Institute of Computer Science

http://kimk.up.krakow.pl/en/

Course given in English

COURSE TITLE winter semester	ECTS CREDIT
Communication and Project Management	1
Management Information Systems	2
User interface design and user experience	3
Object-oriented programming techniques	5
Software engineering	3
Java Network Programming	4
Developing Mobile Applications	4
Web Applications	4

COURSE TITLE summer semester	ECTS CREDIT
Advanced web design 2	4
Quality Assurance of Information Systems	2
Mathematics 1	6
Computer organization and architecture	5
Computer simulations	2
Foundations of modeling and simulations	3
Numerical analysis	4
Multimedia Technology and its Applications	1
Advanced web technologies 1	3

Course title	Advanced web design 2		
Semester (winter/summer)	summer	ECTS	4
Lecturer(s)	Piotr Kukuła		
Department	Institute of Informatics UP		

Course objectives (learning outcomes)

The aim of the course is to acquaint students with advanced mechanisms and techniques used in the process of designing and web development. The course will cover the creation of web services based on open development platforms (frameworks) using scripting languages based on the MVC pattern.

Prerequisites

Knowledge	Procedural and object-oriented programming. Basic web technologies (HTML / XHTML, CSS, JS) and subsidiaries (PHP). Relational databases.
Skills	Basics of programming in PHP and JavaScript. Basics of SQL.
Courses completed	Object-oriented programming 3. Web applications. Advanced web design 1.

Course organization							
Form of classes	W (Lecture)	Group type					
		A (large group)	K (small group)	L (Lab)	S (Seminar)	P (Project)	E (Exam)
Contact hours	10			30			

Teaching methods:

The course is in the form of laboratory classes, during which students work on projects related to the construction of websites using the selected development platform (Framework).
During the course, students receive individual assignment problem (project, paper) and one practical project requiring a comprehensive approach to design, implementation and launch a website based on the selected software platform (Framework) operating in the MVC / MVT.

Assessment methods:

	E – learning	Didactic games	Classes in schools	Field classes	Laboratory tasks	Individual project	Group project	Discussion participation	Student's presentation	Written assignment (essay)	Oral exam	Written exam	Other
					X	X	X	X	X				

Assessment criteria	The final grade depends on the ratings of partial and systematic performed tasks and work on projects. In particular, the assessment of good or very good can get a student who will finish the individual and the group project on time with positive results.

Comments	The course is discussed chosen by leading framework using PHP (recommended) or Python (if the level of the group is sophisticated enough). Group projects students may be (with the consent of leading) realized based on other than discussed in class framework implementing MVC pattern.

Course content (topic list)

1. Design Patterns type MVC.
2. Importance of layers (model, view and controller)
3. Project platforms (frameworks) to support the creation of Web sites based on MVC patterns: the installation and configuration; building, structure and performance; implementations of scripting languages; work in developer environment; database access,
4. Open licenses and development platforms used in constructing websites

Compulsory reading

1. Chris Pitt, „Pro PHP MVC", Apress
2. Oleg Krivtsov, "Using Zend Framework 2", Leanpub
3. Michael Romer, "Web Development with Zend Framework 2", Leanpub
4. http://framework.zend.com/

Recommended reading

1. J. Forcier, P. Bissex, W. Chun, "Python Web Development with Django", Addison-Wesley Professional
2. E. Freeman, E. Freeman, B. Bates, K. Sierra, "Head First Design Patterns", O'Reilly
3. http://pl.php.net/
4. http://www.w3.org/
5. http://www.python.org/

Course title	Communication and project management		
Semester (winter/summer)	winter	ECTS	1
Lecturer(s)	Tomasz Wojtowicz (MSc)		
Department	Institute of Informatics UP		

Course objectives (learning outcomes)

Key skills for work organization in a software development team. Good understanding of communication paths in the team, for different types of communication – strategic, team development, tactic, directly supporting project goals. Selection of communication tools and platforms, depending on the software development methodology.
Basic information about project management activities during different phases of the project like: project planning, scheduling, estimations (size and effort), risk analysis and management, change management.

Prerequisites

Knowledge	Basic information about realization of software projects.
Skills	Basic software development (programming) skills in any computer programming language.
Courses completed	No prerequisite courses required.

Course organization							
Form of classes	W (Lecture)	Group type					
		A (large group)	K (small group)	L (Lab)	S (Seminar)	P (Project)	E (Exam)
Contact hours				15			

Teaching methods:

Lab classes where initially a selected topic is presented to students, and in the latter part of the class students do some practical, hands-on exercises (in groups or individually).
In the second half of the semester students are asked to prepare a plan for a hypothetical software development (or general IT) project, including estimation of the overall effort (and cost), high level plan and activities, risk identification and mitigation analysis, etc. This assignment can be done individually or in pairs.

Assessment methods:

	E – learning	Didactic games	Classes in schools	Field classes	Laboratory tasks	Individual project	Group project	Discussion participation	Student's presentation	Written assignment (essay)	Oral exam	Written exam	Other
					X				X				

Assessment criteria	75% grade weight is from student's assignment (project plan) 25% grade weight is from student's activity and participation (presence) in classes.

Comments	None.

Course content (topic list)

L1. Communication as one of the key factors in success of software development projects.
L2. Channels of communication in the team.
L3. Overview of software development methodologies (waterfall, iterative, agile, etc.)
L4. Activities and roles in the project.
L5. Communication platforms and tools.
L6. Estimation in software development projects (size, cycle time, effort, assumptions, risks).
L7. Overview of selected estimation methods: SLIM, COCOMO, Wideband Delphi, AHP, Agile estimations.
L8. Estimation measures: KLOCs, function points, story points, staff months.
L9. Risks in software development projects.
L10. Selected formal risk management frameworks: Boehm's Win-Win RM, SEI RM, Kontio's RiskIt, P2I2.
L11. Risk identification methods (e.g. DFMEA), risk prioritization.
L12. Security risks, secure programming, secure design methods.

Compulsory reading

[1] Tom DeMarco, „Peopleware: Productive Projects and Teams", 1999
[2] Frederic P. Brooks "The Mythical Man-Month", 1995
[3] Mickey W. Mantle "Managing the Unmanageable: Rules, Tools, and Insights for Managing Software People and Teams", 2012
[4] Murali Chemuturi, "Software Estimation Best Practices, Tools & Techniques: A Complete Guide for Software Project Estimators", 2009.

Recommended reading

Delphi Process: http://www.stellman-greene.com/aspm/content/view/23/38/
AHP Tutorial: http://people.revoledu.com/kardi/tutorial/ahp/
Formal Risk Management: https://www.csiac.org/sites/default/files/Formal_Risk_Management.pdf
FMEA Overview: http://www.tangram.co.uk/TI-HSE-FMEA-Risk_Assessment.html
Secure Coding Principles: http://www.owasp.org/index.php/Secure_Coding_Principles

Course title	Management Information Systems		
Semester (winter/summer)	winter	ECTS	2
Lecturer(s)	Tomasz Wojtowicz (MSc)		
Department	Institute of Informatics UP		

Course objectives (learning outcomes)

Students learn about classification of information systems, needs for management information systems, supporting strategic enterprise goals with MIS, selection of MIS platforms for organization needs. Course covers evaluation of MIS from TCO and ROI perspective.

