaining knowledge on 3D printing and design		
Skills	Gaining skills on 3D printing and design		
Other social competences	Gaining competences on 3D printing and design		

Course title	Artificial Intelligence		
Level of course	first cycle		
Teaching method	laboratory course / lecture		
Person responsible for the course	Przemysław Klęsk	**E-mail address to the person**	pklesk@wi.zut.edu.pl
Course code (if applicable)	WI-1-IAI	**ECTS points**	3
Semester	winter/summer	**Language of instruction**	english
Hours per week	2	**Hours per semester**	30
Objectives of the course	Familiarization with various search techniques for practical problems. Introducing elements of two-person games of perfect information and algorithms for that purpose. Building up the understing of such notions as: heuristics, pay-off, strategy, search horizon. Familiarization with classification and approximation as exemplary tasks within machine learning. Introducing simple artificial neural networks for that purpose. Teaching a possibility of solving optimization problems by means of randomized methods (genetic algorithms). Giving a historical background on AI and problems within it.		
Entry requirements	mathematics algorithms and data structures programming object oriented programming		
Course contents	Gatting familiar with Java, Eclipse IDE, and a set of classes prepared for implementations of search algorithms. Initial implementation of the sudoku solver. Implementation of sudoku solver. Testing - varations on the initial state (making the sudoku harder). Observing the number of visited states and the number of solution. Posing the homework task - programming the solver for the sliding puzzle. Testing homework programs - sliding puzzle solvers. Getting familiar with Java classes prepared for game tree searches (alpha-beta pruning engine). Posing the homework task - programming an AI playing the connect4 game. Testing homework programs - connect4 program: experimentations with different search depths, program vs program games, comments on introduced heuristics (position evaluation). Programming the simple perceptron (in MATLAB). Two-class separation of points on a plane. Observing the number of update steps in learning algorithm influenced by: learning rate coefficient, number of data points (sample size), changes in separation margin. Posing the homework task - implementation of non-linear separation using the simple perceptron together with the kernel trick. Implementation of MLP neural network (in MATLAB) for approximation of a function of two variables. Testing accuracy with respect to: number of neurons, learning coefficient, number of update steps. Posing the homework task: complexity selection for MLP via cross-validation. Genetic algorithm implementation for the knapsack problem, including: at least two selection methods, and two crossing-over methods. Posing the homework task: comparison of GA solutions with exact solutions based on dynamic programming (computation times). Definitions of AI and problems posed within it, e.g.: graph and game tree search problems - n-queens, sliding puzzle, sudoku, minimal sudoku, jeep problem, knapsack problem, traveling salesman problem, prisonner's dilemma, iterated prisonner's dilemma, pattern recognition / classification, imitation game (Turing's test), artificial life and cellular automata, Conway's game of life. Minsky's views on AI. Graph search algorithms: Breadth-First-Search, Best-First-Search, A*, Dijkstra's algorithm. Notion of heuristics. Efficient data structures for implementations of above algorithms: hash map, priority queue (heap). Algorithms for two-person games of perfect information: MIN-MAX, alpha-beta pruning, and their computational complexity. Horizon effect. Data classification (binary, linear) using the simple perceptron (Rosenblatt's perceptron). Forward pass. Learning algorithm. Linear separability of data. Novikoff's theorem on learning convergence (with the proof). Multi-Layer-Perceptron (MLP) artificial neural network. Sigmoid as activation function. On-line vs off-line learning. Derivation of the back-propagation algorithm. Possible variants. Overfitting and complexity selection for MLP via testing or cross-validation. Genetic algorithms for optimization problems. Scheme of main genetic loop. Fitness function. Selection methods in GAs: roulette selection, rank selection, tournaments. "Exploration vs. exploitation" problem. Remarks on convergence, premature convergence (population diversity). Crossing-over methods: one-point, two-points, multiple-point crossing-over. Mutation andits role in GAs (discrete and continuous). Examples of problems: knapsack problem, TSP. Exact solution of knapsack problem via dynamic programming. Exam.		
Assessment methods	Lecture. Case study method. Didactic games. Computer programming. Demonstration. Five short tests (10 minutes long) at the end of each topic during the lab.		

	Five grades for the programs written as homeworks. Final grade for the lab calculated as a weighted mean from partial grades: - tests (weight: 40%), - programs (weight: 60%). Final grade for lectures from the test (1.5 h).
Recommended readings	1. S. Russel, P. Norvig, Introduction to Artificial Intelligence, A Modern Approach, Prentice Hall, 2010, 3rd edition
Knowledge	Student has an elementary knowledge on AI problems and algorithmic techniques applicable to solve them.
Skills	Student can design and implement elementary AI algorithms.

Course title	Artificial Neural Networks		
Level of course	first cycle		
Teaching method	laboratory course / lecture		
Person responsible for the course	Marcin Pluciński	**E-mail address to the person**	Marcin.Plucinski@zut.edu.pl
Course code (if applicable)	WI-1-ANN	**ECTS points**	3
Semester	winter/summer	**Language of instruction**	english
Hours per week	2	**Hours per semester**	30
Objectives of the course	Extending of the knowledge about artificial neural networks, their construction, operation and learning techniques. Gaining practical skills in the application of neural networks to solve real tasks of modeling and classification. Familiarization with the software that could be used in tasks of modeling and classification using neural networks.		
Entry requirements	Basics of algebra and mathematical analysis. Basics of computer science.		
Course contents	Application of simple perceptron neural network to solve classification tasks. Application of feed-forward multilayer neural networks to solve complex real tasks of classification. Application of feed-forward multilayer neural network in modeling (real technical, economic and medical problems). Applications of RBF neural networks in modeling of technical and economic problems. Application of unsupervised learning networks to the data clustering problem. Hopfield network - application to the pattern recognition problem. Final work. The introduction to neural networks. Feed-forward neural networks. The structure and operation of the artificial neuron. Simple Perceptron network - structure and learning methods. Example of learning and action of the network. Selected applications of the Perceptron network. Feed-forward multilayer neural networks. Network learning methods - backpropagation. Examples of learning and operation of the network. Selected network applications. Selection of the optimal network architecture. Neural networks with radial basis function - RBF neural networks. Structure and learning methods. Examples of applications. Probabilistic neural networks. Self-organizing networks - unsupervised learning algorithms. The strucrure and operation of networks. Kohonen's network and learning algorithm. Examples of applications of self-organizing networks. Recursive networks - Hopfield network, Hamming network. Construction, operation, learning methods. Examples of network applications. Evaluation of knowledge.		
Assessment methods	Lecture with presentation. Labs - self-realization of tasks with the application of neural networks. Work will be done using Matlab ANN Toolbox and self-developed software. Lecture: written test. Laboratory: evaluation of tasks carried out during the classes. Laboratory: evaluation of reports. Laboratory: evaluation of final work.		
Recommended readings	1. David Kriesel, A Brief Introduction to Neural Networks, 2012 2. James A. Freeman, David M. Skapura, Neural Networks: Algorithms, Applications, and Programming Techniques, Addison-Wesley Publishing Company, 2005		
Knowledge	The student knows the types of artificial neural networks, their structure, operation and ways of learning. The student knows practical applications of specific types of artificial neural networks.		
Skills	The student has the ability to solve practical problems (economic, technical and other) using artificial neural networks.		

Course title	Audio Signal Processing		
Level of course	first cycle		
Teaching method	laboratory course / lecture		
Person responsible for the course	Mirosław Łazoryszczak	**E-mail address to the person**	Miroslaw.Lazoryszczak@zut.edu.pl
Course code (if applicable)	WI-1-ASP	**ECTS points**	3
Semester	winter/summer	**Language of instruction**	english
Hours per week	2	**Hours per semester**	30
Objectives of the course	Getting familiar with basic issues and selected methods of sound processing.		
Entry requirements	Basics of programming and signal processing.		
Course contents	Audio signal generating and manipulating using selected programming tools. Creating simple GUI framework for audio processing Selected digital filter implementation Audio effects implementation eg. delay, echo, pitch shift etc. Music pitch retrieval methods Basic of sound. Audio perception. Acoustical signal acquisition. Transducers – microphones and speakers. Home recording studios: acoustics and equipment Audio signal representations and sound analysis. Digital filters. Sound effects. Sound modeling and synthesis. Selected applications of audio processing eg. noise reduction, automatic recognition of music. Assessment		
Assessment methods	Presentation lecture Laboratory work Lecture - written exam Labs - written reports		
Recommended readings	1. Rocchesso D., Introduction to Sound Processing, Verona, 2003, https://archive.org/download/IntroductionToSoundProcessing/vsp.pdf		
Knowledge	The student knows the basic attributes of audio signals, the ways of their perception and selected processing methods.		
Skills	The student is able to implement basic problems of sound processing using the selected programming language.		

Course title	Bash – Command Language Interpreter for Engineers		
Level of course	first cycle		
Teaching method	laboratory course / lecture		
Person responsible for the course	Magdalena Szaber-Cybularczyk	**E-mail address to the person**	Magdalena.Szaber@zut.edu.pl
Course code (if applicable)	WI-1-BCL	**ECTS points**	2
Semester	winter/summer	**Language of instruction**	english
Hours per week	2	**Hours per semester**	30
Objectives of the course	Practical skills, allowing the user to type command and scripts which cause actions.		
Entry requirements	None.		
Course contents	Basic commands for files and directories. Redirection, Expansion and Quoting. Positional parameters and arthmetics. Array. Flow control. Functions. What's BASH? Basic commands for files and directories. How to make a script? Using the most popular text-editors. How it works : redirection, expansion, quoting, positional parameters and array. Flow Control : Branching with if and case, looping with while/until and for. How to write a function ?		
Assessment methods	Laboratories. Lectures. Continuous assessment.		
Recommended readings	1. Carl Albing, JP Vossen, Cameron Newham, Bash Cookbook Solutions and Examples for bash Users. 2nd Edition, O'Reilly, 0, 2017 2. Cameron Newham and Bill Rosenblatt. £rd Edition, Learning the Bash Shell, O'Reilly, 0, 2005		
Knowledge	Student will gain knowledge of type command and scripts.		
Skills	Student will gain practical skills in BASH.		

Course title	Big Data analytics tools and software		
Level of course	first cycle		
Teaching method	laboratory course / lecture		
Person responsible for the course	Agnieszka Konys	**E-mail address to the person**	Agnieszka.Konys@zut.edu.pl
Course code (if applicable)	WI-1-BDA	**ECTS points**	4
Semester	winter/summer	**Language of instruction**	english
Hours per week	4	**Hours per semester**	60
Objectives of the course	Familiar with the tools and software for large scale datasets The ability to analyze the characteristics of data reaching the IT system, knowledge of the tasks that need to be dealt with to process this data, and the creation and selection of appropriate methods, computer environment and software in order to effectively solve the tasks.		
Entry requirements	None		
Course contents	Application of information extraction methods and techniques Application of methods and tools for analyzing data from Internet of Things devices Implementation of models for processing large data sets Big data processing and analysis tools Graph Database and Analytics tools Big Data Visualization tools Information extraction from text Methods and techniques for information extraction Methods and tools for analyzing data from Internet of Things devices Models for processing large data sets Big data processing and analysis tools Graph Database and Analytics tools Big Data Visualization tools Exam		
Assessment methods	Informative lectures Discussion Work with computers at laboratories Written exam Continuous assessment		
Recommended readings	1. Martin Kleppmann, Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems, O'Reilly, United States of America, 2017 2. Tom White, Hadoop: The Definitive Guide (4th Edition), O'Reilly, 2015, ISBN: 9781491901632 3. Vince Reynolds, Big Data For Beginners: Understanding SMART Big Data, Data Mining & Data Analytics For improved Business Performance, Life Decisions & More! (Data ... Computer Programming, Growth Hacking, ITIL), Createspace Independent Publishing Platform, 2016		
Knowledge	After the course the student should have knowledge of the methods, algorithms and software to solve particular problems of processing large data sets. After the course the student should have knowledge of the methods and tools for data analysis on large data sets.		
Skills	The student should know how to use methods and tools for data analysis on large data sets. The student should be able to analyze and classify data features, choose the appropriate software and techniques for data processing and apply research results to solve specific problems.		
Other social competences	The student is competent in solving large data processing tasks using modern methods, algorithms and programs and can apply knowledge and skills in this field to solve specific problems.		

Course title	Brain-Computer Interface		
Level of course	first cycle		
Teaching method	laboratory course / lecture		
Person responsible for the course	Izabela Rejer	**E-mail address to the person**	irejer@wi.zut.edu.pl
Course code (if applicable)	WI-1-BCI	**ECTS points**	4
Semester	winter/summer	**Language of instruction**	english
Hours per week	4	**Hours per semester**	60
Objectives of the course	To provide the knowledge about EEG devices, the features of EEG data, and the methods for transforming EEG data to signals used for controling brain computer interfaces. To equip the students with the ability of designing and programming interfaces controlling the external devices with brain waves.		
Entry requirements	None		
Course contents	The applications for EEG data analysis. Tests of different EEG devices. Creating a BCI for a given control task. Testing the interface with real users. Exam. Brain Computer Interface (BCI) - the main paradigms. The main parts of a human brain. The main structure of BCI Controling external devices with BCI. Methods for EEG data preprocessing, feture extraction and classification used in BCI. Exam.		
Assessment methods	Informative lectures. Discussion. Laboratories with computers and EEG devices. The final report describing the created interface, tests results, and the conclusions. The final discussion summing up the knowlegde gained during the lectures.		
Recommended readings	1. Lotte F., Study of Electroencephalographic Signal Processing and Classification Techniques towards the use of Brain-Computer Interfaces in Virtual Reality Applications, 2008, PhD Thesis, https://sites.google.com/site/fabienlotte/phdthesis		
Knowledge	After the lectures the student will be able to: define a BCI, describe the main problems with EEG data, describe the EEG device, descibe different BCI paradigms, choose the processing methods suitable for different paradigms and different EEG data.		
Skills	The student will be able to create the project of a BCI suitable for a given task.		

