STUDY MODULE DESCRIPTION FORM

Name of the module/subject	Code
Technologies of Software Development	**1010512311010517176**

Field of study	Profile of study (general academic, practical)	Year /Semester
Computing	**general academic**	**1 / 1**
Elective path/specialty: **Software Engineering**	Subject offered in: **Polish**	Course (compulsory, elective): **obligatory**
Cycle of study: **Second-cycle studies**	Form of study (full-time,part-time): **full-time**	
No. of hours: Lecture: **30** Classes: **-** Laboratory: **30** Project/seminars: **15**		No. of credits: **6**
Status of the course in the study program (Basic, major, other): **major**	(university-wide, from another field): **from field**	
Education areas and fields of science and art: **technical sciences**; **Technical sciences**		ECTS distribution (number and %): **6 100%**; **6 100%**

Responsible for subject / lecturer:

Jerzy Nawrocki
email: jerzy.nawrocki@put.poznan.pl
tel. +48 61 6652980
Wydział Informatyki
ul. Piotrowo 2, 60-965 Poznań

Responsible for subject / lecturer:

Michał Maćkowiak
email: michal.mackowiak@put.poznan.pl
tel. +48 61 6652944
Wydział Informatyki
ul. Piotrowo 2, 60-965 Poznań

Prerequisites in terms of knowledge, skills and social competencies:

1	**Knowledge**	Student starting this module should have a basic knowledge regarding basic algorithms and computational complexity, object-oriented programming, design patterns, databases, software testing and web applications.
2	**Skills**	Should have skills allowing solving basic problems related to requirements analysis, creating software specification, designing systems and skills that are necessary to acquire information from given sources of information.
3	**Social competencies**	Student should understand the need to extend his/her competences / has the willingness to work in a team. In addition, with respect to the social skills, the student should demonstrate such attitudes as honesty, responsibility, perseverance, curiosity, creativity, manners, and respect for other people.

Assumptions and objectives of the course:

1. Provide students knowledge regarding .NET Framework and corresponding technologies, creating websites using Ruby on Rails framework, scripting, dynamic, functional, distributed, cloud programming.
2. Develop students' skills in solving problems related to creating application using different technologies
3. Present students a set of development technologies for modeling data layer, designing interface layer, defining communication layer between several applications
4. Develop students' teamwork skills in the context of developing software systems
5. Develop students' skills to learn new technologies

Study outcomes and reference to the educational results for a field of study

Knowledge:

1. has advanced and detailed knowledge related to selected areas of computer science, developing web applications, rich user interface applications, scripts - [K2st_W3]
2. has knowledge about new technologies in the area of software development - [K2st_W4]
3. has advanced and detailed knowledge regarding software life cycle which involves developing a software system and testing it - [K2st_W5]

Skills:

1. is able to acquire, combine, interpret and evaluate information from literature, databases and other information sources (in mother tongue and English); draw conclusions, and formulate opinions based on i - [K2st_U1]
2. is able to combine knowledge from different areas of computer science (and if necessary from other scientific disciplines) to formulate and solve engineering tasks related to software development - [K2st_U5]
3. is able to assess usefulness and possibility of employing new developments (methods and tools) - [K2st_U6]
4. is able to design and develop a web application using a database - [K2st_U10]
5. is able to design (according to a provided specification which includes also non-technical aspects) a software system using technologies learned during the course - [K2st_U11]
6. is able to work in a group, performing a role of developer - [K2st_U15]

Social competencies:

1. understands that knowledge and skills related to computer science quickly become obsolete - [K2st_K1]
2. knows how new development technologies and tools could be helpful to solve practical problems like developing a web application - [K2st_K2]

Assessment methods of study outcomes

Formative assessment:

a) lectures:

* based on the answers to the questions which test understanding of material presented on the lectures

b) laboratory classes / tutorials / projects / seminars:

* based on the assessment of the tasks done during classes and as a homework

Summative assessment:

a) verification of assumed learning objectives related to lectures:

* assessment of knowledge and skills, examined by an oral test with open questions. Student can gain 100 points, to pass minimum 50 points are needed

* the final grade is determined using the following scale:

- (90%, 100%] ? 5.0
- (80%, 90%] ? 4.5
- (70%, 80%] ? 4.0
- (60%, 70%] ? 3.5
- (50%, 60%] ? 3.0
- (0%, 50%] ? 2.0

* discussing the results of the examination

b) verification of assumed learning objectives related to laboratory classes / tutorials / projects / seminars:

* assessment of student's preparation to particular laboratory classes and assessment of student's skills needed to realize tasks on these classes
* continuous assessment of student's work during classes - rewarding ability to use learned principles and methods
* assessment of projects realization, including ability to work in team

Possibility to gain additional points by activity on classes:

* elaboration of additional aspects regarding the subject
* effectiveness of applying acquired knowledge to solve problems
* ability to cooperate with the team during solving problems
* providing additional remarks for the lecturer how to improve teaching materials
* highlighting the problems with students' perception to improve the teaching process

Course description

Introduction to .NET Framework. Queries in LINQ. Object relational model in Entity Framework. Graphical user interfaces using Windows Presentation Foundation. Functional programming with F#. Dynamic programming with Ruby. Rapid development of web applications using Rails. Cloud applications using Windows Azure. Distributed programming using Akka.NET. Web development using ASP.NET. Scripting programming using Powershell.

The course consists of fifteen 2-hour laboratory classes and it starts with an instructional session at the beginning of a semester. Students work individually or in teams of 2-4.

Basic bibliography:

1. L. Bass, P. Clements, R. Kazman, "Software architecture in practice", WNT
2. P. Kruchten, "The Rational Unified Process-An Introduction", Addison-Wesley
3. A. Troelsen, P. Japikse, "C# 6.0 and the .NET 4.6 Framework", Apress
4. D. Syme, A. Granicz, A. Cisternino, "Expert F# 4.0", Apress

Additional bibliography:	
Result of average student's workload	
Activity	**Time (working hours)**
1. participating in laboratory classes / tutorials: 15 x 2 hours	30
2. participating in project classes: 15 * 1 hour	15
3. participating in lecture: 15 x 2 hours	30
4. consulting issues related to the subject of the course; especially related to t laboratory classes and projects,	2
5. implementing, running and verifying software application(s) (in addition to laboratory classes) 4. participating in lectures	25
6. studying literature / learning aids (10 pages = 1 hour), 150 pages	150
7. preparing to and participating in exams: 15 hours + 2 hours	17
8. discussing the results of the examination	1

Student's workload		
Source of workload	**hours**	**ECTS**
Total workload	145	6
Contact hours	78	3
Practical activities	70	3