Faculty of Computer Science and Information Technology

WEST POMERANIAN UNIVERSITY OF TECHNOLOGY
IN SZCZECIN, POLAND

THE OFFER FOR INTERNATIONAL STUDENTS
FOR THE YEAR 2023/2024
SECOND DEGREE

	Course title	Person responsible for the course	Semester (winter/summer)	ECTS points	Hours
1	Arduino Prototyping	Janusz Papliński	winter/summer	5	60
2	Artificial Intelligence	Przemysław Klęsk	winter/summer	5	60
3	Audio Signal Processing	Mirosław Łazoryszczak	winter/summer	5	60
4	Big Data analytics tools and software	Agnieszka Konys	winter/summer	5	60
5	Business Intelligence	Przemysław Różewski	winter/summer	5	60
6	C++ programming language	Agnieszka Konys	winter/summer	5	60
7	Compilers	Włodzimierz Bielecki	winter/summer	5	60
8	Computer Games Programming	Radosław Mantiuk	summer	6	75
9	Computer Networks	Grzegorz Śliwiński	winter/summer	5	60
10	Computer System Architecture	Mariusz Kapruziak	winter/summer	5	60
11	Database systems	Przemysław Korytkowski	winter/summer	5	60
12	Digital Systems	Mariusz Kapruziak	winter/summer	5	60
13	Dynamic documents and front-end Web development	Wiesław Pietruszkiewicz	winter	5	60
14	E-commerce and online marketing technologies	Wiesław Pietruszkiewicz	winter	5	60
15	Embedded systems	Mirosław Łazoryszczak	winter/summer	5	60
16	Expert systems	Joanna Kołodziejczyk	winter/summer	5	60
17	Human-Computer Interaction	Adam Nowosielski	winter/summer	5	60
18	Intelligent Decision Systems	Wojciech Sałabun	winter/summer	5	60
19	Introduction to Mathematical Programming	Wojciech Sałabun	winter/summer	5	60
20	Introduction to Natural Language Processing	Joanna Kołodziejczyk	winter/summer	5	60
21	Machine Learning	Przemysław Klęsk	winter/summer	5	60
22	Mobile Application Development	Radosław Maciaszczyk	winter/summer	5	60
23	Parallel Programming	Włodzimierz Bielecki	winter/summer	5	60
24	Programmable control devices	Sławomir Jaszczak	summer	5	60
25	Prolog Programming for Artifcial Intelligence	Joanna Kołodziejczyk	winter/summer	5	60
26	Python Programming Language	Krzysztof Małecki	winter/summer	5	60
27	Signal processing for Brain-Computer Interfaces	Izabela Rejer	winter/summer	5	60
28	Social media and complex network analytics	Jarosław Jankowski	winter	5	60
29	Software Engineering	Łukasz Radliński	winter	5	60

	Course title	Person responsible for the course	Semester (winter/summer)	ECTS points	Hours
30	Алгоритмические основы цифровой обработки сигналов и изображений	Aleksandr Cariow	winter/summer	5	60