Prerequisites

Knowledge	Basic knowledge about databases and data stores, information systems and their use in business and industry.
Skills	No specific prerequisites.
Courses completed	No courses required.

Course organization							
Form of classes	W (Lecture)	Group type					
		A (large group)	K (small group)	L (Lab)	S (Seminar)	P (Project)	E (Exam)
Contact hours	15						

Teaching methods:

Lecture, discussions with students.

Assessment methods:

	E – learning	Didactic games	Classes in schools	Field classes	Laboratory tasks	Individual project	Group project	Discussion participation	Student's presentation	Written assignment (essay)	Oral exam	Written exam	Other
								X		X			

Assessment criteria	75% weight of the final grade is from the assignment. 25% weight of the final grade is from the activity and participation (presence) in classes.

Comments	

Course content (topic list)

L1. Classification and characteristics of management information systems. Basic definitions, generation of MIS systems, domain specific systems, electronic document control, information strategy in the organization.
L2. Introduction to MRP (Material Requirements Planning) and ERP (Enterprise Resource Planning), MRPII, CAM (Computer Aided Manufacturing).
L3. DMS (Document Management Systems), CRM (Customer Relationship Management), SFA (Sales Force Automation).
L4. Domain specific operation support system (e.g. telco OSS/BSS, AEP, CMP, M2M, DSP).
L5. E-commerce support systems, intranets.
L6. Business Inteligence systems, B2B, B2C, B2A, C2C, B2B2C platforms.
L7. MIS systems – project phases and lifecycle.
L8. Legal aspects of MIS.

Compulsory reading

Kenneth C. Laudon , Carol Guercio Traver, “Management Information Systems (12th Edition)” 12th Edition

Recommended reading:
www.comarch.com – really good selection of MIS systems for different domains, with good descriptions.
http://www.hit.bme.hu/~jakab/edu/litr/TMN/oss.pdf - OSS fundamentals
http://www.salesforce.com/uk/crm/what-is-crm.jsp - good introduction to CRM from Salesforce
http://www.personal.psu.edu/axk41/ERP-intro.pdf - introduction to ERP

Course title	User interface design and user experience		
Semester (winter/summer)	winter	ECTS	3
Lecturer(s)	Tomasz Wojtowicz (MSc)		
Department	Institute of Informatics UP		

Course objectives (learning outcomes)

Introduction to user interface design for applications, in particular web and mobile apps. Aspects of cross-platform and portable UI/UIX. Overview of UI/UX design patterns, depending on the application use case, target device and target user group. Basic UI design techniques, ergonomic rules and trends.

Prerequisites

Knowledge	Technologies of mobile or web apps development.
Skills	Programming/developing simple mobile apps and/or internet/web apps.
Courses completed	Introduction to mobile applications. Introduction to web applications.

Course organization							
Form of classes	W (Lecture)	Group type					
		A (large group)	K (small group)	L (Lab)	S (Seminar)	P (Project)	E (Exam)
Contact hours	10			15			

Teaching methods:

New UI/UX topics are introduced to the students during the lecture.
Then, during lab classes, students are picking the topic from the lecture, and enhancing the UI/UX for the projects (mobile or web) they did during their previous courses. Project improvements are done in an agile way, with 2-3 week sprints. Product backlog for each project is populated by the students, but prioritized for execution by the teacher (to feed the sprint planning).

Assessment methods:

	E – learning	Didactic games	Classes in schools	Field classes	Laboratory tasks	Individual project	Group project	Discussion participation	Student's presentation	Written assignment (essay)	Oral exam	Written exam	Other
					X		X	X					

Assessment criteria	In previous semesters, during other courses students were developing mobile and/or web apps. The key factor for the grade in this course is the effect of enhancing these applications from the UI/UX perspective. The app improvements will be assessed from the perspective of UI/UX, information architecture, general visual esthetics, user friendliness for key use cases. Additional points will be given for employing microinteractions and performing end user analysis (e.g. definition of personas for the project).

Comments	

Course content (topic list)

L1. General introduction to UI/UX
L2. General introduction to Information Architecture (IA)
L3. Good UI paradigms for web/internet applications.
L4. User oriented UI/UX techniques (personas, customer journeys/experience maps, mental models, sitepath diagramming) .
L5. Mobile apps trends.
L6. Web apps trends.
L7. UX myths.
L8. Web apps UI/UX dark patterns
L9. Microinteractions.
L10. UI/UX Prototyping tools.
L11. Evaluation of UI/UX.

Compulsory reading

Don’t make me think, Steve Krug
Design of Everyday Things, Donald A. Norman
Seductive Design, Steven P. Anderson
Microinteractions, Designing with Details, Dan Saffer
The Visual Display of Quantitative Information, Edward Tufte

Recommended reading

iOS Human Interface Guidelines
Smashing Magazine, Android UX resources.
UXPin resources.

Course title	Quality Assurance of Information Systems		
Semester (winter/summer)	summer	ECTS	2
Lecturer(s)	Tomasz Wojtowicz (MSc)		
Department	Institute of Informatics UP		

Course objectives (learning outcomes)

Goal of this course is to give students a basic introduction to quality assurance controls in sofrware development projects. Selected, most commonly used quality gates will be presented, various QA oriented techniques used by software developers and different types of tests will be discussed. Students will learn how QA fits different software development methodologies. Basic QA metrics will be presented. Continuous quality improvements methods will be presented.

Prerequisites

Knowledge	Basic knowledge about different software development methodologies. Basic software engineering information.
Skills	Students know how to write a simple program, write some unit tests for that program, specify functional, box level tests.
Courses completed	Object oriented programming. Agile software development methodologies. Web applications.

Course organization							
Form of classes	W (Lecture)	Group type					
		A (large group)	K (small group)	L (Lab)	S (Seminar)	P (Project)	E (Exam)
Contact hours				15			

Teaching methods:

Course is executed in the form of lab classes and via e-learning platform.
In the second half of the semester students are given individual assignments on some specific QA topic.

Assessment methods:

	E – learning	Didactic games	Classes in schools	Field classes	Laboratory tasks	Individual project	Group project	Discussion participation	Student's presentation	Written assignment (essay)	Oral exam	Written exam	Other
	X							X	X				

Assessment criteria	75% weight of the final grade is from the assignment. 25% weight of the final grade is from the activity and participation (presence) in classes.

Comments	

Course content (topic list)

L1. Quality assurance aspect of software development projects (examples of spectacular industry failures, cost of quality and cost of poor quality).
L2. Typical QA activities in the project (reviews, code analysis, testing, metrics, escaped defects analysis, corrective actions, DFMEA, audits, etc.)
L3. Quality Assurance in Agile methodologies.
L4. Overview of automatic code analysis tools.
L5. Examples of formal quality assurance frameworks: DSS, House of Quality.
L6. Change and defects management.
L7. Key Performance Indicators – identification and validation.
L8. Quality Assurance processes for complex, distributed, heterogeneous systems (telecommunication system example).