Course title	Business Intelligence		
Level of course	first cycle		
Teaching method	laboratory course / lecture		
Person responsible for the course	Przemysław Różewski	**E-mail address to the person**	Przemyslaw.Rozewski@zut.edu.pl
Course code (if applicable)	WI-1-BIN	**ECTS points**	4
Semester	winter/summer	**Language of instruction**	english
Hours per week	4	**Hours per semester**	60
Objectives of the course	Understanding key concepts and tools in business intelligence, data analysis, and data visualization.		
Entry requirements	SQL basics, basic understanding of business processes		
Course contents	Analysis of Multiple Business Perspectives Dashboard Design in PowerBI Business Intelligence Concepts Business Analytics Fundamentals Data Description and Visualization Dashboard Design Business Performance Management Systems		
Assessment methods	Informative lectures Cases studies Project Written exam		
Recommended readings	1. Grossmann, Wilfried, Rinderle-Ma, Stefanie, Fundamentals of Business Intelligence, Springer-Verlag Berlin Heidelberg, 2015, DOI: 10.1007/978-3-662-46531-8		
Knowledge	Understanding key concepts in business intelligence, data analysis, and data visualization		
Skills	Be able to effective use Data Visualization and Dashboard tool.		

Course title	C# Programming Language		
Level of course	first cycle		
Teaching method	laboratory course / lecture		
Person responsible for the course	Marcin Pietrzykowski	**E-mail address to the person**	Marcin.Pietrzykowski@zut.edu.pl
Course code (if applicable)	WI-1-CPL	**ECTS points**	3
Semester	winter/summer	**Language of instruction**	english
Hours per week	3	**Hours per semester**	45
Objectives of the course	Familiar with the sytnax, structures and principles used in the c# language The ability to develop an object-oriented program in c# language.		
Entry requirements	None		
Course contents	Introduction to visual Studio IDE and C# Data types, operators Controlling Programmatic Flow Exceptions Constructing Complex Types: classes and structs Inheritance, Abstraction, Object Interfaces Generic Types Generic Collections Input-output operations Threading, parallelism and asynchronous operations Windows Forms Applications Introduction to: Object Oriented Programming, Managed Languages and C# Controlling Programmatic Flow, Manipulating Types Constructing Complex Types, Object Interfaces and Inheritance Generic Types and Collections Input-output operations and multi threading Windows Forms Applications Exam		
Assessment methods	Informative lectures Discussion Work with computers at laboratories project work written exam		
Recommended readings	1. John Sharp, Microsoft Visual C# 2012 Step by Step, 2013 2. Karli Watson, Jacob Vibe Hammer, Jon Reid, Morgan Skinner, Daniel Kemper, Christian Nagel, Beginning Visual C# 2012 Programming, 2012		
Knowledge	After the course the student will know the c# syntax and will be able to define object-oriented programming principles in the context of c# After the course the student will be able to explain what is happening in a c# code.		
Skills	The student will be able to write program in a c# language.		

Course title	C++ programming language		
Level of course	first cycle		
Teaching method	laboratory course / lecture		
Person responsible for the course	Agnieszka Konys	**E-mail address to the person**	Agnieszka.Konys@zut.edu.pl
Course code (if applicable)	WI-1-C++	**ECTS points**	4
Semester	winter/summer	**Language of instruction**	english
Hours per week	4	**Hours per semester**	60
Objectives of the course	Familiar with the syntax, basic programming constructs and principles used in C++ language The ability to write small-scale C++ programs using the acquired skills		
Entry requirements	None		
Course contents	Introduction to C++ and IDE Variables, datatypes and operators Input/output operations Conditionals Loops Arrays Structures Functions Input/output with files Introduction to programming and C++ Structure of a program and basic concepts Variables and fundamental data types Input/output operations Constants and operators Conditionals and loops Arrays and multi-dimensional arrays Structures Functions Exam		
Assessment methods	Informative lectures Discussion Work with computers at laboratories Written exam Continuous assessment		
Recommended readings	1. Bjarne Stroustrup, The C++ Programming Language (Fourth Edition), Addison-Wesley, 2012 2. Daoqi Yang, C++ and Object-Oriented Numeric Computing for Scientists and Engineers, Springer, 2001 3. http://www.cplusplus.com/doc/tutorial/		
Knowledge	After the course the student should be able to understand and use the basic programming constructs of C++ and write small-scale C++ programs using the above skills After the course the student should be able to explain what is happening in a C++ code		
Skills	After the course the student should be able to write small-scale C++ programs using the above skills. The student is able to design and implement an algorithm from scratch as a program in C ++ and is able to properly use various programming libraries to create an effective application.		
Other social competences	The student will acquire the following attitudes: creativity in creating programs, understanding the code and the ability to use technical documentation of C++ programming language.		

Course title	Compilers		
Level of course	first cycle		
Teaching method	laboratory course / lecture		
Person responsible for the course	Włodzimierz Bielecki	**E-mail address to the person**	Wlodzimierz.Bielecki@zut.edu.pl
Course code (if applicable)	WI-1-COM	**ECTS points**	4
Semester	winter/summer	**Language of instruction**	english
Hours per week	4	**Hours per semester**	60
Objectives of the course	To be able to: build lexical analyzers and use them in the construction of parsers; express the grammar of a programming language; build syntax analyzers and use them in the construction of parsers; perform the operations of semantic analysis; build a code generator; discuss the merits of different optimization schemes.		
Entry requirements	You are expected to have some basic programming skills using C, or C++ or java.		
Course contents	Define the simple computer architecture and programming language of this computer Implementation of a lexical analyzer for a defined programming language using the FLEX tool Implementation of the parser for the defined language using the BISON tool Implementation of defined semantic actions Implementation of the code generator for arithmetic expressions for the defined computer architecture Code generation for conditional statements and loops Implementation of the use of single- and multi-dimensional tables Implementation of the code generator for various data types Implementation of the code generator for various data types 3 Compiler structure Lexical analysis Top down parsing Bottom up parsing Lex and Yacc Semantic analysis Code generation, SPIM A simple translator Implementation of function calls		
Assessment methods	Informative / conversational lectures Laboratory exercises Assessment of the degree of practical tasks at the end of each laboratory the Final exam by checking the learning outcomes: presenting questions and assessing the answers		
Recommended readings	1. A.V. Aho, R. Sethi and J.D. Ullman, Compilers - Principles, Techniques, and Tools', Addison-Wesley, Boston, 2007		
Knowledge	The student has basic knowledge in the field of compiler design		
Skills	The student is able to design a simple compiler.		
Other social competences	The student is able to work with colleagues in a group.		

Course title	Computer Games Programming		
Level of course	first cycle		
Teaching method	project course / lecture		
Person responsible for the course	Radosław Mantiuk	**E-mail address to the person**	Radoslaw.Mantiuk@zut.edu.pl
Course code (if applicable)	WI-1-CGP	**ECTS points**	4
Semester	summer	**Language of instruction**	english
Hours per week	5	**Hours per semester**	75
Objectives of the course	Gaining knowledge, skills, and competences on the computer games programming.		
Entry requirements	Programming skills in C/C++ languages.		
Course contents	Implementation of a project involving the implementation of the basic computer game. Introduction to graphic libraries. Geometric transformations. User interface and time synchronisation. Game loop architecture. Aggregated game board. Collision detection. Lights and materials. Materials and texture.		
Assessment methods	Lectures Workshops Finished project (impemented computer game).		
Recommended readings	1. Michael Dawson, Beginning C++ Through Game Programming, Cengage Learning PTR, 2010, 3		
Knowledge	Gaining knowledge on computer games programming.		
Skills	Gaining skills in computer games programming.		
Other social competences	Gaining competences in computer games programming.		

Course title	Computer Networks		
Level of course	first cycle		
Teaching method	laboratory course / lecture		
Person responsible for the course	Remigiusz Olejnik	**E-mail address to the person**	Remigiusz.Olejnik@zut.edu.pl
Course code (if applicable)	WI-1-CTN	**ECTS points**	4
Semester	winter/summer	**Language of instruction**	english
Hours per week	4	**Hours per semester**	60
Objectives of the course	Knowledge of reference models, network standards, protocols of data link layer, network, transport and application layers. Knowledge of current wired and wireless network solutions. Ability of network's performance evaluation. Ability of simple home/office network building. Basic algorithms of data link, network and application layer implementation ability.		
Entry requirements	Basics of programming; Architecture of computer systems; Operating systems fundamentals.		
Course contents	Implementation of the program implementing the CRC algorithm. Implementation of the program implementing the routing algorithm selected. Implementation of the program implementing selected network application (eg. chat, file transfer, etc.) Introduction to simulation of computer networks. Building of a simulation model for a simple network. Introduction to computer networks. Physical layer, transmission media, multiplexing techniques, circuit and packet switching. Data link layer, error detection, flow control, ALOHA and CSMA protocols, protocols without collisions, Ethernet, wireless local area networks, interconnecting. Network layer, routing algorithms and protocols, quality of service, Internet Protocol. Transport layer, protocols, addressing, flow control, UDP, TCP and RTP protocols, Nagle's and Clarke's algorithms. Application layer, DNS, e-mail, WWW, multimedia applications of the networks.		
Assessment methods	Lecture with presentation Laboratory work Lecture - written exam Laboratory work - written reports Laboratory work - evaluation of submitted programs and project		
Recommended readings	1. A. S. Tanenbaum, Sieci komputerowe, Helion, Gliwice, 2004 2. M. Hassan, R. Jain, Wysoko wydajne sieci TCP/IP, Helion, Gliwice, 2004		
Knowledge	Student will gain detailed knowledge of network technologies		
Skills	Student is capable of running simulation package specialized in computer networks Student is able to prepare programs implementing selected networking aspects		

Course title	Computer System Architecture		
Level of course	first cycle		
Teaching method	laboratory course / lecture		
Person responsible for the course	Mariusz Kapruziak	**E-mail address to the person**	Mariusz.Kapruziak@zut.edu.pl
Course code (if applicable)	WI-1-CSA	**ECTS points**	4
Semester	winter/summer	**Language of instruction**	english
Hours per week	4	**Hours per semester**	60
Objectives of the course	Processor programming on different architectures. Knowledge of history and concepts of current processor and computer design.		
Entry requirements	Digital design. Basics of Electronics.		
Course contents	PC Mainboard. Assembler language for x86 processor - native program. Assembler for x86 - stack and mixing C and assembler. Communication port programming (Visual Studio). Sound card programming. Camera programming. Robot control on PC (programming). ARM processor programming FPGA programming (as an alternative to von Neumann processor). Project. SSE and vector units. Von Neumann machine and history of computer architectures. Execution and control unit functionality (on example of x86 and PIC architecture). Memory hierarchy and cache memory (its influence on efforts on program code optimization in particular) ARM architecture and low power designs (like palmtops, smartphones) Protected mode and its influence on modern operation systems, driver design for MS Windows and Linux systems Instruction Level Paralellism (especially superscalar and VLIW/DSP architectures) Modern PC microprocessors Supercomputers and networks of computers aimed to solve particular problems Reconfigurable systems and modern alternatives to von Neumann machines.		
Assessment methods	Lectures Laboratories Project Laboratories project. Laboratory raports. Exam.		
Recommended readings	1. W. Stallings, Computer Organization and Architecture, Prentice Hall, 2003 2. J. Stokes, Inside the Machine, No Starch Press 3. J. Silc, B. Robic, T Ungerer, Processor Architecture From Dataflow to Superscalar and Beyond, Springer Verlag, 1999 4. K. Kaspersky, Code Optimization: Effective Memory Usage, A-List Publishing		
Knowledge	Student knows fundamental processor structures and can describe them.		
Skills	Student can programm basic codes in the assembler language. Student can program code for basic peripheral devices.		

Course title	Computer Vision for Video Surveillance		
Level of course	first cycle		
Teaching method	laboratory course / lecture		
Person responsible for the course	Adam Nowosielski	**E-mail address to the person**	Adam.Nowosielski@zut.edu.pl
Course code (if applicable)	WI-1-CVS	**ECTS points**	3
Semester	winter/summer	**Language of instruction**	english
Hours per week	2	**Hours per semester**	30
Objectives of the course	The main objective of the course is to familiarize students with the range of possibilities and principles of the modern intelligent monitoring systems. Students will be prepared to design intelligent surveillance system performing the tasks of automatic processing, analysis and recognition of digital images.		
Entry requirements	Elementary digital image processing Elementary numerical recipes Elementary programming skills Elementary matrix algebra		
Course contents	Introduction to laboratory classes. Video surveillance at the Faculty and on the campus. The ALPR system. Image acquisition from cameras. Performance verification of available (ready to use, implemented) algorithms for video surveillance, e.g.: background modelling, object detection, object recognition, object tracking Implementation of selected algorithms for video surveillance, e.g.: background modelling, object detection, object recognition, object tracking. Development of a concept of simple video surveillance system. Definition of the scope of the project. Design and implementation of own simple video surveillance system. Introduction to video surveillance systems. Selected issues and classification of monitoring systems. Legal regulations. Systems of video-observation. Hardware in video monitoring systems. Intelligent Building. Intelligent cameras. Mobile wireless platforms. Access control controllers. Thermal imaging for video observation. Intelligent Transport Systems (ITS): ALPR, WIM, HIM, red-light, others. Measuring traffic congestion. Intelligent parking. Background modeling methods. Autoamtic detection and recognition of objects in video surveilance systems. Tracking algorithms. Example implementations of intelligent video surveillance systems: vehicle traffic measurement systems, human traffic analysis, people identification based on biometric features, etc.		
Assessment methods	Lectures: informative, problem solving, conversational Laboratory classes with a computer Problems discution at laboratory classes Discussion of the individual project, brainstorm Assessment of the project created during practical exercises and discussion of the final repot. Presentation and defense of the project in front of a group of students. Progress monitoring in implementation of own video surveillance system. Verification of reports from selected laboratories.		
Recommended readings	1. H. Kruegle, CCTV Surveillance, Second Edition: Video Practices and Technology, Butterworth-Heinemann, 2006, 672 p. 2. R. Gonzalez, R. Woods, S. L. Eddins, Digital Image Processing Using MATLAB 2nd Ed., Gatesmark Publishing, 2009, 827 p. 3. Richard Szeliski, Computer Vision: Algorithms and Applications, Springer, London, 2011		
Knowledge	Students are familiarized with the computer vision methods applicable to video surveillance. Students are acquainted with principles of the modern intelligent monitoring systems.		
Skills	Students are prepared to design intelligent surveillance system performing the tasks of automatic processing, analysis and recognition of digital images.		
Other social competences	The student is aware of the role of video surveillance systems for the society.		