Course title	Arduino Prototyping		
Level of course	second cycle		
Teaching method	laboratory class / project		
Person responsible for the course	Janusz Papliński	**E-mail address to the person**	Janusz.Paplinski@zut.edu.pl
Course code (if applicable)	WI-1-ARD	**ECTS points**	5
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	Artificial Intelligence		
Level of course	second cycle		
Teaching method	laboratory class / 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**	5
Semester	winter/summer	**Language of instruction**	english
Hours per week	4	**Hours per semester**	60
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. Acquirement of competence and practice in construction of fuzzy models of systems, fuzzy calculations and fuzzy control of plants.		
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). 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). 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. Applications of RBF neural networks in modeling of technical and economic problems. Applications of RBF neural networks in classification tasks. Application of unsupervised learning networks to the data clustering problem. Hopfield network - application to the pattern recognition problem. Discovering fuzzy phenomena, fuzzy variables, fuzzy notions in the world. Identification of membership functions for own detected uncertain values from science, technique, medicine, economics, biology etc. Describing membership fumctions by mathematical formulas. Creating rule bases for real systems. Design and implementation of the simple SISO fuzzy system. Design and implementation of the MISO fuzzy system. Application of the fuzzy model in the control system. 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. 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. 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. 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. Diffrence between classical and fuzzy logic. Examples of fuzziness in the real world. Mathematical models of fuzzy linguistic and numerical evaluations: membership functions. Examples of membership functions. Identification of membership functions by experts. 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. Fuzzy control and its structure. Exam.
Assessment methods	Lecture. Case study method. Didactic games. Computer programming. Demonstration. Short tests (10 minutes long) at the end of each topic during the lab. 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 2. A. Piegat, Fuzzy modelling and control, Physica-Verlag, A Springer-Verlag Company, 2001 3. D. Kriesel, A Brief Introduction to Neural Networks, 2012
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	Audio Signal Processing		
Level of course	second cycle		
Teaching method	laboratory class / 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**	5
Semester	winter/summer	**Language of instruction**	english
Hours per week	4	**Hours per semester**	60
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 Sound source localization. Selected digital filter implementation Audio effects implementation eg. delay, echo, pitch shift etc. Music pitch and onset retrieval methods Assessment. Sound synthesis. Sound basics and audio perception. Principles of acoustic. Audio signal characteristics and representations. Sound source localization. Digital filters (FIR, IIR) - parameters, characteristics, design methods. Audio effects (echo, delay, reverb etc.) Elements of music transcription (pitch and onset detection, genre classification etc.). Sound synthesis. Home recording studios: acoustics and equipment (microphones and speakers, mixing consoles) 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 2. Zoelzer U. (ed.), DAFX – Digital Audio Effects, Wiley, 2002		
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 class / 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**	5
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. Be able to design Data Warehouse and use MDX effectively.		
Entry requirements	Basic understanding of main business processes		
Course contents	Instructions for Downloading and Installing the Exercise Environment HIVE: Creating Databases and Tables, SQL SELECT Essentials, Working with Data Types, Working with File Types, Loading Files into HDFS Working with Spark in Python: Use Spark core concepts such as RDDs, transformations, actions to operate on large datasets Application of information extraction methods and techniques Big data processing and analysis tools Big Data Visualization tools Classic Data vs. Big Data Big Data Essentials: Hadoop, HDFS, MapReduce The Hadoop Stack Ecosystem Introduction to NoSQL Databases Orientation to SQL on Big Data Managing Big Data in Clusters: Hive, Hue Introduction to Apache Spark Information extraction from text Methods and techniques for information extraction Big data processing and analysis 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 4. Alejandro Vaisman Esteban Zimányi, Data Warehouse Systems Design and Implementation, Springer-Verlag Berlin Heidelberg, 2013, DOI: 10.1007/978-3-642-54655-6		
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. Student will know how to integrate the Big Data and Data Warehousing.		
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. Student is able to design and querying Data Warehouse.		
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	Business Intelligence		
Level of course	second cycle		
Teaching method	laboratory class / 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**	5
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	Dashboard Design in PowerBI: multi data sources integration, DAX, PowerQuery. Analysis of Data from Multiple Business Perspectives: ETL process design, data quality, visualisation, multidimensional data representation. Business Intelligence Concepts Data Visualization for Analytics and Business Intelligence Storytelling with Data BI tools: Microsoft PowerBI, Google Data Studio Dashboard Design Business Analytics Fundamentals Data Warehouse Concept and Achitectures Extraction, Transformnation and Loading (ETL) process design Multidimensional Data Representation and Manipulation Data Engineering Data Warehouse in Cloud		
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	second cycle		
Teaching method	laboratory class / 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**	5
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	second cycle		
Teaching method	laboratory class / 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**	5
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	second cycle		
Teaching method	project / 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**	6
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 illumination model. 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	second cycle		
Teaching method	laboratory class / lecture		
Person responsible for the course	Grzegorz Śliwiński	**E-mail address to the person**	Grzegorz.Sliwinski@zut.edu.pl
Course code (if applicable)	WI-1-CTN	**ECTS points**	5
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	second cycle		
Teaching method	laboratory class / 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**	5
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. 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	Database systems		
Level of course	second cycle		
Teaching method	laboratory class / 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	Digital Systems		
Level of course	second cycle		
Teaching method	laboratory class / 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-DIG	**ECTS points**	5
Semester	winter/summer	**Language of instruction**	english
Hours per week	4	**Hours per semester**	60
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	Combinatorial logic and arithmetic circuits. Waveforms generation using sequential logic. Selected application implementation (eg. Morse code, audio waveform generation). Advanced topics - resource sharing and optimalization. Assessment. Combinatorial logic. Functional Blocks. Enabling. Decoding. Multiplexer-based combinational circuits. Adder. Subtractor. HDL models of combinatorial circuits. Combinatorial logic design. Sequential logic definitions. Latches. State tables and diagrams. Sequential circuits analysis and design. Verilog/VHDL languages. Basics of FPGA/CPLD devices architectures. Digital circuits technologies. Memories. Static and dynamic, synchronous and asynchronous. RAM types. Synthesis methods and tools of digital systems. Assessment.		
Assessment methods	Lectures. Laboratories. Project Final Exam Laboratory reports. Project.		
Recommended readings	1. M. Morris R. Mano, Michael D. Ciletti, Digital Design, Pearson, 2018, 6 2. C.M. Maxfield, The Design Warrior’s Guide to FPGAs, Linacre House		
Knowledge	Student knows basics of HDL and RTL synthesis. Student knows structures of digital systems.		
Skills	Student is able to program in Verilog/VHDL.		