Compulsory reading

1. Robert C. Martin (Uncle Bob) - Clean Code: A Handbook of Agile Software Craftsmanship
2. Andrew Hunt and David Thomas, The Pragmatic Programmer: From Journeyman to Master, 1999
3. Lisa Crispin - Agile Testing: A Practical Guide for Testers and Agile Teams, 1st Edition
4. Jez Humble , David Farley - Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation (Addison-Wesley Signature Series (Fowler)) 1st Edition

Recommended reading

1. The Six Sigma Handbook, Fourth Edition 1 Jun 2014 by Thomas Pyzdek and Paul A. Keller
2. Continuous Integration: Improving Software Quality and Reducing Risk (Martin Fowler Signature Books) 29 Jun 2007 by Paul M. Duvall and Steve Matyas
3. Testing in Scrum: A Guide for Software Quality Assurance in the Agile World (Rocky Nook Computing) 7 Apr 2014 by Tilo Linz

Course title	Mathematics 1		
Semester (winter/summer)	summer	ECTS	6
Lecturer(s)	Institute of Informatics UP	Dr Łukasz T. Stępień	
Department			

Course objectives (learning outcomes)

The objective of this course is to learn the students to use the theory of sets and mathematical logics, the graph theory, to acquaint the students with the probability and statistics; to demonstrate connections between theory and applications of mathematics. Other course objective is to learn the students to use mathematics in applied computer sciences and to prepare them to continue education at the Master studies level in Computer Science.

Prerequisites

Knowledge	Knowledge of foundations of mathematics at secondary school level
Skills	Logical abstract reasoning. Capability of working with computer. Capability of working within a team.
Courses completed	Mathematics 1

Course organization							
Form of classes	W (Lecture)	Group type					
		A (large group)	K (small group)	L (Lab)	S (Seminar)	P (Project)	E (Exam)
Contact hours	30	30					

Teaching methods:

Lectures, computer presentations, exercises solving during the classes and at home, written tests

Assessment methods:

	E – learning	Didactic games	Classes in schools	Field classes	Laboratory tasks	Individual project	Group project	Discussion participation	Student's presentation	Written assignment (essay)	Oral exam	Written exam	Other
			X					X				X	

Assessment criteria	The assessment is based on oral and written answers to basic theoretical and practical questions. The student receives the mark "good" or "very good", if he/she: • proves fundamentals laws of classical propositional calculus • determines the classes of abstraction and quotient set of equivalence relations • is capable of creating and describing a graph for a concrete issue • applies correctly the principle of mathematical induction, in the proofs • is capable of deriving the formula for general term of a sequence given by a recurrence formula • is capable of computing probability of an event, by using the formulas of combinatorics • is capable of computing an estimation of random variable and/or its expected value. • is to apply statistical methods to experimental data

Comments	

Course content (topic list)

1. Classical propositional calculus, logical sentences. Logical functors (disjunction, conjunction, implication, equivalence, negation). Propositional formulas. Tautologies and countertautologies. The laws (the theses) of classical propositional calculus.
2. Functional calculus. Definition of propositional function. The domain and the truth set of propositional function. The quantifiers. The laws (the theses) of functional calculus.
3. Set algebra. Fundamental sets of numbers. Definition of subset. Definition of: sum, product, difference of two sets and complement of a set.
4. General principle of mathematical induction. The examples of mathematical induction. Counting principles.
5. Recurrence. Recurrence relations.
6. Relations. Equivalence relation. Ordering relations
7. Elements of combinatorics. Pascal's triangle, Newton's binomial, Stirling's numbers, Fibonacci's numbers
8. Fundamentals of graph theory.
9. Fundamentals of coding theory. Euler's theorem. Elements of cryptography.
10. Axiomatic approach to probability.
11. Cartesian product of probability space.
12. Classic probability.
13. Discrete and continuous random variable. Distribution and density of random variables. Mathematical expectation and dispersion. Gauss law.
14. Elements of mathematical statistics and its applications

Compulsory reading

1. Andrzej Grzegorczyk, An Outline of Mathematical Logic. Fundamental Results and Notions Explained with All Details, D. Reidel Publishing Company, Dordrecht-Holland/Boston-USA, PWN, Warszawa, 1974.
2. Kazimierz Kuratowski and Andrzej Mostowski, Set theory, with an Introduction to Descriptive Set Theory, North-Holland Publishing Company, PWN, Warszawa 1976
3. Helena Rasiowa, Introduction to Modern Mathematics, North-Holland, PWN, Warszawa 1973.

Recommended reading

1. T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, Introduction to Algorithms, 3/e, The MIT Press, 2009
2. Reinhard Diestel. Graph Theory, http://diestel-graph-theory.com/index.html 2012
3. Aleksandr Grińko, Michał Karpuk, Vladimir Mityushev, Vladimir Mitiouchev (junior), Natalia Ryłko, Ekonometria od podstaw z przykladami na EXCELU, Wydawnictwo Pracowni Komputerowej Jacka Skalmierskiego, Gliwice 2010

Course title	Object-oriented programming techniques		
Semester (winter/summer)	winter	ECTS	5
Lecturer(s)	Dr hab. Prof UP Piotr Czerski Mgr Piotr Kukuła		
Department	Institute of Informatics UP		

Course objectives (learning outcomes)

The aim of the course is to extend knowledge about the selected object-oriented programming language. The course will also discuss chosen design patterns.

Prerequisites

Knowledge	The student knows the basics of analysis, design and object-oriented programming and the basics of C++, Java or C#
Skills	Student can prepare and save basic algorithms and data structures in C++, Java or C#. He designs and develops basic programs in C++, Java or C# using OOP methodology.
Courses completed	Object Oriented Programming 1 and 2

Course organization							
Form of classes	W (Lecture)	Group type					
		A (large group)	K (small group)	L (Lab)	S (Seminar)	P (Project)	E (Exam)
Contact hours	15			30			1

Teaching methods:

The course consists of lectures and exercises conducted in a laboratory. Within the laboratories, students design and create preset programs in the selected object-oriented programming language, which are then discussed. Students in the individual work implement development projects that they are discussed with the teacher. Additionally students take part in activities with the use of e-learning platform.

Assessment methods:

	E – learning	Didactic games	Classes in schools	Field classes	Laboratory tasks	Individual project	Group project	Discussion participation	Student's presentation	Written assignment (essay)	Oral exam	Written exam	Other
					X			X				X	X

Assessment criteria	To assess good or very good deserves a student who can take advantage of the programs advanced solutions operating on the basis of the chosen class-containers, design patterns, and can justify their selection for specific tasks.

Comments	

Course content (topic list)

Containers: collections, lists, queues, sets, maps
Iterators,
Generic types,
Selected design patterns.