Course title	Database systems		
Level of course	first cycle		
Teaching method	laboratory course / lecture		
Person responsible for the course	Przemysław Korytkowski	**E-mail address to the person**	Przemyslaw.Korytkowski@zut.edu.pl
Course code (if applicable)	WI-1-DSY	**ECTS points**	5
Semester	winter/summer	**Language of instruction**	english
Hours per week	4	**Hours per semester**	60
Objectives of the course	Design of relational databases SQL language proficiency Practical knowledge of MS SQL Server.		
Entry requirements	No requirements		
Course contents	ERD diagrams. Database schema modelling. SQL - data definition language: CREATE DABABASE, CREATE TABLE, ALTER TABLE, INSERT, UPDATE, DELETE, TRUNCATE, DROP TABLE. SQL - data manipulation language: SELECT, WHERE, GROUP BY, ORDER BY, HAVING SQL: data manipulation language: JOINS, subqueries. Indexes, query execution planning, EXPLAIN eXtensible Markup Language Privileges Relational model of data. Database management system Entity Relationship Diagrams. Relational database modelling. Structured Query Language (SQL) Normal forms and functional dependencies. Transactions, ACID, logging, concurrency, conflict seriazability, locking, deadlocks. I/O model and indexing Joins: nested loop join, block nested loop join, index nested loop join, sort-merge join, hash join. Relational algebra and query optimization. eXtensible Markup Language (XML) Database security: discretionary access control, role-based access control, mandatory access control. SQL injections.		
Assessment methods	Informative lectures Written exam		
Recommended readings	1. Garcia-Molina, Ullman, Widom, Database Systems. The complete book, Pearson, Upper Saddle River, 2009 2. Ramez Elmasri, Shamkant B. Navathe, Fundamentals of Database Systems, Pearson, Boston, 2016, 7		
Knowledge	Student is able to describe various types of databases. Student is able to explain query optimization process in BDMS.		
Skills	Student is able to design a database. Student is able to freely create SQL code.		

Course title	Data Mining Algorithms		
Level of course	first cycle		
Teaching method	laboratory course / lecture		
Person responsible for the course	Przemysław Klęsk	**E-mail address to the person**	pklesk@wi.zut.edu.pl
Course code (if applicable)	WI-1-DMA	**ECTS points**	4
Semester	winter/summer	**Language of instruction**	english
Hours per week	3	**Hours per semester**	45
Objectives of the course	Building the understanding about learning from data. Familiarization with probabilistic, tree-based, and boosted classifiers, and the related algorithms. Familiarization with rules mining and related algorithms.		
Entry requirements	mathematics programming algorithms and data structures		
Course contents	Programming the naive Bayes classifier (MATLAB) - for 'wine data set' (in class) and a selected data set (homework). Programming the Apriori algorithm - mining association rules. Programming an exhaustive generator of decision rules (for given premise length). Programming the CART algorithm - building a complete tree. Programming heuristics for pruning CART trees. Review of some elements of probability calculus. Derivation of Naive Bayes classifier. Remarks on computational complexity with and without the naive assumption. Bayes rule. LaPlace correction. Beta distributions. Mining association rules by means of Apriori algorithm. Support and confidence measures. Finding frequent sets (induction). Rules generation mechanics. Remarks on the hashmap data structure applied for Apriori algorithm. Pareto-optimal rules. Remarks on decision rules generation. Decision trees and CART algorithm. Impurity functions and their properties. Best splits as minimizers of expected impurity of children nodes. CART greedy algorithm. Tree pruning heuristics (by depth, by penalizing number of leafs). Recursions for traversing the subtrees (greedy and exhaustive). Ensemble methods: bagging and boosting (meta classifiers). AdaBoost algorithm. Exponential criterion vs zero-one-loss function. Real boost algorithm. Exam.		
Assessment methods	Lectures. Computer programming. Four short tests (15 minutes long) at the end of each topic during the lab. Four grades for the programs written as homeworks. Final grade for the lab calculated as a weighted mean from partial grades: - tests (weight: 40%), - programs (weight: 60%). Final grade for lectures from the test (2 h).		
Recommended readings	1. M. J. Zaki, W. Meira Jr, "Data Mining and Analysis - Fundamental Concepts and Algorithms", Cambridge University Press, 2014 2. P. Klęsk, Electronic materials for the course available at: http://wikizmsi.zut.edu.pl, 2015		
Knowledge	Student has an elementary knowledge on data mining algorithms and notions.		
Skills	Student can implement (MATLAB or Python) data mining algorithms presented during lectures.		

Course title	Data Warehousing and Big Data		
Level of course	first cycle		
Teaching method	laboratory course / lecture		
Person responsible for the course	Przemysław Różewski	**E-mail address to the person**	Przemyslaw.Rozewski@zut.edu.pl
Course code (if applicable)	WI-1-DWB	**ECTS points**	4
Semester	winter/summer	**Language of instruction**	english
Hours per week	4	**Hours per semester**	60
Objectives of the course	Be able to design Data Warehouse and use MDX effectively.		
Entry requirements	SQL basics, basic understanding of main business processes		
Course contents	Conceptual and Logical Data Warehouse Design ETL process design SQL Server Integration Services SQL Server Analysis Services (SSAS) Power BI Data Warehouse Concepts Conceptual and Logical Data Warehouse Design Querying Data Warehouses (MDX) Extraction, Transformation, and Loading (ETL) Working with Big Data Integration of Big Data and Data Warehousing New Data Warehouse Technologies (Spatial, Trajectory, Semantic Web)		
Assessment methods	Informative lectures Written exam Project		
Recommended readings	1. Alejandro Vaisman Esteban Zimányi, Data Warehouse Systems Design and Implementation, Springer-Verlag Berlin Heidelberg, 2013, DOI: 10.1007/978-3-642-54655-6		
Knowledge	Student will know how to integrate the Big Data and Data Warehousing.		
Skills	Student is able to design and querying Data Warehouse.		

Course title	Deep learning for visual computing		
Level of course	first cycle		
Teaching method	laboratory course / project course / lecture		
Person responsible for the course	Radosław Mantiuk	**E-mail address to the person**	Radoslaw.Mantiuk@zut.edu.pl
Course code (if applicable)	WI-1-DLV	**ECTS points**	5
Semester	winter/summer	**Language of instruction**	english
Hours per week	4	**Hours per semester**	60
Objectives of the course	Gaining knowledge, skills, and competences about CNNs for visual computing		
Entry requirements	Programming skills in a scripting language (Phyton).		
Course contents	Introduction to CNN toolkit. Input data acquisition task. Learning and validation of CNN. Cross-validation example. Calibration of the network. Implementation of a project involving the acquisition of input data and learning CNN for identification of objects in images. Introduction to convolutional neural networks (CNN). CNN toolkits. Input data acquisition. Tutorial: solving basic object classification problem. Learning CNN with cross-validation.		
Assessment methods	Lecture Workshops Finished project on detection task using CNN.		
Recommended readings	1. Ragav Venkatesan, Baoxin Li, Convolutional Neural Networks in Visual Computing: A Concise Guide, CRC Press, 2017		
Knowledge	Gaining basic knowledge on CNNs for visual computing.		
Skills	Gaining skills on training CNNs.		
Other social competences	Gaining competence in training CNNs.		

Course title	Digital Circuits		
Level of course	first cycle		
Teaching method	laboratory course / lecture		
Person responsible for the course	Mirosław Łazoryszczak	**E-mail address to the person**	Miroslaw.Lazoryszczak@zut.edu.pl
Course code (if applicable)	WI-1-DCI	**ECTS points**	4
Semester	winter/summer	**Language of instruction**	english
Hours per week	4	**Hours per semester**	60
Objectives of the course	Practical skills in basic digital circuits modeling using VHDL		
Entry requirements	Boolean algebra fundamentals		
Course contents	Laboratory rules and equipment Software tools introduction Digital logic gates and boolean functions simplicfication Combinatorial logic and programable devices Discrete flip-flops principle of operation Sequential circuits design VHDL based sequential circuits design Selected peripherals handling: LEDs, buttons, displays, connectivity etc. Designing a fully functional digital system Hardware design modeling. Hardware descriprion languages. Introduction to VHDL. Base VHDL syntax. Simulation and synthesis constructs. Combinatorial logic. Functional Blocks. Enabling. Decoding. Multiplexer-based combinational circuits. Adder. Subtractor. VHDL models of combinatorial circuits. Combinatorial logic design. Sequential logic definitions. Latches. State tables and diagrams. Sequential circuits analysis. Sequential circuits design. Integrated cirquits technology. Parameters. Programmable devices. Registers and transfer operations. Microoperations (arithmetic, logic and shift). Control of register transfers. Memory. Static and dynamic RAM. Asynchronous and synchronous DRAM. Computer designs basics Assessment		
Assessment methods	Lecture with presentations Self-performed laboratory tasks Written exam Reports evaluation		
Recommended readings	1. Mano M.M.R., Kime Ch.R., Martin T., Logic & Computer Design Fundamentals, 5th Edition, Pearson, 2016 2. Mano M.M.R, Ciletti M.D., Digital Design: With an Introduction to the Verilog HDL, VHDL, and SystemVerilog, Pearson, 2018, 6		
Knowledge	The student knows the structure and rules of operation of basic digital circuits: logical and sequential, knows the principles of simple design circuits using hardware description language.		
Skills	The student can builld basic digital circuits: logical and sequential, and implement simple circuits using hardware description language.		

Course title	Dynamic documents and front-end Web development		
Level of course	first cycle		
Teaching method	laboratory course / lecture		
Person responsible for the course	Jarosław Jankowski	**E-mail address to the person**	Jaroslaw.Jankowski@zut.edu.pl
Course code (if applicable)	WI-1-DDO	**ECTS points**	3
Semester	winter	**Language of instruction**	english
Hours per week	3	**Hours per semester**	45
Objectives of the course	Understanding selected programming languages and data processing methods in dynamic Web systems.		
Entry requirements	The basics of HTML language		
Course contents	Dynamic access to web page elements in object-oriented document model Dynamic modification of Web content Formatting content using CSS sheets Capturing events using Java Script Construction of validators and forms Encoding data using XML language Integration of selected components and construction of asynchronous applications Use of selected libraries in dynamic document programming Document object model CSS sheets Application of Java Script in dynamic documents XML markup language AJAX and asynchronous processing Selected applications and libraries integrating these technologies		
Assessment methods	Lecture with presentations and examples Practical exercises in laboratories Lecture - Written examin with practical questions, questions in the form of a selection and description - a total of 10 questions Overall assessment based on reports and attendance		
Recommended readings	1. Bogdan Brinzarea, AJAX and PHP: Building Modern Web Applications, PACKT, London, 2012 2. Anne Boehm, Zak Ruvalcaba, HTML5 and CSS3, Murach, NY, 2015		
Knowledge	Wiedza w zakresie programowania dokumentów dynamicznych w systemach internetowych		
Skills	Umiejętność programowania dokumentów dynamicznych z wykorzystaniem wiodących technologii		
Other social competences	Kompenetecje z zakresie programowania dokumentów dynamicznych i pracy zespołowej		

Course title	E-commerce and online marketing technologies		
Level of course	first cycle		
Teaching method	laboratory course / lecture		
Person responsible for the course	Jarosław Jankowski	**E-mail address to the person**	Jaroslaw.Jankowski@zut.edu.pl
Course code (if applicable)	WI-1-ECO	**ECTS points**	3
Semester	winter	**Language of instruction**	english
Hours per week	3	**Hours per semester**	45
Objectives of the course	Increasing the knowledge in the area of technologies used in electronic marketing Increasing the knowledge in the area of strategies in electronic marketing		
Entry requirements	The basics of HTML language		
Course contents	Use of selected behavioral analysis systems for web site users Configuration and planning of ad campaigns using ad servers Use of contextual advertising and search engines Search engine positioning Use of selected social media platforms and social network analysis in marketing Modeling diffusion of marketing messages in social networks Use of e-commerce platforms and recommendation systems Application of selected methods of extraction of knowledge in electronic marketing Communication models in electronic marketing Performance measurement and optimization of advertising campaigns Marketing in social media Search engine marketing Marketing and email communication Electronic commerce platforms and recommendation algorithms Multivariate optimization and maximization of conversions		
Assessment methods	Lecture with presentations and examples. Laboratory exercises and practical tasks. Lecture: summary assessment. Written examination with practical questions, questions in the form of a choice and description. Laboratories: assessment based on reports and attendance.		
Recommended readings	1. Kenneth C. Laudon, Carol Guercio Traver, E-Commerce, Pearson, NY, 2017 2. Rob Stokes, eMarketing: The essential guide to marketing in a digital world, QUIRK, London, 2014		
Knowledge	Wiedza w zakresie wdrażania i eksploatacji systemów marketingu elektronicznego. Wiedza w zakresie metod analitycznych, przetwarzania danych i algorytmów wykorzystywanych w systemach marketingu elektronicznego.		
Skills	Umiejętność wdrażania i eksploatacji systemów marketingu elektronicznego. Posiada umiejętność stosowania metod analitycznych i algorytmów przetwarzania danych wykorzystywanych w systemach marketingu elektronicznego.		
Other social competences	Kompetencje w zakresie wdrażania i eksploatacji systemów marketingu elektronicznego. Komepetencje w zakresie zastosowań metod analitycznych w systemach marketingu elektronicznego.		