Course title	Dynamic documents and front-end Web development		
Level of course	second cycle		
Teaching method	laboratory class / lecture		
Person responsible for the course	Wiesław Pietruszkiewicz	**E-mail address to the person**	Wieslaw.Pietruszkiewicz@zut.edu.pl
Course code (if applicable)	WI-1-DDO	**ECTS points**	5
Semester	winter	**Language of instruction**	english
Hours per week	4	**Hours per semester**	60
Objectives of the course	Understanding selected programming languages and data processing methods in dynamic Web systems.		
Entry requirements	A basic understanding of internet technologies		
Course contents	Preparation of development environment HTML Cascade Style Sheets CSS preprocessors Web desing & user experience Web media JavaScript CMS/Fes - selected CMS/F Server-side - template engine Server-side - selected web application framework Introduction to the web-based systems Web-development environment Markup lanaguages - with focus on HTML Web styling - Cascade Style Sheets & preprocessors Web design principles User experience & web evaluation Webmedia standards JavaScript basics JavaScript common libraries Data in web systems - XML, JSON & Web Storage Content Management Systems and Frameworks Server-side technologies - a review of the most popular ones Selected server-side technology - programming basics and template engines Selected server-side technology - webapp frameworks Newest trends in the web-development		
Assessment methods	Lectures with presentations, and review of case studies Laboratory-based practical exercises Lectures - Written exam with knowledge-oriented choice questions, and skill-oriented open-ended questions Laboratory classes - Overall assessment based on reports and attendance		
Recommended readings	1. Anne Boehm, Zak Ruvalcaba, HTML5 and CSS3, Murach, NY, 2015 2. David Flanagan, Javascript: The Definitive Guide: Master the World's Most-Used Programming Language, O'Reilly UK Ltd., 2020		
Knowledge	Knowledge required to design dynamic web documents		
Skills	Skills required to develop dynamic web documents		

Course title	E-commerce and online marketing technologies		
Level of course	second cycle		
Teaching method	laboratory class / lecture		
Person responsible for the course	Wiesław Pietruszkiewicz	**E-mail address to the person**	Wieslaw.Pietruszkiewicz@zut.edu.pl
Course code (if applicable)	WI-1-ECO	**ECTS points**	5
Semester	winter	**Language of instruction**	english
Hours per week	4	**Hours per semester**	60
Objectives of the course	To deliver knowledge about area of e-commerce, and skills necessary to prepare & perform an e-commerce project To deliver knowledge about area of online marketing, and skills necessary to prepare & perform an online marketing project		
Entry requirements	A basic understanding of internet technologies		
Course contents	Hosted webshops Preparation of an environment for a self-hosted webshop Development of a self-hosted webshop - configuration, themes, products, shipping, payments & order management Visual online marketing Web content - including content marketing Mailing & newsletters Search engines - SEO & SEM Social media - channels, presence & content Analytics of data in e-commerce and online marketing Business evaluation of digital commerce and marketing Introduction to the commercial Internet E-commerce models Review of IT technologies used in e-commerce Webshops & trading platforms Payment gateways and other specialised systems System integration in e-commerce Basics of online marketing Online marketing strategies Search engines - optimisation and marketing Social media - characteristics & usages Social marketing strategies Social media software integration Content and behaviour analysis - including intelligent systems in e-commerce and online marketing Digital commerce and marketing from business perspective		
Assessment methods	Lectures with presentations, and review of case studies Laboratory-based practical exercises Lectures - Written exam with knowledge-oriented choice questions, and skill-oriented open-ended questions Laboratory classes - Overall 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	Knowledge required to plan e-commerce activities Knowledge required to plan online marketing activities		
Skills	Skills required to conduct an e-commerce project Skills required to conduct an online marketing project		