Compulsory reading

1. Bruce Eckel, "*Thinking in C++*", Pearson Education (US)
2. Bruce Eckel, "*Thinking in Java*", Prentice-Hall
3. Bjarne Stroustrup, "*The C++ Programming Language*", Addison Wesley Professional
4. Ian Griffiths, Matthew Adams i Jesse Liberty, "*Programming C#*", O'Reilly Media

Recommended reading

1. Nicolai M. Josuttis "The C++ Standard Library", Addison-Wesley U.S.A.
2. Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides "Design Patterns: Elements of Reusable Object-Oriented Software", Addison-Wesley Professional
3. Alan Shalloway, James Trott "Design Patterns Explained: A New Perspective on Object-oriented Design", Addison-Wesley Professional

Course title	Computer organization and architecture		
Semester (winter/summer)	summer	ECTS	5
Lecturer(s)	dr hab. prof. UP Piotr Czerski		
Department	Institute of Informatics UP		

Course objectives (learning outcomes)

1. Acquiring the ability to analyze and design digital circuits combinational and sequential.
2. Acquiring knowledge about the construction and operation of the exemplary computer (Von Neumann machine) with control hardware solutions from the point of view of logic.

Prerequisites

Knowledge	Elements of Boolean algebra at the high school level
Skills	They require no preliminary skills.
Courses completed	There is no need for introductory courses.

Course organization							
Form of classes	W (Lecture)	Group type					
		A (large group)	K (small group)	L (Lab)	S (Seminar)	P (Project)	E (Exam)
Contact hours	15	20					

Teaching methods:

1. Lecture - presentation in PowerPoint.
2. Exercise - analysis and design of digital combinational and sequential circuits in the simple example of a computer (von Neumann machine) with control hardware and micro programmable solutions from the point of view of logic.

Assessment methods:

	E – learning	Didactic games	Classes in schools	Field classes	Laboratory tasks	Individual project	Group project	Discussion participation	Student's presentation	Written assignment (essay)	Oral exam	Written exam	Other
								X	X				

Assessment criteria	The evaluation of a good or very good can get a student who has demonstrated adequate knowledge of design and analysis of digital combinational circuits, digital sequential circuits using flip-flops

Comments	

Course content (topic list)

1. Number Systems.
2. Fundamentals of bivalent Boolean algebra.
3. Minimize the logic functions.
4. Logic gates.
5. Design and analysis of digital combinational circuits (decoders, multiplexers, half adders, adders).
6. HDL Hardware Description Languages.
7. Designing combinational circuits using programmable logic PLD.
8. Flip-flops
9. Design and analysis of digital sequential circuits using flip-flops D-type and JK (arithmetic logic unit).
10. Registers and counters.
11. Designing an example computer-controlled arrangement.
12. Designing an example computer-controlled micro programmable.

Compulsory reading

M. Morris Mano, Charles R. Kime, Logic and Computer Design Fundamentals, 2004 Pearson Eduction. Inc..

Recommended reading

M. Morris Mano, Computer System Architecture, 1972 Prentice-Hall,

Course title	Software engineering		
seemster	winter	ECTS*	3
Lecturer(s)	Dr Inż Krzysztof Mazela		
Department	Institute of Informatics UP		

Course objectives (learning outcomes)

Genesis of software engineering and its definition and aims. The processes of software production its models , and software lifetime cycle.
Software tools and software development environments.
Requirements gathering , requirements engineering and requirements evolution, Methods and models of software : structural and object oriented analysis(eg UML language). Methods of software design . Use of Application Programming Interface (API). Introduction to Design patterns. Introduction to software validation, testing and code inspection. Introduction to software evolution. Software development management

Prerequisites

Knowledge	Basic knowledge from algorithms , operating system, embedded systems , knowledge of C++ or Java programming language.
Skills	Skills in creating simple algorithms, Object oriented programming programming skills in C++ or Java
Courses completed	Algorithms and Data structures, Creating systems. Object Oriented programming

Course organization							
Form of classes	W (Lecture)	Group type					
		A (large group)	K (small group)	L (Lab)	S (Seminar)	P (Project)	E (Exam)
Contact hours	22			33			

Teaching methods:

Lecture and Laboratory classes

Assessment methods:

	E – learning	Didactic games	Classes in schools	Field classes	Laboratory tasks	Individual project	Group project	Discussion participation	Student's presentation	Written assignment (essay)	Oral exam	Written exam	Other
										x	x		

Assessment criteria	

Comments	

Course content (topic list)

1. Software engineering , genesis, definition, aims.
2. The processes of software production its models , and software lifetime cycle.
3. Software tools and software development environments.
4. Requirements gathering , requirements engineering and requirements evolution.
5. UML.
6. Design Patterns.
7. Introduction to software validation, testing and code inspection.

Compulsory reading

1. Ian Sommervile Inżynieria Oprogramowania, WNT- wybrane fragmenty
2. K.Beck, A.Cynthia, Wydajne programowanie – Extreme
Programming, Mikom, 2005 - wybrane fragmenty
3. A. Cockburn, Jak pisać efektywne przypadki użycia, WNT, Warszawa 2004.
4. S.Covey, 7 nawyków skutecznego działania REBIS,
2002. - wybrane fragmenty
5. M.Fowler, K.Scott, UML w kropelce, LTP, 2002.
6. R. Pressman, Software Engineering, McGraw-Hill,
New York 1997

Recommended reading

Course title	Java Network Programming		
Semester (winter/summer)	winter	ECTS	4
Lecturer(s)	dr Łukasz Bibrzycki		
Department	Institute of Informatics UP		

Course objectives (learning outcomes)

The objective of the course is to provide students with basic programming skills to create the Java programs working in distributed network environments. The course also presents basic aspects of multithread and concurrent Java programming.

Prerequisites

Knowledge	Theoretical background in computer networks and network services. Basic skills in object oriented analysis, design and programming
Skills	Ability to code basic algorithms and data structures in Java. Ability to create, debug and execute object oriented programs
Courses completed	Object oriented programming

Course organization							
Form of classes	W (Lecture)	Group type					
		A (large group)	K (small group)	L (Lab)	S (Seminar)	P (Project)	E (Exam)
Contact hours	15			30			

Teaching methods:

The course is divided into lectures where theoretical aspects of multithreaded and network Java programming will be presented and laboratory exercises. In laboratory students will create Java programs using selected network programming components provided by the Java framework. Additionally students will present and discuss programs which they create both individually and in groups.

Assessment methods:

	E – learning	Didactic games	Classes in schools	Field classes	Laboratory tasks	Individual project	Group project	Discussion participation	Student's presentation	Written assignment (essay)	Oral exam	Written exam	Other
					x			x					

Assessment criteria	To obtain a good or very good score the student must: - design and implement the TCP and UDP socket based Java programs - describe selected network protocols - know the details of multithreaded programming in Java - be able to design network protocols and create programs based on them - understand the safety aspects of selected network protocols, - be able to connect from the level of the Java application to relational databases and perform basic operations on schemas and datasets.