Course title	EEG signal analysis in Matlab		
Level of course	first cycle		
Teaching method	laboratory course / lecture		
Person responsible for the course	Izabela Rejer	**E-mail address to the person**	irejer@wi.zut.edu.pl
Course code (if applicable)	WI-1-EEG	**ECTS points**	4
Semester	winter/summer	**Language of instruction**	english
Hours per week	4	**Hours per semester**	60
Objectives of the course	To teach students how to record, process and analyze EEG signals in Matlab environments.		
Entry requirements	None		
Course contents	Introduction to Matlab programming OpenVibe platform Sending data from OpenVibe to Matlab Recording EEG signals with 19-channel Discovery 20 device Removing artifacts from EEG signal Spatial and temporal filtering Extracting different brain activity patterns from EEG recording Exam. EEG signals - main characteristics Main types of artifacts and methods for removing them Spectral analysis of EEG signal (Fourier transform) Extracting different brain activity patterns from EEG recording Exam.		
Assessment methods	Informative lectures. Discussion. Laboratories with computers and EEG devices. The final report describing the detailed results of the analysis of the EEG signal acquired durings laboratories and processed in Matlab environment. The final discussion summing up the knowlegde gained during the lectures.		
Recommended readings	1. Lotte F., Study of Electroencephalographic Signal Processing and Classification Techniques towards the use of Brain-Computer Interfaces in Virtual Reality Applications, 2008, PhD Thesis, https://sites.google.com/site/fabienlotte/phdthesis 2. S. W. Smith, Digital Signal Processing: A practical Guide for Engineers and Scientists, 2003 3. Official Matlab site: http://www.mathworks.com/help/matlab/		
Knowledge	After the lectures the student will be able to: define a BCI, describe the main problems with EEG data, describe the EEG device, descibe different BCI paradigms, choose the processing methods suitable for different paradigms and different EEG data.		
Skills	The student will be able to create the project of a BCI suitable for a given task.		

Course title	Embedded systems		
Level of course	first cycle		
Teaching method	laboratory course / lecture		
Person responsible for the course	Mirosław Łazoryszczak	**E-mail address to the person**	Miroslaw.Lazoryszczak@zut.edu.pl
Course code (if applicable)	WI-1-EMS	**ECTS points**	4
Semester	winter/summer	**Language of instruction**	polish
Hours per week	4	**Hours per semester**	60
Objectives of the course	The ability to classify, describe and build microcontroller based embedded systems		
Entry requirements	Computer systems architecture Programming basics		
Course contents	Arduino as a popular embedded system. Selected application for Arduino board. AVR microcontroller family. Develpment environment and assembler in embedded systems AVR microcontroller family. Introduction to C programming using selected microcontroller platform. LEDs and LED display handling Switches, keyboard and debouncing. ARM Cortex-M family. Toolchain. Programming using selected evaluation boards using available peripherals (displays, audio, networks etc.) Implementing RTOS components. Building own system using peripheral modules like UART, LCD display, a/c and c/a converters, audio input/output etc. Reconfigurable embedded systems and soft processors. Introduction to embedded systems: real time issues, power consumptions, software architectures. Popular microcontroler families and their architectures (e.g. AVR, ARM) Main peripheral modules used in microcontrollers (timer/counter, UART, interrupt controller, ADC, etc.) Selected input/output devices (displays, keyboards, a/c and c/a converters, motors, sensors) and communication interfaces. Buses used in embedded systems (SPI, I2C, I2S, 1W) Embedded operating systems. Selected RTOSes. Operation principles. Programming examples. Reconfigurable devices in embedded control and compputing. Exam		
Assessment methods	Lecture with presentations Laboratory Written exam Lab reports		
Recommended readings	1. Joseph Yiu, The Definitive Guide to ARM Cortex-M3 and Cortex-M4 Processors, Elsevier, 2014 2. Edward A. Lee, Sanjit A. Seshia, Introduction to embedded systems. A cyber-physical systems approach., MIT Press, 2017 3. Microcontroller vendors, Documentation of selected microcontrollers, 2011		
Knowledge	The students is able to describe, classify and analyze embedded systems based on selected microcontrollers with or without operating systems.		
Skills	The student can implement and build simple embedded systems due to the functional requirements.		

Course title	Essentials of Fuzzy Logic		
Level of course	first cycle		
Teaching method	laboratory course / lecture		
Person responsible for the course	Marcin Pluciński	**E-mail address to the person**	Marcin.Plucinski@zut.edu.pl
Course code (if applicable)	WI-1-EFL	**ECTS points**	4
Semester	winter/summer	**Language of instruction**	english
Hours per week	4	**Hours per semester**	60
Objectives of the course	Acquirement of competence and practice in construction of fuzzy models of systems, fuzzy calculations and fuzzy control of plants		
Entry requirements	Basic knowledge of high mathematics		
Course contents	Discovering by student fuzzy phenomena, fuzzy variables, fuzzy notions in the world Constructing membership functions for own detected uncertain values from science, technique, medicine, economics, biology etc. Describing membership fumctions by mathematical formulas. Trasformation of vertical membership functions into horizontal functions. Constructing rule bases for real systems and checking their logical consistence Training in fuzzyfication, rule premises evaluation, conclusion activation of individual rules, aggregation of individual rule conclusions in one resulting conclusion of the rule base and its defuzzification Constructing fuzzy models for real systems. Calculation of the fuzzy model output for given values of its inputs for models of various dimensionality Constructing neuro-fuzzy networks for a given fuzzy model Constructing expert fuzzy controllers for a given real plant Constructing fuzzy controllers on the basis of a plant-model Diffrence between classical and fuzzy logic. Examples of fuzziness in the real world. Necessity of fuzziness use. Short history of fuzzy logic development. Mathematical models of fuzzy linguistic and numerical evaluations : membership functions. Examples of membership functions. Vertical and horizontal models of membership functions. Identification of membership functions by experts. Typical errors made during the identification. Classical (vertical) and new, horizontal fuzzy arithmetic. Transformation of vertical in horizontal membership functions. Examples of calculations. Granular arithmetic and mathematics. Fuzzy models of systems. Components of fuzzy models: fuzzification, premise evaluation, determination of activated membership functions of paricular rules, determining of the resulting membership function of the rule base and its defuzzification. Constructing fuzzy models for chosen real problems and calculating model ouputs for give model inputs. Neuro-fuzzy networks as self-learning fuzzy models. Fuzzy control and its structure. Classic, expert fuzzy control and control based on the model of the controlled plant. Examples of real applications of fuzzy logic.		
Assessment methods	Informational lecture with presentations Laboratory training in individual solving of problems delivered by an academition Lectures: summarizing evaluation of knowledge assimilated by student in form of an exam and of evaluation the student activity shown during lectures Laboratory: forming evaluation of the student based on the student activity and ability shown at solving problems given by an academician		
Recommended readings	1. Andrzej Piegat, Fuzzy modeling and control, Physica-Verlag, A Springer-Verlag Company, 2001, 1 2. Witold Pedrycz, Fernando Gomide, Fuzzy systems engineering, Wiley-Interscience, Hoboken, New Jersey, USA, 2007, 1		
Knowledge	The student has knowledge about fuzzy sets, fuzzy modelling and their practical applications.		
Skills	The student has the ability to analyse fuzzy models work, to create them for chosen real problems, and to use them in control systems.		

Course title	Expert systems		
Level of course	first cycle		
Teaching method	laboratory course / lecture		
Person responsible for the course	Joanna Kołodziejczyk	**E-mail address to the person**	Joanna.Kolodziejczyk@zut.edu.pl
Course code (if applicable)	WI-1-ESY	**ECTS points**	3
Semester	winter/summer	**Language of instruction**	english
Hours per week	3	**Hours per semester**	45
Objectives of the course	To learn the basic knowledge in expert systems. Student will have the ability to recognize areas of implementation. Students will be able to design, build and implement rule-based expert systems.		
Entry requirements	Algorithms and data structures		
Course contents	CLIPS - installing and dealing with facts Rules constract in CLIPS Excerises with simple user interface communication in CLIPS Functions and advanced CLIPS programming Project in CLIPS From CLIPS to JESS History of Expert Systems. The begining, early solutions. Fomal representation of knowladge in expert systems. Dealing with uncertainty. Propositional logic as a method of knowladge representation. First predicate logic. Prolog programming language. Uncetrainty - probablistic view. Bayes theorem and bayesian networks. Fuzzy expert systems. Expert systems based on certainty factor.		
Assessment methods	Presentation, lecture Discussion durig lecture. Developing software in CLIPS Test checking the knowledge on expert systems Short programming tasks in CLIPS Programming project - make your own expert system		
Recommended readings	1. Russel S., Norvig P, Artificial Intelligence A modern approach, Prentice Hall, 2003 2. Clips online documentation, 2016		
Knowledge	Student understand a structure of the expert system. Has a knowladge on representation forms and how the uncertatinty could be represented. Can name and explain how well-known expert systems work.		
Skills	Students has the ability to develop expert systems in CLIPS and JESS.		

Course title	F# Programming Language		
Level of course	first cycle		
Teaching method	laboratory course / lecture		
Person responsible for the course	Marcin Pietrzykowski	**E-mail address to the person**	Marcin.Pietrzykowski@zut.edu.pl
Course code (if applicable)	WI-1-FPL	**ECTS points**	3
Semester	winter/summer	**Language of instruction**	english
Hours per week	3	**Hours per semester**	45
Objectives of the course	Familiar with the sytnax, structures and principles used in the f# language The ability to develop a program in f# language.		
Entry requirements	None		
Course contents	Introduction to visual Studio IDE and F# Declaring values and functions, pattern matching basics Recursive and higher order functions Option types, tuples and records Lists and sequences Sets, maps and discriminated unions Control flows Arrays Mutable data and mutable collections I/O operations Classes and operator overloding Inheritance and interfaces F# advanced Introduction to: Functional Programming and F# programming language Working With Functions Immutable Data Structures Imperative Programming Object Oriented Programming F# Advanced Exam		
Assessment methods	Informative lectures Discussion Work with computers at laboratories project work written exam		
Recommended readings	1. Robert Pickering, Beginning F#, 2009 2. Don Syme, Adam Granicz, Antonio Cisternino, Expert F#, 2007		
Knowledge	After the lecture the student will know the f# syntax and will be able to define programming concepts used in the f# language. After the lecture the student will be able to explain what is happening in a f# code.		
Skills	The student will be able to write program in a f# language.		

Course title	FPGA Design and reconfigurable computing		
Level of course	first cycle		
Teaching method	laboratory course / lecture		
Person responsible for the course	Mariusz Kapruziak	**E-mail address to the person**	Mariusz.Kapruziak@zut.edu.pl
Course code (if applicable)	WI-1-FDC	**ECTS points**	3
Semester	winter/summer	**Language of instruction**	english
Hours per week	3	**Hours per semester**	45
Objectives of the course	FPGA programming in Verilog. Basics of VHDL. General knowledge of FPGA technology.		
Entry requirements	Digital design. Basics of electronics.		
Course contents	FPGA - basics of Verilog. FPGA - VGA display. FPGA - motor control + encoder. CPLD - low power programming. FPGA editor. FPGA - audio processing + DSP resources. Project. Picoblaze - soft processor. Basics of FPGA/CPLD devices architectures. Verilog language. VHDL language. SystemVerilog and TLM (Transaction Level Modeling) Synthesis methodology. Detailed FPGA structure of modern devices.		
Assessment methods	Lectures. Laboratories. Project Final Exam Laboratory reports. Project.		
Recommended readings	1. C.M. Maxfield, The Design Warrior’s Guide to FPGAs, Linacre House 2. S. Sutherland, S. Davidmann, P. Flake, SystemVerilog for Design, A Guide to Using SystemVerilog for Hardware Design and Modeling, Springer, 2011		
Knowledge	Student knows basics of HDL and RTL synthesis. Student knows structures of FPGA devices.		
Skills	Student is able to program in Verilog/VHDL.		

