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
SECOND DEGREE

	Course title	Person responsible for the course	Semester (winter/summer)	ECTS points	Hours
1	Audio Signal Processing	Mirosław Łazoryszczak	winter/summer	3	30
2	Big Data analytics tools and software	Agnieszka Konys	winter/summer	4	60
3	Brain-Computer Interface	Izabela Rejer	winter/summer	4	60
4	C# Programming Language	Marcin Pietrzykowski	winter/summer	3	45
5	C++ programming language	Agnieszka Konys	winter/summer	4	60
6	Computer Networks	Remigiusz Olejnik	winter/summer	4	60
7	Computer Vision for Video Surveillance	Adam Nowosielski	winter/summer	3	30
8	Database systems	Przemysław Korytkowski	winter/summer	5	60
9	Data Mining Algorithms	Przemysław Klęsk	winter/summer	3	45
10	Digital Circuits	Mirosław Łazoryszczak	winter/summer	4	60
11	EEG signal analysis in Matlab	Izabela Rejer	winter/summer	4	60
12	Embedded systems	Mirosław Łazoryszczak	winter/summer	4	60
13	Expert systems	Joanna Kołodziejczyk	winter/summer	3	45
14	F# Programming Language	Marcin Pietrzykowski	winter/summer	3	45
15	Fundamentals of Error-Correcting Block Codes	Dorota Majorkowska-Mech	winter	3	30
16	Graphical User Interface in .NET	Marcin Pietrzykowski	winter/summer	2	30
17	Human-Computer Interaction	Adam Nowosielski	winter/summer	3	30
18	Intelligent Decision Systems	Wojciech Sałabun	winter/summer	4	60
19	Introduction to Natural Language Processing	Joanna Kołodziejczyk	winter/summer	4	60
20	Introduction to the Internet of Things	Remigiusz Olejnik	winter/summer	4	60
21	Intro to Mathematical Programming	Wojciech Sałabun	winter/summer	4	60
22	Intro to Statistic: Making Decisions Based on Data	Wojciech Sałabun	winter/summer	4	60
23	Knowledge Engineering and Ontology Development	Agnieszka Konys	winter/summer	4	60
24	LaTeX	Remigiusz Olejnik	winter/summer	2	30
25	Machine Learning	Przemysław Klęsk	winter/summer	2	30
26	Prolog Programming for Artifcial Intelligence	Joanna Kołodziejczyk	winter/summer	3	45
27	Software engineering	Łukasz Radliński	winter	3	45
28	Stochastic Optimization	Jan Rodziewicz-Bielewicz	winter/summer	4	60
29	АЛГОРИТМИЧЕСКИЕ ПРИЁМЫ И ТРЮКИ В ЦИФРОВОЙ ОБРАБОТКЕ СИГНАЛОВ И	Aleksandr Cariow	winter/summer	4	30

	Course title	Person responsible for the course	Semester (winter/summer)	ECTS points	Hours
	ИЗОБРАЖЕНИЙ				

Course title	Audio Signal Processing		
Level of course	second 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-2-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	Big Data analytics tools and software		
Level of course	second 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-2-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	second 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-2-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	C# Programming Language		
Level of course	second 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-2-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	second 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-2-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		

Course title	Computer Networks		
Level of course	second 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-2-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 Vision for Video Surveillance		
Level of course	second 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-2-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	second 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-2-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	second 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-2-DMA	**ECTS points**	3
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.		
Skills	Student can implement (Python or MATLAB) algorithms presented during lectures.		

Course title	Digital Circuits		
Level of course	second 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-2-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	EEG signal analysis in Matlab		
Level of course	second 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-2-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	second 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-2-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. Assessment		
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	Expert systems		
Level of course	second 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-2-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	second 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-2-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	Fundamentals of Error-Correcting Block Codes		
Level of course	second 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-2-FEC	**ECTS points**	3
Semester	winter	**Language of instruction**	english
Hours per week	2	**Hours per semester**	30
Objectives of the course	Knowledge and skills in error-correcting codes construction. 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	second 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-2-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	second 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-2-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	second 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-2-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 Natural Language Processing		
Level of course	second 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-2-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	second 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-2-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	second 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-2-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	second 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-2-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	second 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-2-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, 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		

Course title	LaTeX		
Level of course	second 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-2-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	second 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-2-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 has an elementary knowledge on machine learning algorithms and techniques suitable for data analysis.		
Skills	Student can implement (Python or MATLAB) algorithms presented during lectures.		