Course title	Embedded systems		
Level of course	second cycle		
Teaching method	laboratory class / 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**	5
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. Assessment. 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 class / 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**	5
Semester	winter/summer	**Language of instruction**	english
Hours per week	4	**Hours per semester**	60
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 Expert Systems in CLIPS Prolog - logic programming - syntax Expert systems in Prolog Membership functions identyfication the simple SISO fuzzy system design and implementation The MISO fuzzy system design and implementation Expert Systems - definitions, examples. Historical examples and ideas. Konowledge representation - propositional logic. Knowledge representation - First order predicate. First order logic to programming in logic. Dealing with uncetrainty - probablistic view. Bayes theorem and bayesian networks. Probabilisitic rule based expert systems Expert systems based on certainty factor. Fuzzy logic intrudution - mathematical fundamentals Fuzzy expert systems - fuzzifiaction, inference, rules development Fuzzy expert systems examples		
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	Human-Computer Interaction		
Level of course	second cycle		
Teaching method	lecture / laboratory class / project		
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**	5
Semester	winter/summer	**Language of instruction**	english
Hours per week	4	**Hours per semester**	60
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. Implementation of a prototype or own idea in the field of HCI.		
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 class / 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**	5
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 Mathematical Programming		
Level of course	second cycle		
Teaching method	laboratory class / 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**	5
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	Introduction to Natural Language Processing		
Level of course	second cycle		
Teaching method	laboratory class / 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**	5
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	Machine Learning		
Level of course	second cycle		
Teaching method	laboratory class / 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**	5
Semester	winter/summer	**Language of instruction**	english
Hours per week	4	**Hours per semester**	60
Objectives of the course	Developping a general understanding about data analysis and machine learning methods. 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 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. 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. 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. 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	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. M. J. Zaki, W. Meira Jr, "Data Mining and Analysis - Fundamental Concepts and Algorithms", Cambridge University Press, 2014		

	3. 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. Student has an elementary knowledge on data mining algorithms and notions.
Skills	Student can implement (in Python or MATLAB) several machine learning algorithms and techniques. Student can implement (MATLAB or Python) data mining algorithms presented during lectures.