Course title	Fundamentals of Error-Correcting Block Codes		
Level of course	first cycle		
Teaching method	lecturing course / lecture		
Person responsible for the course	Dorota Majorkowska-Mech	**E-mail address to the person**	Dorota.Majorkowska-Mech@zut.edu.pl
Course code (if applicable)	WI-1-FEC	**ECTS points**	3
Semester	winter	**Language of instruction**	english
Hours per week	2	**Hours per semester**	30
Objectives of the course	Knowledge of error-correcting codes Skills in error-correcting codes construction		
Entry requirements	Basics of linear and abstract algebra.		
Course contents	Calculation of the minimum distance, detection and correction capability of line code codes. Examination of the properties of algebraic structures. Construction of extended Galois fields. Finding primitive elements of extended Galois fild, minimal polynomials and conjugates of elements. Linear block codes: matrix description, standard array, syndrome. Constructing of Hamming codes. Cyclic codes: polynomial and matrix description of cyclic codes, encoding, syndrome computation, error detection and decoding. Constructing some examples of cyclic codes. Written test. The discrete communication channel. Types of errors and types of error-correcting codes. Block codes, minimum distance, error-detecting and error-correcting capabilities of a block code. Algebraic structures: groups, rings, fields, vector spaces. Construction of extended Galois fields. Structure of extended Galois fields, primitive elements, minimal polynomials and conjugates. Linear block codes: matrix description, standard array, syndrome, Hamming codes, Hamming spheres and perfect codes. Cyclic codes: polynomial and matrix description of cyclic codes, encoding, syndrome computation, error detection and decoding. Important classes of cyclic codes. Written exam.		
Assessment methods	Lecture with presentations Solving problems on board (workshop) Written exam Written test		
Recommended readings	1. Richard E. Blahut, Algebraic Codes for Data Transmission, Cambridge University Press, New York, 2003		
Knowledge	Students has knowledge in error-correcting codes construction		
Skills	Students has skills in error-correcting codes construction		

Course title	Graphical User Interface in .NET		
Level of course	first cycle		
Teaching method	laboratory course / lecture		
Person responsible for the course	Marcin Pietrzykowski	**E-mail address to the person**	Marcin.Pietrzykowski@zut.edu.pl
Course code (if applicable)	WI-1-GUI	**ECTS points**	2
Semester	winter/summer	**Language of instruction**	english
Hours per week	2	**Hours per semester**	30
Objectives of the course	Familiar with Windows Forms and Windows Presentation Foundation The ability to develop Windows Form Application and Windows Presentation Foundation Application.		
Entry requirements	None		
Course contents	Introduction to Windows Forms Controls, Forms, Containers and Applications, Menus, Toolbars, Dialogs Settings, Resources Building Controls, Inheritance and Reuse, Property Grids, Data binding Introduction to Windows Presentation Foundation XAML Sizing, Positioning and Transforming Elements, Layout with Panels Input Events, Content Controls, Item Controls Image, Text, Other Controls, Resources, Data Binding Windows Forms Fundamentals Custom Controls Modern Controls Data Binding and Windows Forms Techniques Building a WPF Application WPF Controls Data Binding and Rich Media Exam		
Assessment methods	Informative lectures Discussion Work with computers at laboratories project work written exam		
Recommended readings	1. Chris Sells, Windows Forms Programming in C#, 2003 2. Matthew MacDonald, Pro .NET 2.0 Windows Forms and Custom Controls in C#, 2005 3. Adam Nathan, WPF 4.5 Unleashed, 2013		
Knowledge	After the course the student will possess knowledge about Windows Forms After the course the student will possess knowledge about Windows Presentation Foundation		
Skills	After the course students will be able to design and create Windows Form Application After the course students will be able to design and create Windows Presentation Foundation Application.		

Course title	Human-Computer Interaction		
Level of course	first cycle		
Teaching method	laboratory course / lecture		
Person responsible for the course	Adam Nowosielski	**E-mail address to the person**	Adam.Nowosielski@zut.edu.pl
Course code (if applicable)	WI-1-HCI	**ECTS points**	3
Semester	winter/summer	**Language of instruction**	english
Hours per week	2	**Hours per semester**	30
Objectives of the course	The main objective of the course is to familiarize students with the current trends in human-computer interaction. New approaches like touchless interaction as well as classical methods are discussed and analyzed during the course. Students are familiarized with the wide range of modern equipment, software and algorithms of human-computer interaction.		
Entry requirements	Elementary programming skills		
Course contents	Introduction to human-computer interaction. Improving everyday computing: mouse gestures, virtual assistants, etc. Detection and recognition of the user. Who is the user? – assessment of sex, age and emotional state. Touchless interaction: gestures recognition, hand operated interfaces, head operated interfaces, touchless text entry. Eyetracking - determining the areas of interest on the screen. Assistive technologies for user with disabilities. Introduction to human-computer interaction. Improving everyday computing: mouse gestures, virtual assistants, etc. Detection and recognition of the user. Who is the user? – assessment of sex, age and emotional state. Touchless interaction: gestures recognition, hand operated interfaces, head operated interfaces, touchless text entry. Eyetracking - determining the areas of interest on the screen. Assistive technologies for user with disabilities.		
Assessment methods	Lectures: informative, problem solving, conversational Laboratory classes with a computer Problems discution at laboratory classes Final grade based on continuous assessment of tasks carried out during the classes. Verification of reports from selected laboratories.		
Recommended readings	1. A. Dix, J. Finlay, G. D. Abowd, R. Beale, Human-Computer Interaction, Pearson, 2004, 3rd Edition 2. B. Shneiderman, C. Plaisant, Designing the User Interface: Strategies for Effective Human-Computer Interaction, Pearson Addison-Wesley, 2009, 5th Edition 3. D. K. Kumar, S. P. Arjunan, Human-Computer Interface Technologies for the Motor Impaired, CRC Press, 2015 4. Daniel Wigdor, Dennis Wixon, Brave NUI World: Designing Natural User Interfaces for Touch and Gesture, Morgan Kaufmann, 2011, 1st Edition		
Knowledge	Students are familarized with the current trends in human-computer interaction. They gain knowledge about new approaches like touchless interaction as well as classical methods.		
Skills	Students are familiarized with the wide range of modern equipment, software and algorithms of human-computer interaction.		
Other social competences	Student has the consciousness of building communication systems in the strict connection with a social group that is the addressee of the given solutions (culture, norms, status). Student is aware of the responsibility for the wrong interpretation of the communication message.		

Course title	Intelligent Decision Systems		
Level of course	first cycle		
Teaching method	laboratory course / lecture		
Person responsible for the course	Wojciech Sałabun	**E-mail address to the person**	wsalabun@wi.zut.edu.pl
Course code (if applicable)	WI-1-IDS	**ECTS points**	4
Semester	winter/summer	**Language of instruction**	english
Hours per week	4	**Hours per semester**	60
Objectives of the course	To provide the knowledge about multi-criteria decision-making methods which are used to solving decision problems To equip the students with the ability of solving decision problems by using MCDM methods		
Entry requirements	None		
Course contents	Intro to solving decision problems by using WSM and WPM methods Intro to solving decision problems by using TOPSIS methods Intro to solving decision problems by using AHP methods Intro to solving decision problems by using ELECTRE methods Intro to solving decision problems by using ANP methods Intro to solving decision problems by using Fuzzy Logic Exam Description of decision making problems (structure, elements etc.) Review of the MCDM methods (achievements and main directions of researches) The WSM and WPM methods (examples, application, benefits, defects, etc.) The AHP and ANP methods (examples, application, benefits, defects, etc.) The ELECTRE methods (examples, application, benefits, defects, etc.) The TOPSIS methods (examples, application, benefits, defects, etc.) The Fuzzy methods in decision-making (examples, application, benefits, defects, etc.) Exam		
Assessment methods	Informative lectures Discussion Laboratories with computers The discussion summing up the knowledge gained during the lectures Written exam		
Recommended readings	1. Scientific papers and materials provided by the lecturer		
Knowledge	After the lectures the student will be able to define a MCDM problem, describe main MCDM methods, and choose the method suitable for a decision problem		
Skills	The student will be able to choose MCDM method for a problem. The student will be able to solve a multi-criteria problem.		

Course title	Introduction to front-end application development		
Level of course	first cycle		
Teaching method	laboratory course / lecture		
Person responsible for the course	Bartłomiej Małachowski	**E-mail address to the person**	Bartlomiej.Malachowski@zut.edu.pl
Course code (if applicable)	WI-1-JSW	**ECTS points**	3
Semester	winter/summer	**Language of instruction**	english
Hours per week	3	**Hours per semester**	45
Objectives of the course	To be able to independently develop a simple Single Page Application in JavaScript with REST data-exchenge capabilities.		
Entry requirements	Basic knowledge on procedural and object-oriented programming. Good knowledge on HTML and CSS languages		
Course contents	Basic JS programming - control statemets, loops, functions, objects and prototypes Modification and dynamic building of documents with DOM API Using several basic web APIs - geolocation, canvas, media etc. Async HTTP requests with XMLHttpRequest API, JSON - serialization and parsing Implementation of simple REST webservice client Development of example Single Page Application, integration of external APIs (ex. Google Maps, Social media services etc.) Principles of JavaScript programming Document Object Model API Geo Location API and other pupular Web Javascript APIs Asynchronous HTTP requests with XMLHttpRequest API Development tools: dependency management, building and deployment Principles of RESTfull web services and JSON data format Single page applications - principles and development		
Assessment methods	Auditorial lectures Individual work - programming taks Evalution of developed programming tasks through code review made by the teacher Final exam		
Recommended readings			

Course title	Introduction to Natural Language Processing		
Level of course	first cycle		
Teaching method	laboratory course / lecture		
Person responsible for the course	Joanna Kołodziejczyk	**E-mail address to the person**	Joanna.Kolodziejczyk@zut.edu.pl
Course code (if applicable)	WI-1-NLP	**ECTS points**	4
Semester	winter/summer	**Language of instruction**	english
Hours per week	4	**Hours per semester**	60
Objectives of the course	To understand the methods used to solve practical problems of NLP, in particular, information retrieval summarization, machine translation To apply the existing NLP libraries, determine the advantages and disadvantages of different systems, evaluate and compare the results		
Entry requirements	The course does not require any previous knowledge. Python familiarity will be useful.		
Course contents	Python - accessing and processing text Python - text categorizing and tagging Text classification Extracting information from text Sentence analyzis Grammar analyzis Semantics analysis Text processing: regular expressions, tokenization, sentesces segmentation; n-gram language models Naïve bayes and logistics regression – text calssicication Lexical semantics, words as vectors, Artifiacl neural networks Tagging, Hidden Markov Models Recursive neural network Encoder- decoder networks, or sequence-to-sequence models Parsing Question Ansewring, Dialog, Chatbots		
Assessment methods	Lectures presentation Discussion Developing software in Python Testing of knowledge through a multiple choice test Continuous assessment Project work		
Recommended readings	1. Jurafsky, D., Martin, J., Speech and language processing: An introduction to speech recognition, computational linguistics and natural language processing, Prentice Hall, 2008 2. Bird, S., Klein, E., Loper, E, Natural language processing with Python, O'Reilly Media, Inc.,, 2009		
Knowledge	Student understand the basics of natural language processing (NLP). Has a knowladge on language modeling, text classification, summarization, and machine translation.		
Skills	Students will learn how to use existing NLP libraries and software packages but also the mathematical models underlying computational linguistics.		

Course title	Introduction to the Internet of Things		
Level of course	first cycle		
Teaching method	laboratory course / project course		
Person responsible for the course	Remigiusz Olejnik	**E-mail address to the person**	Remigiusz.Olejnik@zut.edu.pl
Course code (if applicable)	WI-1-ARD	**ECTS points**	4
Semester	winter/summer	**Language of instruction**	english
Hours per week	4	**Hours per semester**	60
Objectives of the course	To gain: 1. theoretical and practical skills in Arduino programming, 2. ability of advanced hardware projects preparation.		
Entry requirements	Basics of: C programming, electronics and computer systems architecture.		
Course contents	1. Introduction to Arduino, its hardware and software design, IDE. 2. The art of Arduino programming – sketch and its structure: setup(), loop(), comments; data types; variables; arithmetic, logical, conditional, relational, increment operators; constants; functions; flow control: if, if...else, for, while, do...while; arrays; strings; digital I/O; analog I/O; time; math; random; serial communication; libraries; PWM; interrupts; I2C; SPI; SD card; wired and wireless networking. 3. Detailed overview of all sensors that will be used during laboratory. 4. Examples built-in the IDE. Hello world! sketch. 5. Using of breadboard, resistors and LEDs, buttons, switches, digital inputs, analog inputs, digital outputs, PWM. 6. Light: LED, fading LED, 2-color LED, RGB LED, LED bar graph, 7-digits LED display, dot-matrix LED display, LCD display. 7. Sensors: humidity, temperature, pressure, raindrops, PIR, ultrasonic, sound, knock, vibration, photo resistor, tilt, infrared, Hall magnetic, rotary encoder, flame, joystick, metal touch, mercury switch, detection of gases, 3D accelerometer, obstacle avoidance IR, optical broken light, laser. 8. Outputs: motor control: DC motor, servo motor, stepper motor; relay module 9. Sound: tone library, microphone, buzzer, speaker. 10. Analog and digital inputs: reading analog voltage, external keyboard and mouse. 11. RFID module, SD storage, GPS receiver. 12. Ethernet shield, wireless communication. Implementation of selected problem: 1. Hardware design proposal. 2. Software implementation of the problem's solution. 3. Preparation of the project's documentation.		
Assessment methods	Laboratory work and project Laboratory – evaluation of the reports submitted after each class Project – evaluation of the final project, along with its documentation		
Recommended readings	1. Michael Margolis, Arduino cookbook, O'Reilly, 2013 2. John Boxall, Arduino workshop: a hands on introduction with 65 projects, No Starch Press, 2013 3. Arduino Home https://www.arduino.cc/		
Skills	Student will gain theoretical and practical skills in Arduino programming, along with ability of advanced hardware projects preparation		

Course title	Intro to Mathematical Programming		
Level of course	first cycle		
Teaching method	laboratory course / lecture		
Person responsible for the course	Wojciech Sałabun	**E-mail address to the person**	wsalabun@wi.zut.edu.pl
Course code (if applicable)	WI-1-IMP	**ECTS points**	4
Semester	winter/summer	**Language of instruction**	english
Hours per week	4	**Hours per semester**	60
Objectives of the course	The course introduces to techniques for solving optimization tasks based on mathematical programming methods		
Entry requirements	None		
Course contents	Linear programming: geometric method Linear programming: simplex algorithm Transportation theory: transport task Program Evaluation and Review Technique (PERT) Critical Path Method (CPM) Traveling salesman problem: computing a solution Exam Intro to linear programming Applications of linear programming Intro to transportation theory Applications of transportation theory Intro to network Programming Applications of network programming Traveling salesman problem Exam		
Assessment methods	Informative lectures Discussion Laboratories with computers The discussion summing up the knowledge gained during the lectures Written exam		
Recommended readings	1. Scientific papers and materials provided by the lecturer		
Knowledge	After the lectures the student will be able to define and descrbe: -linear programming methods and problems, -transportation task methods and problems, -network programming methods and problems, -traveling salesman problem.		
Skills	The student will be able to use the methods which will be presented on the laboratories		