Course title	Prolog Programming for Artifcial Intelligence		
Level of course	second 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-2-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 test 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	Software engineering		
Level of course	second 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-2-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 Software analysis and modelling User interface wireframing and design, processing design 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	Stochastic Optimization		
Level of course	second 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-2-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. Test 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	АЛГОРИТМИЧЕСКИЕ ПРИЁМЫ И ТРЮКИ В ЦИФРОВОЙ ОБРАБОТКЕ СИГНАЛОВ И ИЗОБРАЖЕНИЙ		
Level of course	second 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-2-APR	**ECTS points**	4
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. Задачей дисциплины является: - обучение студентов теоретическим знаниям построения систем цифровой обработки сигналов и изображений, а также алгоритмическим трюкам и приёмам, приводящим к снижению вычислительной сложности разрабатыаемых алгоритмов и процессорных структур. - привитие студентам практических навыков по методологии инженерных расчетов основных характеристик и показателей эффективности разрабатываемых алгоритмов.
Entry requirements	Предмет не требует каких-либо специальных знаний. Все необходимые теоретические сведения и понятия будут предоставлены и объяснены в процессе проведения занятий.
Course contents	Изучение элементов алгебры кронекеровых произведений, как наиболее удобной формы описания, синтеза и реализации алгоритмов ЦОС и ЦОИ.. Изучение набора базовых (эталонных) структур матриц, допускающих эффективную факторизацию, приводящую к минимизации арифметической сложности реализации макроопераций ЦОС и ЦОИ. Изучение универсальной методики рационализации вывчислений векторно-матричных произведений. Рассмотрение примера, Решение практических задач на разработку быстрых алгоритмов вычисления векторно-матричных произведений. Определение и выдача индивидуальных заданий на разработку конкретных алгоритмов ЦОС и ЦОИ, характеризующихся уменьшенной арифметической сложностью и использующих принципы распараллеливания вычислений. Обсуждение текущего состояния решения индивидуальныз заданий, связанных с проектированием алгоритмов. Консультации. Подсказки. Обсуждение текущего состояния решения индивидуальныз заданий, связанных с проектированием алгоритмов. Консультации. Подсказки. Зачётное занятие. Оценка и обсуждение правильности решения индивидуальных заданий. Обсуждение достоинств и недостатков предложенных решений. Заключительная дискуссия. Обзор основных методов и задач цифровой обработки сигналов (ЦОС) и изображений (ЦОИ). Изучение известных алгоритмических приёмов и трюков, позволяющих сократить объём вычислений при решении задач ЦОС и ЦОИ (методы Штрассена, Винограда, трюк Гаусса и т.д.) Представление основных макроопераций цифровой обработки сигналов и изображений с помощью объектов алгебры матриц Рационализация вычисдений векторно-матричных произведений. Демонстрация новых приемов и способов сокращения количества арифметических операций при вычислении векторно-матричных произведений. Универсальная (авторская) методика синтеза быстрых алгоритмов алгоритмов вычисления векторно-матричных произведений. Примеры синтеза алгоритмов векторно-матричных преобразований с уменьшенным числом арифметических операций. Синтез быстрых алгоритмов для решения основных задач ЦОС и ЦОИ (круговая и линейная свертка, FDWT / IDWT, DCT, DFT, Хартли, Хаара, Уолша-Адамара, Слэнт-преобразование и других дискретных преобразований). Распараллеливание вычислений, как способ сокращения времени реализации вычислительных процессов. параллельные алгоритмы векторно-матричных произведений. параллельные алгоритмы реализации базовых макроопераций ЦОС и ЦОИ. Обсужление возможностей и особенностей построения эффективных структур специализированных процессорных узлов и модулей, использующих преимущества, получаемых от вводимых рационализаций. Заключение. Зачётное занятие, выставление оценок за теоретическую часть предмета.
Assessment methods	Лекции с использованием мультимедийных презентаций и практические занятия. Оценки за решение домашних заданий. В конце - экзамен в форме опроса.
Recommended readings	1. Блейхут Р., Быстрые алгоритмы цифровой обработки сигналов, Пер. с англ. - М.: Мир, Москва, 1989, - 448с. 2. Нуссбаумер Г., Быстрое преобразование Фурье и алгоритмы вычисления сверток, Пер. с англ. - М.: Радио и связь, Москва, 1985, - 248с. 3. Хуанг Т. С., Эклунд Дж. О., Нуссбаумер Г., Быстрые алгоритмы в цифровой обработке изображений, Пер. с англ. М.: Радио и связь, Москва, 1984, - 224 с. 4. Макклеллан Дж.Г., Рейдер Ч.М., Применение теории чисел в цифровой обработке сигналов, Изд-во: М.: Радио и связь, Изд-во: М.: Радио и связь, 1983 г.;осква, 1983, - 264 с.
Other social competences	

	В результате освоения дисциплины обучающиеся должны знать: - основные алгоритмические трюки и способы рационализации алгоритмов ЦОС, предназначенных для реализации в FPGA; - уметь: - синтезировать высокоэффективные алгоритмы ЦОС, подходящие для реализации на FPGA; - описывать вычислительные процедуры в матричной форме, верифицировать их с помощью моделирования; - владеть: - навыком освоения большого объема информации; - навыками постановки научно-исследовательских задач и навыками самостоятельной работы.