Comments	

Course content (topic list)

1. Multithreaded programming in Java
2. Java SE API classes used for reading from and writing to files
3. Selected network protocols (echo, http, SMTP)
4. Creating client applications based on TCP sockets (Socket class)
5. Creating multithreaded Server based on TCP sockets (ServerSocket class)
6. Access remote databases trough JDBC
7. Client-server applications based on UDP sockets
8. Distributed applications based on Java RMI and IDL

Compulsory reading

1. Cay S. Horstmann, Gary Cornell, Core Java Volume I--Fundamentals (9th Edition), Prentice Hall; 9 edition (December 7, 2012)
2. Cay S. Horstmann, Gary Cornell, Core Java, Volume II--Advanced Features (9th Edition), Prentice Hall; 9 edition (March 6, 2013)

Recommended reading

1. Brian Goetz, Java Concurrency in Practice, Addison-Wesley Professional; 1 edition (May 19, 2006)

Course title	Computer simulations		
Semester (winter/summer)	summer	ECTS	2
Lecturer(s)	prof. dr hab. Władimir Mitiuszew	prof. dr hab. Władimir Mitiuszew	
Department	Institute of Informatics UP		

Course objectives (learning outcomes)

The main objective is to teach students to computer simulations of the problems on economy, biology, physics and technology. The attention is paid to practical laboratory exercises during which students work out individual and collective projects of the basic tasks in simulations. The goal is also to get knowledge in the topics of computer mathematics and applied computer sciences in order to prepare students for the third level education (PhD study).

Prerequisites

Knowledge	Foundations of mathematics (algebra, calculus, differential equations) and mathematical modeling. Data structures and basic algorithms. English language.
Skills	To work with mathematical objects: calculations of derivatives and integrals, solution to the main differential equations, vector-matrix analysis. To work with computer. To work in a group.
Courses completed	Foundation of programming, Theoretical foundations of computer sciences, Mathematics 3, Foundations of modeling and simulations, Algorithms and data structure, programming procedures

Course organization							
Form of classes	W (Lecture)	Group type					
		A (large group)	K (small group)	L (Lab)	S (Seminar)	P (Project)	E (Exam)
Contact hours	10			20			

Teaching methods:

Assessment methods:

	E – learning	Didactic games	Classes in schools	Field classes	Laboratory tasks	Individual project	Group project	Discussion participation	Student's presentation	Written assignment (essay)	Oral exam	Written exam	Other
		x			x	x	x	x	x		x	x	

Assessment criteria	Individual and team projects with use of the symbolic and numerical computations and of graphics and animation performed and checked in interactive regime

Comments	

Course content (topic list)

1. Foundations of computer simulations. Packages for numerical and symbolic computations. Numerical and symbolic computations, graphics and animation.
2. The main ordinary differential equations and their applications.
3. The main partial differential equations and their applications.
4. Computational simulations by usage of the asymptotic methods
5. Financial simulations and data analysis.

Compulsory reading

1. H. Gliński, R. Grzymkowski, A. Kapusta, D. Słota, *Mathematica 8*, Wydawnictwo Pracowni Komputerowej Jacka Skalmierskiego, Gliwice, 2012. (wybrane fragmenty)
2. V. Mityushev, W. Nawalaniec N. Ryłko, A. Malevich, *Podstawy matematyki przemysłowej,*
3. *tom 1 – „Matematyczne modelowanie i symulacje komputerowe”*, *tom 2 – „Zagadnienia wielowymiarowe”, tom 3 – „Podstawy obliczeń, przykłady”*, Wydawnictwo Pracowni Komputerowej Jacka Skalmierskiego, Gliwice, 2010.
4. A.I. Borisenko, I.E. Tarapov, *Vector and tensor analysis with applications*, Dover, 1979.
5. R. Courant, D. Hilbert, *Methods of Mathematical Physics, v. 1, 2*, Wiley-Interscience, 1989.

Recommended reading

1. *Mathematica 8. Handbook*, Wolfram Research, 2006.
2. A. Grinko, A. Karpuk, V. Mityushev (Junior), V. Mityushev, N. Ryłko, *Ekonometria od podstaw z przykładami na EXCELU*, Wydawnictwo Pracowni Komputerowej Jacka Skalmierskiego, Gliwice, 2010.
3. D. Basmadjian, R. Farnood, *The Art of Modeling in Science and Engineering with Mathematica, Second Edition*, Chapman & Hall/CRC, 2006.
4. V. Andrianov, L. I. Manevitch, *Asymptotology: Ideas, Methods, and Applications*, Kluwer Academic Publishers, 2002.
5. K.Krupa, *Modelowanie symulacja i prognozowanie Systemy ciągłe*, Wydawnictwo WNT, 2008.
6. Z. Bubnicki, *Teoria i algorytmy sterowania*, Wydawnictwo Naukowe PWN, 2005.
7. J.A. Adam, *Mathematics in Nature: Modeling Patterns in the Natural World*, Princeton University Press, 2006.
8. V. V. Mityushev; S. V. Rogosin, *Constructive Methods for Linear and Nonlinear Boundary Value Problems for Analytic Functions: Theory and Applications*, Chapman & Hall/CRC Press, Boca Raton, 2000.
9. F. F. Cap, *Mathematical Methods in Physics and Engineering with Mathematica*, Chapman & Hall/CRC 2003.

S. Mangano, *Mathematica Cookbook*, O'Reilly Media, 2010.

Course title	Foundations of modeling and simulations		
Semester (winter/summer)	summer	ECTS	3
Lecturer(s)	prof. dr hab. Władimir Mitiuszew		
Department	Institute of Informatics UP		

Course objectives (learning outcomes)

The main course objectives is to teach students to develop mathematical models from the statements of practical tasks to application. Exemplary problems arisen, for example, in informatics systems (networks), financial applications, biology, mathematics, physics and technology are discussed. The special attention is paid to practical laboratory exercisers during which the students prepare individual and collective projects on the stated problems. The goal of the course is also to get knowledge in applied mathematics and computer sciences in order to be prepared to the second level education in applied mathematics and informatics.

Prerequisites

Knowledge	Basic knowledge of calculus and algebra. Basic knowledge of data structures and algorithms. Basic knowledge of English.
Skills	Skills to operate by mathematical objects: calculation of derivatives and integrals, vector-matrix operations. Skills to use computer on the basic level. Communication and project management skills to work in a team. Foundations of programming, theoretical foundations of computer science, mathematics,
Courses completed	Mathematics 1, Mathematics 2, Foundations of programming, Theoretical foundations of computer science, Algorithms and data structures, Programming procedures, Communication and work with projects.

Course organization							
Form of classes	W (Lecture)	Group type					
		A (large group)	K (small group)	L (Lab)	S (Seminar)	P (Project)	E (Exam)
Contact hours	15			30			

Teaching methods:

Individual and group projects with usage of the symbolic-numerical computations performed and verified in the interactive regime.