Course title	Intro to Statistic: Making Decisions Based on Data		
Level of course	first cycle		
Teaching method	laboratory course / lecture		
Person responsible for the course	Wojciech Sałabun	**E-mail address to the person**	wsalabun@wi.zut.edu.pl
Course code (if applicable)	WI-1-IST	**ECTS points**	4
Semester	winter/summer	**Language of instruction**	english
Hours per week	4	**Hours per semester**	60
Objectives of the course	The course introduces to techniques for visualizing relationships in data and systematic techniques for understanding the relationships using mathematics.		
Entry requirements	None		
Course contents	Visualizing relationships in data (seeing relationships in data and predicting based on them, simpson's paradox, etc.) Probability (Bayes Rule, correlation vs. causation, etc.) Estimation (maximum likelihood estimation, mean, median, mode, standard deviation, variance, etc.) Outliers and normal distribution (outliers, quartiles, binomial distribution, central limit theorem, manipulating normal distribution, etc.) Inference (confidence intervals, hypothesis testing, etc.) Regression (linear regression, correlation, etc.) Exam Visualizing relationships in data (seeing relationships in data and predicting based on them, simpson's paradox, etc.) Probability (Bayes Rule, correlation vs. casuation, etc.) Estimation (maximum likelihood estimation, mean, median, mode, standard deviation, variance, etc.) Outliers and normal distribution (outliers, quartiles, binomial distribution, central limit theorem, manipulating normal distribution, etc.) Inference (confidence intervals, hyphotesis testing, etc.) Regression (linear regression, correlation, etc.) Exam		
Assessment methods	Informative lectures Discussion Laboratories with computers The discussion summing up the knowledge gained during the lectures Written exam		
Recommended readings	1. Scientific papers and materials provided by the lecturer		
Knowledge	After the lectures the student will be able to define and describe presented statistical techniques and measures		
Skills	The student will be able to calculate and use the main statistical measures and techniques		

Course title	Knowledge Engineering and Ontology Development		
Level of course	first cycle		
Teaching method	laboratory course / lecture		
Person responsible for the course	Agnieszka Konys	**E-mail address to the person**	Agnieszka.Konys@zut.edu.pl
Course code (if applicable)	WI-1-KEO	**ECTS points**	4
Semester	winter/summer	**Language of instruction**	english
Hours per week	4	**Hours per semester**	60
Objectives of the course	Familiar with the syntax, structures and principles used in OWL language The ability to design and write small-scale ontologies and to use reasoning mechanisms		
Entry requirements	None		
Course contents	Introduction to the ontologies Protégé ontology editor and OWL language Building an OWL ontology: defining class hierarchy OWL object property characteristics Building an OWL ontology: defining individuals and data type properties Graphical visualization of the ontology Describing and defining classes The application of reasoning mechanisms and query tools The application of plugins and tools to manage the ontology Introduction to the ontologies Ontology editors and standards for ontology description Selected approaches to the ontology construction and knowledge engineering methods Building an OWL ontology Primitive and defined classes Selected reasoning mechanisms and Open World Reasoning Reusing of existing ontologies Creating other OWL constructs in Protégé Restriction types Ontology-based solutions to knowledge extraction Exam		
Assessment methods	Informative lectures Discussion Work with computers at laboratories Written exam Project work		
Recommended readings	1. Michael K. Smith, Chris Welty, and Deborah L. McGuinness, OWL Web Ontology Language Guide, 2004, http://www.w3.org/TR/owl-guide/ 1. Matthew Horridge (eds.), A Practical Guide To Building OWL Ontologies Using Protege 4 and CO-ODE Tools Edition 1.2, The University of Manchester, Manchester, 2009 2. Protege tutorial. Available from http://protege.stanford.edu/		
Knowledge	After the course the student should be able to understand and use the basic ontology constructs in OWL After the course the student should be able to design and construct a small-scale ontology		
Skills	After the course the student should be to write small-scale ontologies using the above skills. After the course the student should be able to project from scratch and implement OWL ontology.		
Other social competences	The student will acquire the following attitudes: creativity in creating ontologies, understanding OWL components, construct small-scale ontology and using reasoning mechanism to ectract the knowledge.		

Course title	LaTeX		
Level of course	first cycle		
Teaching method	laboratory course / lecture		
Person responsible for the course	Remigiusz Olejnik	**E-mail address to the person**	Remigiusz.Olejnik@zut.edu.pl
Course code (if applicable)	WI-1-LAT	**ECTS points**	2
Semester	winter/summer	**Language of instruction**	english
Hours per week	2	**Hours per semester**	30
Objectives of the course	Practical skills in typesetting of engineering documents using LaTeX system.		
Entry requirements	Ability to use a computer running Linux or MS Windows operating system.		
Course contents	Preparing of documents of increasing complexity; changing of the font type and size, defining of the text layout, tables, complex mathematical formulas and mathematical texts; creating and inserting pictures; analysis of style files and preparation own styles for journals, books, reports and thesis; merging results of all exercises in a single document with the form of a book, with table of contents, bibliography, appendices and index. Description of the installation and initialization of the package, setting of environment variables, hyphenation file. LaTeX input file and the principles of its building, permanent elements of the file. Structure of the document: the division of the document into parts, chapters, sections, paragraphs, etc., title page, the main file and included files, creating of a table of contents, table of figures and tables, attaching a bibliography, creating an index, references to the labels, usage of the counters. Defining own classes of documents: building of the style definition file and possibilities of changing its content. Defining of running heads for page headings and footers, defining of parameters for lists, floating objects, defining of headers for chapter and subsections, changing of the format of the table of contents and bibliography. Predefined classes of document and format, format definition file declared in the preamble (page size, the type of numbering, margins, running head, footer). Defining the type and size of fonts, special characters, accents, Polish diacritic characters. Length measures, horizontal and vertical spacing, references, breaking lines and pages. Defining of indivisible elements. Multiple columns usage. Greek and Cyrillic alphabet. Mathematical texts: mathematical environment, using mathematical expressions and symbols (indices, fractions, roots, equations and their systems, matrices, complex formulas), spacing and bold in math mode. Special text structures: defining minipages, lists and tables, creating pictures and including them into document, language of geometric figures definition. Changes to the definitions, creating of own definitions and defining a new environment. Creating new variable objects. Correction of the errors: error messages and warnings in LaTeX and TeX, error correction capabilities.		
Assessment methods	Lecture with presentation Laboratory work - individual preparation of the document with increasing complexity Lecture - oral exam Laboratory work - evaluation of submitted document that has been prepared during the course		
Recommended readings	1. L. Lamport, LaTeX: A Document Preparation System, Addison-Wesley, Boston, 1994 2. F. Mittelbach et al., The LaTeX Companion (Tools and Techniques for Computer Typesetting), Addison-Wesley, Boston, 2004		
Knowledge	Student has knowledge about typesetting engineering documents with LaTeX system		
Skills	Student has practical skills in typesetting of engineering documents with LaTeX system		

Course title	Machine Learning		
Level of course	first cycle		
Teaching method	laboratory course / lecture		
Person responsible for the course	Przemysław Klęsk	**E-mail address to the person**	pklesk@wi.zut.edu.pl
Course code (if applicable)	WI-1-DAM	**ECTS points**	2
Semester	winter/summer	**Language of instruction**	english
Hours per week	2	**Hours per semester**	30
Objectives of the course	Developping a general understanding about data analysis and machine learning methods.		
Entry requirements	mathematics algorithms and data structures programming probability calculus and statistics		
Course contents	Programming PCA in MATLAB. Programming CART trees in MATLAB. Programming SVM optimization tasks (several versions) in MATLAB. Programming MARS algorithm in MATLAB. Principal Component Analysis (PCA) as a method for dimensionality reduction. Review of notions: variance, covariance, correlation coefficient, covariance matrix. Minimization of projection lengths of data points onto a given direction. Derivation of PCA. Interpretation of eigenvalues and eigenvectors. Decision trees - CART algorithm. Impurity functions, greedy generation of a complete tree. Pruning heuristics for decision trees (depth-based, leaves-based). Support Vector Machines (SVM). Distance of data points from the decision hyperplane. Separation margin. Formulation of the SVM optimization task without and with Lagrange multipliers. Support vectors - what are they? Soft-margin SVM and related optimization tasks. SVMs with non-linear decision boundary using the kernel trick. Multivariate Adaptive Regression Splines (MARS) for approximation tasks. Construction of splines. Least-squares approximation with arbitrary bases (in particular MARS splines). Learning algorithm. Similarities to CART. Exam.		
Assessment methods	Lecture. Computer programming. Four short tests (15 minutes long) at the end of each topic during the lab. Four grades for the programs written as homeworks. Final grade for the lab calculated as a weighted mean from partial grades: - tests (weight: 40%), - programs (weight: 60%). Final grade for lectures from the test (2 h).		
Recommended readings	1. M. J. Zaki, W. Meira Jr, Data Mining and Analysis - Fundamental Concepts and Algorithms, Cambridge University Press, 2014 2. P. Klęsk, Electronic materials for the course available at: http://wikizmsi.zut.edu.pl, 2015		
Knowledge	Student posesses an elementary knowledge on machine learning algorithms and techniques of data analysis.		
Skills	Student can implement (in Python or MATLAB) several machine learning algorithms and techniques.		

Course title	Microprocessor design and soft-processors		
Level of course	first cycle		
Teaching method	laboratory course / project course / lecture		
Person responsible for the course	Mariusz Kapruziak	**E-mail address to the person**	Mariusz.Kapruziak@zut.edu.pl
Course code (if applicable)	WI-1-MDS	**ECTS points**	3
Semester	winter/summer	**Language of instruction**	english
Hours per week	3	**Hours per semester**	45
Objectives of the course	Designing of PCB with processor on board. Processor internal structure and soft processor creation on FPGA. General knowledge about processor internal structure.		
Entry requirements	Basics of Eelctronics. Digital Design.		
Course contents	Soft processor structure of example processor in details. Processor PCB design. Processor programming Implementation of a chosen processor or a processor system. Different implementations of the ALU from inside. Synthesis of a control unit Low power technologies – methodologies, its advantages and pitfalls. Processor example in Verilog. Hardware description languages. Dynamic instruction set processors and processors with dynamic structure. PCB design for processor.		
Assessment methods	Lectures. Laboratories. Project. Final exam. Laboratory reports. Project.		
Recommended readings	1. P. Ienne, R. Leupers, Customizable Embedded Processors: Design Technologies and applications, Morgan Kaufmann 2. J. Nurmi, Processor Design: System-On-Chip Computing for ASICs and FPGAs, Springer, 2007 3. D. Liu, Embedded DSP Processor Design, Volume 2: Application Specific Instruction Set Processors, Morgan Kaufmann, 2008		
Knowledge	Student knows in details basic processor structures and can tailor it to the project.		
Skills	Student can design (hardware and software) a processor system. Student can create custom soft-processor in Verilog/VHDL on FPGA device.		

Course title	Mobile Application Development		
Level of course	first cycle		
Teaching method	laboratory course / project course / lecture		
Person responsible for the course	Radosław Maciaszczyk	**E-mail address to the person**	Radoslaw.Maciaszczyk@zut.edu.pl
Course code (if applicable)	WI-1-MAD	**ECTS points**	4
Semester	winter/summer	**Language of instruction**	english
Hours per week	3	**Hours per semester**	45
Objectives of the course	The main objective of the course is to intorduction to Androis OS Students will be prepared to create applications for mobile devices with Android OS		
Entry requirements	Knowledge of at least one object programming language, Preferred Java language		
Course contents	Introduction to Android Application Fundamentals User Interface Sensors and Location Data Storage Connectivity Camera and audio Introduction to project Project Documentation Presentation procjet Introducing to mobile device. The History of Android Application Fundamentals Activity lifecycle User Interface Sensors Threads and Services Storing and retrieving data Networking Multimedia Location Services.		
Assessment methods	Lectures: informative, problem solving, conversational. Laboratory classes with a computer Problems discution at laboratory classes Discussion of the individual project, brainstorm Assessment of the project created during practical exercises and discussion of the final repot. Verification of reports from selected laboratories. Presentation and defense of the project in front of a group of students.		
Recommended readings	1. Ian F. Darwin, Android Cookbook, Problems and Solutions for Android Developers, O'Reilly, 2012 2. Zigurd Mednieks, Laird Dornin, G. Blake Meike, Masumi Nakamura, Programming Android, 2nd Edition-Java Programming for the New Generation of Mobile Devices, O'Reilly, 2012		
Knowledge	After the lectures the student will be able to know the architecture of the Android application		
Skills	After course students knows how writing android applications using good rules.		