Course title	Mobile Application Development		
Level of course	second cycle		
Teaching method	laboratory class / project / 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**	5
Semester	winter/summer	**Language of instruction**	english
Hours per week	4	**Hours per semester**	60
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 Kotlin Data Binding ViewModel, Live Data Testing in Android Useful library in android Introduction to project Project Documentation Presentation project Introducing to mobile device. The History of Android Application Fundamentals Components lifecycles Architecture Components User Interface Sensors Threads and Services Storing and retrieving data Networking 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	second cycle		
Teaching method	laboratory class / 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**	5
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		
Level of course	second cycle		
Teaching method	lecture / laboratory class		
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**	5
Semester	summer	**Language of instruction**	english
Hours per week	4	**Hours per semester**	60
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. 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	Physics - basics of the electricity Electronics - basics of DC systems Basic knowledge of the selected programming language (C/C++, Java, Python etc.) Physics - a general knowledge of dynamical systems		
Course contents	Introduction to programmable controllers Sensors and actuators. 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. Feedback loop control Two state control algorithm. PID control algorithm Exam Basics of the ST programming Syntesis of the logic functions Syntesis of the state machine Basic discrete and analog actuators & sensors Pre-processing of the analog signals in the ST programming Syntesis of the two state control algorithm Syntesis of the PID control algorithm Stability and quality analysis Synthesis of the selected real time control system (temperature, position, speed etc.)		
Assessment methods	Conversational lecture Information lecture Laboratory exercises Programming projects Oral test Final project with oral test Oral or the written test 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 and ability of logic functions and machines state synthesis. General knowledge of the ST language syntax related to the feedback loop control.		
Skills	Ability of using general syntax of the ST language (logic functions, machines state, timers, counters, SET-RESET functions) 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	second cycle		
Teaching method	laboratory class / 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**	5
Semester	winter/summer	**Language of instruction**	english
Hours per week	4	**Hours per semester**	60
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 Blind and informed search Expert systems in Prolog Game playing Planing Reasoning with uncertenty Learning a decision tree		
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 Programming Language		
Level of course	second cycle		
Teaching method	laboratory class / 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**	5
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	Signal processing for Brain-Computer Interfaces		
Level of course	second cycle		
Teaching method	laboratory class / 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**	5
Semester	winter/summer	**Language of instruction**	english
Hours per week	4	**Hours per semester**	60
Objectives of the course	To teach students how to process, analyze, and classify EEG signal. To equip students with the ability of designing and programming interfaces controlling the external devices with user's brain activity.		
Entry requirements	None		
Course contents	Introduction to Matlab programming Building a script for signal recording. Building a script with instructions for a user (supervised session). Recording EEG signal for off-line processing. EEG signal preprocessing (spatial, spectral, and statistical data filtering). Implementing feature extraction methods. Implementing (and testing) classification methods. Adapting all the scripts for an on-line BCI. Tests with real users. Presenting the test results. EEG signals - main characteristics The definition and types of Brain Computer Interfaces (BCI). Removing artifacts from EEG signal (EEG signal preprocessing). Spectral analysis of EEG signal (Fourier transform) Extracting features for BCI control. The classification rules/schemes used in BCIs. Final discussion.		
Assessment methods	Informative lectures. Discussion. Laboratories with computers and EEG devices. The presentation describing the interface created during laboratories and tests results. An oral exam in a form of discussion summing up the knowlegde gained during the lectures.		
Recommended readings	1. Official Matlab site: http://www.mathworks.com/help/matlab/ 2. 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 3. S. W. Smith, Digital Signal Processing: A practical Guide for Engineers and Scientists, 2003		
Knowledge	After the lectures the student will be able to: define a BCI, describe the main problems with EEG data, descibe different BCI paradigms, choose the processing methods suitable for different paradigms and different EEG data.		
Skills	The student will be able to create a Brain-Computer Interface suitable for a given task.		

Course title	Social media and complex network analytics		
Level of course	second cycle		
Teaching method	laboratory class / 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**	5
Semester	winter	**Language of instruction**	english
Hours per week	4	**Hours per semester**	60
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 Multilayer networks Dynamic networks Social influence maximisation Epidemic spreading in 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	second cycle		
Teaching method	laboratory class / 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**	5
Semester	winter	**Language of instruction**	english
Hours per week	4	**Hours per semester**	60
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 lab 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. Design patterns. Software versioning. Software Quality Assurance and Testing. Software Project Risk Management. Estimation and Prediction in Software Engineering. Software Development Methodologies. Software Evolution and Maintenance. 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. Ian Sommerville, Software Engineering, Pearson, 2015, 10 2. Bruegge B., Dutoit A.H., Object-Oriented Software Engineering Using UML, Patterns and Java, Prentice Hall, 2009, 3rd edition 3. 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	Алгоритмические основы цифровой обработки сигналов и изображений		
Level of course	second cycle		
Teaching method	auditory class / 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**	5
Semester	winter/summer	**Language of instruction**	russian
Hours per week	4	**Hours per semester**	60
Objectives of the course	Обучить студентов теоретическим знаниям и основным алгоритмам цифровой обработки сигналов (ЦОС), и изображений (ЦОИ) Привить студентам практические навыки по методологии разработки эффективных алгоритмов и структур вычислительных модулей для систем ЦОС и ЦОИ.		
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 исследование 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 с использованием мультимедийных презентаций. Практические занятия. экзамен устный в форме собеседования письменный или устный зачёт коллоквиум		
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.