Assessment methods:

	E – learning	Didactic games	Classes in schools	Field classes	Laboratory tasks	Individual project	Group project	Discussion participation	Student's presentation	Written assignment (essay)	Oral exam	Written exam	Other
		x			x	x	x	x			x	x	

Assessment criteria	The exam mark is based first of all on the individual projects and partially on the group projects and on the oral and written answers on the main theoretical and practical questions (exams oral and written). The mark "good" and "very good" can be got by a student who can: • Develop a mathematical model • Propose methods of solutions or of description of phenomena • Perform acting computer simulations • Formulate the proper results and perhaps correct the mathematical model • Present the result of computer simulations in visual form including graphs, animations, can make conclusions on the basis of her/his results and can give practical recommendations

Comments	

Course content (topic list)

1. Development of the mathematical models. Continuous and discrete, linear and non-linear, deterministic and stochastic models. Elements of dimensional analysis.
2. Application of symbolic and numerical computations to mathematical treatment of the task, graphics, animation.
3. Newton's method. Banach Fixed Point Theorem and its applications to differential and integral equations.
4. Construction and application of the ordinary differential equations and their applications.
5. Least Square Method.
6. Small parameter methods, perturbation methods, polynomial and rational approximations and their implementation.
7. Special functions, integral representations, numerical and symbolic computations
8. Introduction of delta-function
9. Series and polynomials. Pade approximation.
10. Implementation of various objects.

Compulsory reading

1. H. Gliński, R. Grzymkowski, A. Kapusta, D. Słota, *Mathematica 8*, Wydawnictwo Pracowni Komputerowej Jacka Skalmierskiego, Gliwice, 2012.
2. S. Mangano, *Mathematica Cookbook*, O'Reilly Media, 2010.
3. V. Mityushev, W. Nawalaniec N. Ryłko, A. Malevich, *Podstawy matematyki przemysłowej, tom 1 – „Matematyczne modelowanie i symulacje komputerowe”*, *tom 2 – „Zagadnienia wielowymiarowe”, tom 3 – „Podstawy obliczeń, przykłady”*, Wydawnictwo Pracowni Komputerowej Jacka Skalmierskiego, Gliwice, 2010.

Recommended reading

10. *Mathematica 8. Handbook*, Wolfram Research, 2006.
11. A. Grinko, A. Karpuk, V. Mityushev (Junior), V. Mityushev, N. Ryłko, *Ekonometria od podstaw z przykładami na EXCELU*, Wydawnictwo Pracowni Komputerowej Jacka Skalmierskiego, Gliwice, 2010.
12. D. Basmadjian, R. Farnood, *The Art of Modeling in Science and Engineering with Mathematica, Second Edition*, Chapman & Hall/CRC, 2006.
13. V. Andrianov, L. I. Manevitch, *Asymptotology: Ideas, Methods, and Applications*, Kluwer Academic Publishers, 2002.
14. A.I. Borisenko, I.E. Tarapov, *Vector and tensor analysis with applications*, Dover, 1979.
15. K.Krupa, *Modelowanie symulacja i prognozowanie Systemy ciągłe*, Wydawnictwo WNT, 2008.
16. Z. Bubnicki, *Teoria i algorytmy sterowania*, Wydawnictwo Naukowe PWN, 2005.
17. J.A. Adam, *Mathematics in Nature: Modeling Patterns in the Natural World*, Princeton University Press, 2006.
18. V. V. Mityushev; S. V. Rogosin, *Constructive Methods for Linear and Nonlinear Boundary Value Problems for Analytic Functions: Theory and Applications*, Chapman & Hall/CRC Press, Boca Raton, 2000.
19. R. Courant, D. Hilbert, *Methods of Mathematical Physics, v. 1, 2*, Wiley-Interscience, 1989.
20. F. F. Cap, *Mathematical Methods in Physics and Engineering with Mathematica*, Chapman & Hall/CRC 2003.

Course title	Numerical analysis		
Semester (winter/summer)	summer	ECTS	4
Lecturer(s)	prof. dr hab. Władimir Mitiuszew		
Department	Institute of Informatics UP		

Course objectives (learning outcomes)

The main objective is to teach students to use modern numerical methods to various partial differential equations and to apply them to practical problems. Student can state properly a boundary value problem and an initial problem, to present a method to solve it (finite difference method, finite element method). Student can implement a computer method of solution, represent it and explain practical results of his/her computations.

Prerequisites

Knowledge	Foundations of mathematics (algebra, calculus, differential equations) and mathematical modeling. Data structures and basic algorithms. English language.
Skills	To work with mathematical objects: calculations of derivatives and integrals, solution to the main differential equations, vector-matrix analysis. To work with computer. To work in a group.
Courses completed	Foundation of programming, Theoretical foundations of computer sciences, Mathematics 3, Foundations of modeling and simulations, Algorithms and data structure, programming procedures

Course organization							
Form of classes	W (Lecture)	Group type					
		A (large group)	K (small group)	L (Lab)	S (Seminar)	P (Project)	E (Exam)
Contact hours	10			20			

Teaching methods:

Assessment methods:

	E – learning	Didactic games	Classes in schools	Field classes	Laboratory tasks	Individual project	Group project	Discussion participation	Student's presentation	Written assignment (essay)	Oral exam	Written exam	Other
		x			x	x	x	x	x		x	x	

Assessment criteria	Individual and team projects with use of the symbolic and numerical computations and of graphics and animation performed and checked in interactive regime

Comments	

Course content (topic list)

6. General notation of the mathematical physics equations.
7. Types of the partial differential equations.
8. Boundary value problems and initial problems.
9. Foundations of finite differences method.
10. Foundations of finite element method.
11. Foundations of the collacation method
12. Computational methods to solve boundary value problems and initial problems. Packets for numerical computations.
13. Optimalization problems.
14. Application of the perturbations and asymptotic methods.

Compulsory reading

1. V. Mityushev, W. Nawalaniec N. Ryłko, A. Malevich, *Podstawy matematyki przemysłowej, tom 1 – „Matematyczne modelowanie i symulacje komputerowe”*, *tom 2 – „Zagadnienia wielowymiarowe”, tom 3 – „Podstawy obliczeń, przykłady”*, Wydawnictwo Pracowni Komputerowej Jacka Skalmierskiego, Gliwice, 2010.
2. A.A. Samarski, Wstęp do teorii schematów różnicowych, Nauka, Moskwa 1971.
3. Grzymkowski, R., Słota, D.: Computational Methods for Integral Equations. \'Sl\k{a}sk Technological University Publ., Gliwice (2015)
4. Kincaid, D, Cheney, W.: Numerical Analysis. Mathematics of Scientific Computing. 3d edition. AMS, Rhode Island (2009)
5. Pinsky, M.A.: Partial Differential Equations and Boundary-Value Problems with Applications. Third Edition. AMS, Rhode Island (1998)

Recommended reading

1. A.I. Borisenko, I.E. Tarapov, *Vector and tensor analysis with applications*, Dover, 1979.
2. V. V. Mityushev; S. V. Rogosin, *Constructive Methods for Linear and Nonlinear Boundary Value Problems for Analytic Functions: Theory and Applications*, Chapman & Hall/CRC Press, Boca Raton, 2000.