Course title	Parallel Programming		
Level of course	first cycle		
Teaching method	laboratory course / lecture		
Person responsible for the course	Włodzimierz Bielecki	**E-mail address to the person**	Wlodzimierz.Bielecki@zut.edu.pl
Course code (if applicable)	WI-1-PAP	**ECTS points**	4
Semester	winter/summer	**Language of instruction**	english
Hours per week	4	**Hours per semester**	60
Objectives of the course	To develop an understanding of major parallel programming models To be able to identify promising applications of parallel computing To be able to develop typical parallel algorithms and implement prototype parallel programs using API OpenMP To be able to analyze the performance of parallel programs		
Entry requirements	Compilers 1 You are expected to have some basic programming skills using C or C++.		
Course contents	Pragma parallel Pragma For Pragma Sections Pragma Single Pragma Critical Coding an algorithm in OpenMP Evaluating speed-up of an OpenMP program Applying Amhdal's and Gustafson's laws Introduction: From serial to parallel thinking. A history of parallel computers and lessons learned from them. Dependences in programs Basic loop transformations		

	API OpenMP, version 2. Processes and threads Fork-Join model What does OpenMP stand for? Limitations of OpenMP OpenMP Directive Responsibility Synchronization in OpenMP Pragma Parallel and its clauses What is a structured block Control of the number of threads in a parallel region Dynamic threads Nested parallel regions Parallel directive restrictions Private, firstprivate, shared, and default clauses Purpose of the DO / for directive and its restrictions Ordered clause Last private clause Schedule clause Reduction clause Nowait clause Default scoping rules in OpenMP Exceptions to the rule that unscoped variables are made shared by default. Removing anti dependences Removing output dependences Removing data flow dependences TREADPRIVATE clause COPYIN clause Pragma SECTIONS and its clauses Restrictions of pragma Sections Pragma single, its clauses and restrictions Combined constructs Restrictions of work-sharing constructs Orphan directives Scopes in an orphan construction Nested parallelism Environment variables Run-Time Library Routines Need for synchronization CRITICAL directive and its restrictions Atomic directive, its restriction Using the lock routines to implement a critical section BARRIER directive, its restrictions ORDERED directive, its restrictions MASTER directive, its restrictions FLUSH directive, its restrictions Parallel Program Performance metrics. Key factors impacting performance Cashes and Locality Locality and Schedules False sharing Inconsistent parallelization How barriers impact performance How critical sections impact performance Good Practice improving performance Deterministic program Program granularity Program locality How caches work Program speed-up Program efficiency AMDAHL'S LAW GUSTAFSON'S LAW Parallel algorithm design Performance models
Assessment methods	Informative / conversational lectures Laboratory exercises the Final exam by checking the learning outcomes: presenting questions and assessing the answers Assessment of the degree of practical tasks at the end of each laboratory
Recommended readings	1. Rohit Chandra Ramesh Menon Leo Dagum David Kohr Dror Maydan Jeff McDonald, Parallel Programming in OpenMP, Morgan Kaufmann, 2001 2. Thomas Rauber, Parallel Programming: for Multicore and Cluster Systems, Springer, 2010
Knowledge	The student has basic knowledge in the OpenMP standard.
Skills	The student is able to write parallel programs in the OpenMP standard.
Other social competences	The student is able to work with colleagues in a group.

Course title	Programmable control devices 1		
Level of course	first cycle		
Teaching method	laboratory course / lecture		
Person responsible for the course	Sławomir Jaszczak	**E-mail address to the person**	Slawomir.Jaszczak@zut.edu.pl
Course code (if applicable)	WI-1-PD1	**ECTS points**	3
Semester	winter/summer	**Language of instruction**	english
Hours per week	3	**Hours per semester**	45
Objectives of the course	General knowledge about : sensors and actuators , real time operation systems, logic functions, timers and counters, machine state syntesis in the Structured Text language. Ability to syntesize logic functions, timers and counters, machine state syntesis in the Structured Text language.		
Entry requirements	Physics - basics of the electricity Electronics - basics of DC systems Basic knowledge of the selected programming language (C/C++, Java, Python etc.)		
Course contents	Basics of the ST programming Syntesis of the logic functions Syntesis of the state machine Programming of the selected electrical motor - DC o AC Introduction to programmable controllers Sensors and actuators. Real time operation systems Basics of the Structured Text language. Logic functions in the Structured Text language. Timers and counters in the Structured Text language Machine state syntesis in the Structured Text language. Tips and tricks in the Structured Text language. Exam		
Assessment methods	Conversational lecture Information lecture Laboratory exercises Programming projects Oral test Final project with oral test		
Recommended readings	1. Kelvin T. Erickson, Programmable Logic Controllers, Dogwood Valley Press, 2016 2. B&R, Structured Text, B&R, 2017		
Knowledge	General knowledge of the ST language syntax and ability of logic functions and machines state synthesis.		
Skills	Ability of using general syntax of the ST language (logic functions, machines state, timers, counters, SET-RESET functions)		

Course title	Programmable control devices 2		
Level of course	first cycle		
Teaching method	laboratory course / lecture		
Person responsible for the course	Sławomir Jaszczak	**E-mail address to the person**	Slawomir.Jaszczak@zut.edu.pl
Course code (if applicable)	WI-1-PD2	**ECTS points**	3
Semester	summer	**Language of instruction**	english
Hours per week	3	**Hours per semester**	45
Objectives of the course	General knowledge about feedback loop control structures and basic analog control algorithms (two state, PID etc.) Programming skills in structured text : Pre-processing of analog signals Syntesis of the two state control algorithm Syntesis of the PID control algorith		
Entry requirements	Basic knowledge of the selected programming language (C/C++, Java, Python etc.) Physics - a general knowledge of dynamical systems		
Course contents	Pre-processing of the analog signals in the ST programming Syntesis of the two state control algorithm Syntesis of the PID control algorithm Synthesis of the selected real time control system - speed or position control system Introduction to the feedback loop control Analog sensors and actuators. Two state control algorithm. PID control algorithm Exam		
Assessment methods	Conversational lecture Information lecture Laboratory exercises Oral or the written test Programming projects Final project with the oral test		
Recommended readings	1. Kelvin T. Erickson, Programmable Logic Controllers, Dogwood Valley Press, 2016 2. B&R, Structured Text, B&R, 2017		
Knowledge	General knowledge of the ST language syntax related to the feedback loop control.		
Skills	Ability of using general syntax of the ST language (PID controller, types conversion, scaling-averaging-filtering functions)		

Course title	Prolog Programming for Artifcial Intelligence		
Level of course	first cycle		
Teaching method	laboratory course / lecture		
Person responsible for the course	Joanna Kołodziejczyk	**E-mail address to the person**	Joanna.Kolodziejczyk@zut.edu.pl
Course code (if applicable)	WI-1-PPA	**ECTS points**	3
Semester	winter/summer	**Language of instruction**	english
Hours per week	3	**Hours per semester**	45
Objectives of the course	Knowledge in Prolog programming and the ability to recognize different algorithms from Artificial Inteligence Ability to implement some (search, reasoning, inductive programming, belief networks) AI algoritghm using Prolog programming language		
Entry requirements	The course does not require any previous knowledge		
Course contents	Simple example - facts and rules Declarative and procedural meaning Operators and arithmetic Lists in Prolog Eight queens problem solution Cut, negation and backtracking Build in predicates Debugging Tree and graph representation and search Expert systems (if then) Minimax - game playing From First predicate logic to Prolog Prolog syntax, lists, operators, arithmetics Backtracking and build in predicates Program examples - search blind and informed Expert systems in Prolog Game playing		
Assessment methods	Lecture, presentation Discussion, learning by doing Software developing in Prolog Short programming tasks Writing exam or quiz from knowledge representation and Prolog.		
Recommended readings	1. Ivan Bratko, Prolog programming for Artificial Intelligence, Pearson Education, 2001		
Knowledge	Explain the logic programming paradigm. Understand the resoninig in Prolog. Represent knowledge in First Predicate Logic and Prolog syntax.		
Skills	Develop a given algorithm in Prolog using build-in and own predicates. Debug the Prolog code. Describe how the result is obtained.		

Course title	Python GUI Programming		
Level of course	first cycle		
Teaching method	laboratory course / lecture		
Person responsible for the course	Krzysztof Małecki	**E-mail address to the person**	Krzysztof.Malecki@zut.edu.pl
Course code (if applicable)	WI-1-PGP	**ECTS points**	3
Semester	winter/summer	**Language of instruction**	english
Hours per week	3	**Hours per semester**	45
Objectives of the course	The student knows the implementation methods of programs with a graphical user interface in Python, knows how to identify interface elements that facilitate data retrieval and visualization.		
Entry requirements	The student knows the lexis and syntax of Python to the extent necessary to independently construct simple procedural programs, knows how to use the integrated programming environment. The student knows the basic principles of constructing algorithms.		
Course contents	Displaying the basic elements of the GUI. Input and display data from the user. Implementation of the event handler. Implementation of various algorithmic tasks Introductory and organizational issues. Basic elements of the user interface. Input and output of data. Reacting to events and the main event handling loop. Possibility of graphical data representation using ready-made components. Communication with the program using the menu. Basic techniques for testing user interface programs. Final test.		
Assessment methods	Information lecture Problem lecture Labs Independent exercises Based on observation of work during laboratory classes The final test		
Recommended readings	1. John. E. Grayson, Python and Tkinter Programming, Manning Publications, 2000		
Knowledge	One knows the techniques of visual programming.		
Skills	The student can prepare an application with a basic GUI		
Other social competences	The student understands the role of a programmer in the software development process		

Course title	Python Programming Language		
Level of course	first cycle		
Teaching method	laboratory course / lecture		
Person responsible for the course	Krzysztof Małecki	**E-mail address to the person**	Krzysztof.Malecki@zut.edu.pl
Course code (if applicable)	WI-1-PYT	**ECTS points**	4
Semester	winter/summer	**Language of instruction**	english
Hours per week	4	**Hours per semester**	60
Objectives of the course	Presentation of Python programming rules and syntax. Developing practical programming skills in Python.		
Entry requirements	None.		
Course contents	The work environment. The first program. Exercises in procedural programming. Exercises in object-oriented programming. Exercises in reading and writing to text, binary and XML files. Debuging and testing. The final project. The examination of the final project. Basic information about Python and programming environments. Introduction to procedural programming (types of variables, complex data types, collections, arithmetical and logical operators, programm control commands, functions, input/output operations, lists, tuples, sets, dictionaries) Programm control command (conditional instruction, loops, exeption handling). Modules and packages. Python language libraries. Files support - reading and saving to binary, text and XML files. Object-oriented programming (classes, atributes, methods). Class inheritance and polymorphism. Own data types and colletions. Class decorators. Debugging, testing. The final test.		
Assessment methods	Wykład informacyjny z prezentacją multimedialną oraz z użyciem komputera. Labaratory: self-solving tasts withe the support of the teacher. The final test. Labaratory: current assessment od learning process and the assessment of the final project.		
Recommended readings	1. Charles Severance, Python for everybody, 2016 2. Programming Python, Mark Lutz, O'Reilly Media, USA, 2011		
Knowledge	After the course the student is able to understand the basic programming constructs of Python language		
Skills	Student is able to use basic programming constructs of Python language and he/she is able to write the small-scale Python scripts		

Course title	Social media and complex network analytics		
Level of course	first cycle		
Teaching method	laboratory course / lecture		
Person responsible for the course	Jarosław Jankowski	**E-mail address to the person**	Jaroslaw.Jankowski@zut.edu.pl
Course code (if applicable)	WI-1-SMC	**ECTS points**	3
Semester	winter	**Language of instruction**	english
Hours per week	3	**Hours per semester**	45
Objectives of the course	To acquaint students with the methods and algorithms of complex network analysis To acquaint students with the methods of modeling behavior in complex networks		
Entry requirements	Basic programming skills		
Course contents	Computational tools and libraries for network analysis Network visualization tools Analizy teoretycznych modeli sieci Determining and analyzing network metrics Algorithms for recognizing communities in networks Dynamic network analysis Analyzes of multilayer networks Agent systems in modeling network phenomena Modeling influence and forming opinions in social networks Fundamentals of modeling information propagation processes Modeling information propagation processes using the cascade model Modeling information propagation processes using the threshold model Social network sampling Real network analysis Introduction to social media and complex networks Network metrics and visualisation Community detection in social networks Dynamic and multilayer networks Modeling information spread in networks Social networks sampling Network robustness		
Assessment methods	Lecture with presentations and examples Laboratory exercises and implementation of practical tasks Lecture: summary assessment. Written credit with practical questions, questions in the form of a selection and description. Laboratories: assessment based on reports and attendance.		
Recommended readings	1. Zuhair M., Kadry S., Python for Graph and Network Analysis, Springer, Berlin, 2017 2. Hanneman R.A., Riddle M., Introduction to social network methods, Riverside, Los Angeles, 2005 3. Barabási A.L., Network science, Cambridge university press, Cambridge, 2016		
Knowledge	Knowledge of modeling and analysis of complex networks and knowledge of modeling behavior in complex networks.		
Skills	The ability to model and analyze complex networks and the ability to model behavior in complex networks		
Other social competences	As a result of the course, the student will develop an active cognitive attitude and a desire for professional development		

Course title	Software engineering		
Level of course	first cycle		
Teaching method	laboratory course / lecture		
Person responsible for the course	Łukasz Radliński	**E-mail address to the person**	lradlinski@zut.edu.pl
Course code (if applicable)	WI-1-SEN	**ECTS points**	3
Semester	winter	**Language of instruction**	english
Hours per week	3	**Hours per semester**	45
Objectives of the course	Possess knowledge and obtain practical skills in developing main products of software engineering process. Usage of techniques and tools for development process where outcomes from one stage flow to subsequent stages. Practicing individual and team-based work in a software project.		
Entry requirements	Basic knowledge and skills in object-oriented programming, relational databases.		
Course contents	Introduction to software engineering labs. Organisational issues. Preparing student environment. Project definition and scope Writing user and system specifications Use cases and their specifications User interface wireframing and design, processing design Software analysis and modelling Database design Implementation of the prototype of the architecture Definition of test cases Project presentation and grading Introduction to software engineering. Gathering customer/user requirements. Writing user and system specifications. Software analysis and modelling - UML diagrams. Software designing. Architectural patterns. Data design. Software versioning. Basics of Software Testing. Test for grading		
Assessment methods	Informative lecture with demonstration Lab exercises Project Individual exercises Individual or group project Test with open questions		
Recommended readings	1. Bruegge B., Dutoit A.H., Object-Oriented Software Engineering Using UML, Patterns and Java, Prentice Hall, 2009, 3rd edition 2. Larman C., Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development, Prentice Hall, 2004, 3rd Edition		
Knowledge	Describes main terms, processes and techniques used in software engineering.		
Skills	Can create software project documentation with requirements specification, architectural design, and main test cases.		
Other social competences	Ability to communicate with non-technical people		