3. R. Courant, D. Hilbert, *Methods of Mathematical Physics, v. 1, 2*, Wiley-Interscience, 1989.
4. Andrianov, I.V., Manevitch, L.I.: Asymptotology: Ideas, Methods, and Applications. Kluwer Academic Publishers, Dordrecht etc (2002)
5. Ghoniem, N., Walgraef, D.: Instabilities and Self-Organization in Materials. V. 1--2, Oxford Univ. Press, Oxford (2008)
6. Lawler, G.F.: Random Walk and the Heat Equation. AMS, Rhode Island (2010)
7. Logan, J.D.: Applied Mathematics. Wiley, 4th Edition, Hoboken (2013)

Course title	Developing Mobile Applications		
Semester (winter/summer)	winter	ECTS	4
Lecturer(s)	dr inż. Łukasz Bibrzycki mgr inż. Piotr Kukuła		
Department	Institute of Informatics UP		

Course objectives (learning outcomes)

The objective of the course is to provide basic information on developing mobile applications for Android platform and discuss tools used for developing, testing and debugging applications for Android platform

Prerequisites

Knowledge	Working knowledge of the object oriented programming
Skills	Ability to write object oriented programs in Java programming language
Courses completed	

Course organization							
Form of classes	W (Lecture)	Group type					
		A (large group)	K (small group)	L (Lab)	S (Seminar)	P (Project)	E (Exam)
Contact hours	10			30			

Teaching methods:

The course has a form of laboratory exercises where students create projects in groups or individually. The laboratory is accompanied by lectures where general aspects of programming for Android Framework are presented.

Assessment methods:

	E – learning	Didactic games	Classes in schools	Field classes	Laboratory tasks	Individual project	Group project	Discussion participation	Student's presentation	Written assignment (essay)	Oral exam	Written exam	Other
					x			x					

Assessment criteria	Good or very good score can be obtained by a student who presents exceptional skills in creating applications for Android platform.

Comments	

Course content (topic list)

1. Introduction to mobile operating systems
2. Java for the Android platform – specific solutions
3. Development tools
4. GUI on Android platform
5. Application development with Android SDK
6. Using sensors in applications (GPS, accelerometer, megnetometer, etc.)
7. Basics of OpenGL ES
8. Creating native applications with Android NDK
9. Application publishing with Google Play

Compulsory reading

1. Charlie Collins, Michael Galpin, Matthias Kaeppler, Android in Practice 1st Edition, Manning Publications; 1 edition , 2011
2. Selected subjects from: http://developer.android.com

Recommended reading

1. Greg Milette, Adam Stroud, Professional Android Sensor Programming 1st Edition, Wrox; 1 edition 2012

Course title	Web Applications		
Semester (winter/summer)	winter	ECTS	4
Lecturer(s)	Dr. hab. Martin Malčík		
Department	Institute of Informatics UP		

Course objectives (learning outcomes)

The aim of the course is to familiarize students with the technology to create web applications using framework Ruby on Rails.
Outcomes:
Knowledge of Ruby programming
Knowledge of Rails usage
Skills in programming in Ruby
Skills in creating web application in Rails

Prerequisites

Knowledge	It can be useful general knowledge of programming in scripting languages such as Perl or Python. Basics of object-oriented programming. Basic knowledge of HTML and CSS. General knowledge of the architecture of web applications (presentation layer, web server, database).
Skills	Designing simple websites in HTML and CSS. Designing simple relational databases. Implementation of applications using object-oriented languages (ability to design and implement classes, their methods, unit tests).
Courses completed	

Course organization							
Form of classes	W (Lecture)	Group type					
		A (large group)	K (small group)	L (Lab)	S (Seminar)	P (Project)	E (Exam)
Contact hours	10			30			

Teaching methods:

Demonstration
Life programming
Explanation
Practical exercises
Presentation

Assessment methods:

	E – learning	Didactic games	Classes in schools	Field classes	Laboratory tasks	Individual project	Group project	Discussion participation	Student's presentation	Written assignment (essay)	Oral exam	Written exam	Other
	2					1						1	

Assessment criteria	E-Learning test 1– more than 90 % in order to pass E-Learning test 2 – more than 90 % in order to pass Written test – functional program in Ruby Group Project – Successful presentation and explanation of the project

Comments	

Course content (topic list)

1. Basics of the programming language Ruby
◦ Structure and run programs
◦ The data types, expressions and operators
◦ Instructions and flow control, exception handling
◦ Classes and Objects
◦ Files
◦ Functions
◦ Regular Expressions

2. Creating of a Web application - Rails
◦ Installation and lauching
◦ Architecture and philosophy of the framework
◦ URL, HTTP service and views
◦ Templates and forms processing
◦ Sample Applications
◦ Advanced features and mechanisms
◦ Advanced application deployment
◦ Tools for application development
◦ Support mechanisms automated testing applications

Compulsory reading

Huw Collingbourne - The Little Book of Ruby, 4th Edition. Open source

Michal Hartl – Learn Web Development with Rails. https://www.railstutorial.org/book. Read online free.

Recommended reading

Huw Collingbourne - The Book of Ruby, 4th Edition.
Chris Pine – Learn to Program, The Facets of Ruby Series
The Rails Style - Guidehttps://github.com/bbatsov/rails-style-guide
The Ruby Style Guide - https://github.com/bbatsov/ruby-style-guide#syntax
Dan Saffer – Microinteractions. Designing with Details. O´Reilly, 2013.
R.C. Martin - „Czysty kod : podręcznik dobrego programisty", Helion 2010
Geoffrey James – The Tao of Programming

Course title	Multimedia Technology and its Applications		
semester	summer	ECTS*	1
Lecturer(s)	mgr Alicja Pituła		
Department	Chair of Computer Science and Computational Methods		

Course objectives (learning outcomes)

The goal of the course is to prepare students to use various multimedia tools in order to create multimedia communication systems.

Prerequisites

Knowledge	-
Skills	-
Courses completed	-

Course organization							
Form of classes	W (Lecture)	Group type					
		A (large group)	K (small group)	L (Lab)	S (Seminar)	P (Project)	E (Exam)
Contact hours				15			

Teaching methods:

Blended-learning course.
Laboratory exercises are held in the computer lab.
Student gets the access to the e-learning platform with online learning materials.

Assessment methods:

	E – learning	Didactic games	Classes in schools	Field classes	Laboratory tasks	Individual project	Group project	Discussion participation	Student's presentation	Written assignment (essay)	Oral exam	Written exam	Other
	X				X	X	X	X	X				

Assessment criteria	Best assessements grades can get a student that creates advanced, interactive multimedia application.

Comments	

Course content (topic list)

1. Webwriting and webusability.
2. Elements of copyright. Creative Commons.
3. Repositories of images, music and video samples.
4. Cloud computing. SAAS.
5. Working with various online apllications.
6. Screencasts and tutorials making.
7. Online publication, self-publishing systems.
8. Working with audio and video materials.
9. Multimedia presentations.
10. Visualization od data.

Compulsory reading

On-line materials and workbooks

Recommended reading

Course title	Advanced web design 2		
Semester (winter/summer)	summer	ECTS	4
Lecturer(s)	Piotr Kukuła		
Department	Institute of Informatics UP		

Course objectives (learning outcomes)

The aim of the course is to acquaint students with advanced mechanisms and techniques used in the process of designing and web development. The course will cover the creation of web services based on open development platforms (frameworks) using scripting languages based on the MVC pattern.

Prerequisites

Knowledge	Procedural and object-oriented programming. Basic web technologies (HTML / XHTML, CSS, JS) and subsidiaries (PHP). Relational databases.
Skills	Basics of programming in PHP and JavaScript. Basics of SQL.
Courses completed	Object-oriented programming 3. Web applications. Advanced web design 1.

Course organization							
Form of classes	W (Lecture)	Group type					
		A (large group)	K (small group)	L (Lab)	S (Seminar)	P (Project)	E (Exam)
Contact hours	10			30			

Teaching methods:

The course is in the form of laboratory classes, during which students work on projects related to the construction of websites using the selected development platform (Framework).
During the course, students receive individual assignment problem (project, paper) and one practical project requiring a comprehensive approach to design, implementation and launch a website based on the selected software platform (Framework) operating in the MVC / MVT.

Assessment methods:

	E – learning	Didactic games	Classes in schools	Field classes	Laboratory tasks	Individual project	Group project	Discussion participation	Student's presentation	Written assignment (essay)	Oral exam	Written exam	Other
					X	X	X	X	X				

Assessment criteria	The final grade depends on the ratings of partial and systematic performed tasks and work on projects. In particular, the assessment of good or very good can get a student who will finish the individual and the group project on time with positive results.

Comments	The course is discussed chosen by leading framework using PHP (recommended) or Python (if the level of the group is sophisticated enough). Group projects students may be (with the consent of leading) realized based on other than discussed in class framework implementing MVC pattern.

Course content (topic list)

5. Design Patterns type MVC.
6. Importance of layers (model, view and controller)
7. Project platforms (frameworks) to support the creation of Web sites based on MVC patterns: the installation and configuration; building, structure and performance; implementations of scripting languages; work in developer environment; database access,
8. Open licenses and development platforms used in constructing websites

Compulsory reading

5. Chris Pitt, „Pro PHP MVC", Apress
6. Oleg Krivtsov, "Using Zend Framework 2", Leanpub
7. Michael Romer, "Web Development with Zend Framework 2", Leanpub
8. http://framework.zend.com/

Recommended reading

1. J. Forcier, P. Bissex, W. Chun, "Python Web Development with Django", Addison-Wesley Professional
2. E. Freeman, E. Freeman, B. Bates, K. Sierra, "Head First Design Patterns", O'Reilly
3. http://pl.php.net/
4. http://www.w3.org/
5. http://www.python.org/

Course title	Advanced web technologies 1		
Semester (winter/summer)	summer	ECTS	3
Lecturer(s)	Jozef Kapusta		
Department	Institute of Computer Science		

Course objectives (learning outcomes)

This course will cover advanced Web coding concepts and teach students how to add an extra layer of usability to a Web page using a current scripting language or tool. Students will learn how to create accessible modern web applications that integrate current Web standards. Course content will include Web Grid Systems, CSS Framework, JavaScript and JQuery technologies, AJAX, PHP and other related web techniques and technologies.

Prerequisites

Knowledge	Procedural and object programming. Basic web technologies (HTML / XHTML, CSS) and foundations of technologies for server side scripting (PHP). Relational databases, SQL language.
Skills	HTML, CSS programming, PHP programming foundations, SOL language
Courses completed	Object-oriented Programming. Internet Applications

Student Learning Outcomes

	Learning Outcomes
Knowledge	W01: Upon a successful completion of the course, the students will be able to understand and define own grid systems on web pages. W02: Students will be able to understand and use the CSS Framework technologies. W03: Students will learn to program in a current client-side scripting language W04: Students will be able to describe how server-side and client-side technologies can cooperate. W05: Students will be able to design and implement user interfaces based on the AJAX technology.

Skills	Learning Outcomes
	U01: Students will learn how to bring together client-side and server-side scripting to create a real-world solution for a contemporary web site. U02: Students will use fundamental skills to design modern web pages. U03: Students will know select and apply web technologies for processing, and presenting information in web pages. U04: Students will use JavaScript Language and JQuery to transfer data and add interactive components to the web pages.

Social competence	Learning Outcomes
	K01: Students will gain experience working with teammates on a web design projects.

Course organization													
Form of classes	W (Lecture)	Group type											
		A (large group)		K (small group)		L (Lab)		S (Seminar)		P (Project)		E (Exam)	
Contact hours						25							

Teaching methods:

The course is conducted in the form of laboratory classes. The students work on simple projects during the classes. Projects are aimed to CSS Frameworks, JavaScript, PHP and other related web techniques and technologies.

Each student will work on the web development project, which should be completed to the end of the semester. This final project for the course involves a suitable combination of web technologies learned during the semester.

Assessment methods:

	E – learning	Didactic games	Classes in schools	Field classes	Laboratory tasks	Individual project	Group project	Discussion participation	Student's presentation	Written assignment (essay)	Oral exam	Written exam	Other
					x	x	x	x	x				

Assessment criteria	The final grade dependents on the evaluations of partial work on the simple projects at classes and the quality of final projects. The final project is aimed to CSS Frameworks, JavaScript, PHP and other related web techniques and technologies. The project files must be based on the consistent design rules. All project files must be validated with the current W3C requirements and must be submitted to the virtual learning environment (Moodle) on time.

Comments	

Course content (topic list)

1. Responsive Web Design, Elements of Responsive Web Design, Web Page Grid System
2. CSS Frameworks
3. BootStrap, CSS settings and fundamental HTML elements styled at BootStrap, Bootstrap Components, Bootstrap JavaScript Plugins
4. Bootstrap Examples and Applications, Bootstrap Collections of Shortcut
5. Javascript, Document Object Model Scripting, JQuery, Selectors in JQuery, JQuery Effects
6. PHP Examples and Applications
7. AJAX, Asynchronous web communication, XMLHttpRequest Object, AJAX in JQuery, AJAX Applications

Compulsory reading

1. http://getbootstrap.com
2. http://www.w3schools.com/bootstrap/
3. https://www.w3schools.com/jquery/
4. https://www.w3schools.com/js/js_ajax_intro.asp
5. https://www.w3schools.com/jquery/jquery_ajax_intro.asp
6. Z. Kessin. HTML5. Programowanie aplikacji, Wyd. Helion, 2012

Recommended reading

1. D. S. McFarland, JavaScript & jQuery. Nieoficjalny podręcznik, Wyd. Helion, Gliwice 2012
2. A. MacCaw, JavaScript. Aplikacje WWW, Wyd. helion 2012
3. S. Stefanov, JavaScript. Wzorce, Wyd. helion 2012
4. J. Eichorn, AJAX i JavaScript. Tworzenie i optymalizacja aplikacji sieciowych, Wyd. helion, 2007