Course title	Software for Engineering Calculations		
Level of course	first cycle		
Teaching method	laboratory course / lecture		
Person responsible for the course	Marcin Pluciński	**E-mail address to the person**	Marcin.Plucinski@zut.edu.pl
Course code (if applicable)	WI-1-SEC	**ECTS points**	2
Semester	winter/summer	**Language of instruction**	english
Hours per week	3	**Hours per semester**	45
Objectives of the course	Learning of main features and functions of Mathcad. Learning of MATLAB and its programming language.		
Entry requirements	Basics of computer science.		
Course contents	Mathcad - basics of the operation. Evaluation of values of mathematical expresions. Defining of variables and using them in calculations. Creating of user functions. Creating of 2D and 3D plots. Solving equations. Formatting of Mathcad sheets. Performing of symbolic algebraic manipulations. Filnal work in Mathcad. Exercises in programming in MATLAB. Creating of matrices. Matrix operations. Matrix indexing. Exercises in creating 2D and 3D plots. Creating and running scripts. Exercises in creating and running functions. Exercises in solving of simple numerical problems. Simulink and selected MATLAB toolboxes. Exercises in creating MATLAB programs with GUI (Graphic User Interface). Final work in MATLAB. Introduction to software for engineering calculation - overview of systems and their possibilities. Mathcad - possibilities of the program and basics of its operation. Solving of basic tasks such as: calculations, matrix operations, plotting, creating of user functions, solving equations, transforming of symbolic expressions. MATLAB - a program for engineering calculations. General assumptions of the system design. The organization of work with the system. The definition of variables in MATLAB. Methods of matrix creation. Basic matrix operations. Basics of programming in MATLAB (scripts, functions, control commands). Types of variables and data structures in MATLAB and related commands. 2D and 3D plots in MATLAB. Fundamentals of data analysis in MATLAB. The basic numerical procedures. Overview of selected toolboxes. Evaluation of knowledge.		
Assessment methods	Lecture with presentation. Lab - solving of selected problems in Mathcad. Lab - programming of selected problems in MATLAB. Laboratory: evaluation of tasks realized during the classes. Lecture: written test.		
Recommended readings	1. The Mathworks Inc., MATLAB - the language of Technical Computing, available online, 2015 2. PTC, PTC Mathcad Tutorials, avaiable online, 2016		
Knowledge	The student knows the most important functions and possibilities of Mathcad. The student knows capabilities of MATLAB and its language (syntax, use, available functions, categories of tasks that can be solved with it).		
Skills	The student has the ability to use Mathcad in engineering and scientific calculations. The student has the ability to use MATLAB in engineering and scientific calculations and to program in its language.		

Course title	Stochastic Optimization		
Level of course	first cycle		
Teaching method	laboratory course / lecture		
Person responsible for the course	Jan Rodziewicz-Bielewicz	**E-mail address to the person**	rj26733@zut.edu.pl
Course code (if applicable)	WI-1-STO	**ECTS points**	4
Semester	winter/summer	**Language of instruction**	english
Hours per week	4	**Hours per semester**	60
Objectives of the course	To introduce and discuss algorithm that were inspired by biological phenomenon (part of Artificial Intelligence domain). Application of different algorithms in various real and test problems		
Entry requirements	Basic programming skills		
Course contents	Optimization - simple heuristics Genetic algorithm implementation Evolution strategies implementation Particle Swarm Optimization algorithm implementation Differential evolution implementation Ant colony optimization for discrete problems - implementation Immune systems - Clonalg, anomaly detection Neural networks - supervised learning - implementation Neural network - usupervised Hybrid solutions - implementation Computation intelligence - introduction Evolutionary algorithm Optimization task - chalanges Evolution strategies Differential evolution Particle Swarm Optimization as a robust optimization method in continues domain Ant colony optimization for discrete problems. Artificial Immune Systems as an optimization tool Neural networks - supervised Neural networks - unsupervised Hybrid methaheuristics		
Assessment methods	Lecture with presentation and conversation Software development. Quiz checking the knowlage on biologicaily inspired algorrithms. Examination of programming tasks		
Recommended readings	1. Thomas Weise, Global Optimization Algorithms - Theory and Application, online book, 2011		
Knowledge	Student will know how to apply different algorithms and will be aware of the power, and the limitations, of discussed during the course methods.		
Skills	Practical skills of implementing, analysing and testing algorithms described during the course.		

Course title	Web application development with Angular framework		
Level of course	first cycle		
Teaching method	laboratory course / lecture		
Person responsible for the course	Bartłomiej Małachowski	**E-mail address to the person**	Bartlomiej.Malachowski@zut.edu.pl
Course code (if applicable)	WI-1-JSW	**ECTS points**	3
Semester	winter/summer	**Language of instruction**	english
Hours per week	3	**Hours per semester**	45
Objectives of the course	After the course a student will be able to independently develop Angular app with web service data-exchange and authentication capabilities.		
Entry requirements	At least intermediate level in object-oriented programming Good knowledge on HTML and CSS languages		
Course contents	Setting up environement for writing and running angular apps Wrting simple components Writing simple services Handling simple REST web service in Angular app Handling forms in angular app Development of simple CRUD app Adding authentication to angular app Angular app testing Basic concepts of Angular framework: architecture (modules, components, services), TypeScript vs Javascript, AngularJs vs Angular Principles of writing and running Angular apps: settig up environment, command line tools, app creation, scalfolding, running an application in development mode, building of production ready app Working with components and databinding Services and dependency injection Routing Handling forms Making HTTP requests Authentication in angular apps		
Assessment methods	Individual work - programming tasks Auditorial lectures Final exam Evalution of developed programming tasks through code review made by the teacher		
Recommended readings	1. Angular project team, Angular framework official docs and tutorials, Angular project, web, 2019, 7, https://angular.io/docs 2. Greg Lim, Beginning Angular with Typescript, Greg Lim, 2018, 3		

Course title	Алгоритмические основы цифровой обработки сигналов и изображений		
Level of course	first cycle		
Teaching method	lecturing course / lecture		
Person responsible for the course	Aleksandr Cariow	**E-mail address to the person**	Alexandr.Tariov@zut.edu.pl
Course code (if applicable)	WI-1-AOC	**ECTS points**	3
Semester	winter/summer	**Language of instruction**	russian
Hours per week	2	**Hours per semester**	30
Objectives of the course	Целью освоения настоящей дисциплины является формирование и систематизация знаний в области обработки сигналов и изображений представленных в цифровой форме. Основной задачей дисциплины является обучение студентов теоретическим знаниям и алгоритмам построения систем цифровой обработки сигналов и изображений, а также привитие им практических навыков по методологии инженерных расчетов основных характеристик и показателей эффективности изучаемых алгоритмов и систем.		
Entry requirements	Требования к предварительной подготовке обучающегося: Знание основ элементарной математики, матричной алгебры, цифровой техники.		
Course contents	Элементы матричной алгебры. Представление одномерного сигнала в виде вектора, двумерного (изображения) - в виде матрицы. Специальные типы матриц. Единичная и нулевая матрицы. Матрицы сдвига, перестановки, растяжения, дублирования. Изучение операций конкатенации, тензорного (кронекеровского) произведения, прямой суммы. Графическое представление алгоритмов ЦОС в виде сигнальных графов. Изучение и ислледование особкнностей векторно-матричных процедур БПФ. Решение примеров на построение алгоритмов БПФ (по основанию 2 и 4) для конкретных значений исходных последовательностей данных. Изучение особенностей построения быстрых алгоритмов дискретных ортогональных преобразований (ДОП) для различных длин исходных последовательностей данных. Решение задач на построение быстрых алгоритмов ДОП Уолша, Хаара, Хартли и т.д. Решение задач на построение быстрых алгооитмов одномерной и двумерной свёртки. Разработка алгоритмов быстрой свёртки (круговой и линейной) во временной и частотной областях. Решение задач на применение методов "overlap-save" и "overlap-add". Решение задач на построение алгоритмов прямого и обратного дискретного вейвлет-преобразования в базисе фильтров Добеши. Зачётное занятие. Подведение итогов изучения предмета и выставление оценок. Введение. Аналитический обзор и обсуждение основных задач, методов и приложений цифровой обработки сигналов (ЦОС). История ЦОС. Преимущества ЦОС. Достоинства и недостатки ЦОС. Элементы матричной алгебры. Представление основных операций цифровой обработки сигналов и изображений с помощью объектов алгебры матриц (в том числе в виде матрично-матричных и векторно-матричных произведений). Спектр цифрового сигнала. Дискретное преобразование Фурье (ДПФ). Свойства ДПФ. Быстрое преобразование Фурье (БПФ), алгоритмы с прореживанием по времени и частоте. Операция "бабочка". Двоично-инверсная адресация. Алгоритм Винограда. ДПФ действительных последовательностей. Обобщение ДПФ. Дискретные ортогональные преобразования в базах Уолша, Хаара, Виленкина, Хартли. Дискретное косинус-преобразование. Быстрые алгоритмы дискретных ортогональных преобразований в перечисленных базисах. Цифровые свёртка и корреляция. Круговая и линейная свёртка. Быстрые алгоритмы вычисления круговой свёртки. Цифровая фильтрация. ФИльтры КИХ и БИХ. Реализация операции фильтрации с помощью дискретных ортогональных преобразований. Вычисление линейной свёртки с помощью круговой. Фильтрация длинных последовательносьтей: методы "overlap-save" и "overlap-add". Вейвлет-технологии. История. Определение вейвлета. Многоуровневая декомпозиция и реконструкция. Алгоритм Малла - дискрентое вейвлет-преобразование. Фильтры Добеши. Вычислительные процедуры дискретного вейвлет-преобразования. Вейвлетоподобные преобразования. Элементная база процессоров цифровой обработки сигналов и изображений. Тенденции развития специализированных микросистем ЦОС. Распараллеливание вывчислений: конвейерная и векторная обработка данных. Обзор и обсуждение достоинств и недостатков современных параллельных СБИС-структур, ориентированных на реализацию задач ЦОС. Диалектические аспекты ускорения вычислений. Высокопроизводительные вычисления: единство и борьба противоположностей.		
Assessment methods	Лекции и практические занятия с использованием мультимедийных презентаций. Отличная оценка выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины, проявляющему интерес к данной предметной области, продемонстрировавшему умение уверенно и творчески применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений. Четвёрка с плюсом выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений. Четвёрка выставляется студенту, показавшему систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, правильное обоснование принятых решений, с некоторыми недочетами.		

	Оценка 3,5 выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но недостаточно грамотно обосновывает полученные результаты. Оценка 3 выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач некоторые неточности. Оценка 2 в выставляется студенту, показавшему фрагментарный, разрозненный характер знаний, допускающему ошибки в формулировках базовых понятий, нарушения логической последовательности в изложении программного материала, слабо владеет основными разделами учебной программы, необходимыми для дальнейшего обучения и с трудом применяет полученные знания даже в стандартной ситуации.
Recommended readings	1. Рабинер Л. Гоулд Б., Теория и применение цифровой обработки сигналов., Пер. с англ. Зайцева А.Л. Назаренко Э.Г. - М: Мир, Москва, 1978, - 835с. 2. Дагман, Э.Е.; Кухарев, Г.А., Быстрые дискретные ортогональные преобразования, Издательство: Наука, Новосибирск, 1983, - 232 с. 3. Юкио Сато, Обработка сигналов: первое знакомство, М: Додэка-XXI, 2010, – 176 с. 4. Прэтт У., Цифровая обработка изображений, Пер. с англ.—М.: Мир, Пер. с англ.—М.: Мироскова, 1982, два тома, — 312 с. 5. Блейхут Р, Быстрые алгоритмы цифровой обработки сигналов, Мир, Москва, 1989, - 448с. 6. Нуссбаумер Г., Быстрое преобразование Фурье и алгоритмы вычисления сверток, Пер. с англ. - М.: Радио и связь, Москва, 1985, - 248с. 7. Ахмед Н., Рао К.Р., Ортогональные преобразования при обработке цифровых сигналов, Пер. с англ. — М.: "Связь", Москва, 1980, — 248 с. 8. Хуанг Т. С., Эклунд Дж. О., Нуссбаумер Г., Быстрые алгоритмы в цифровой обработке изображений, Перю с англ.б М.: Радио и связь,, Москва, 1984, — 224 с.
Knowledge	Знать: - преимущества цифровой обработки сигналов и иё роль в проектировании приборов, устройств и узлов телекоммуникационных информационных систем; - математический аппарат для описания цифровых сигналов и изображений; - основные методы и алгоритмы цифровой обработки сигналов и изображений; - области применения цифровой обработки сигналов; - современную элементную базу для реализации систем цифровой обработки сигналов;
Skills	Уметь: - математически описывать цифровые сигналы и изображения; - проектировать (проводить синтез и рассчитывать параметры) базовых алгоритмов цифровой обработки сигналов и изображений; - применять полученные знания и методы обработки сигналов для решения практических задач ЦОС и ЦОИ, - самостоятельно приобретать новые знания в области цифровой обработки сигналов и изображений.
Other social competences	Владеть: - математическими и алгоритмическими методами проектирования и оценки систем цифровой обработки сигналов; - ориентироваться в современной литературе по цифровой обработке сигналов.