COURSE DESCRIPTION CARD - SYLLABUS

Course name
Technologies of Software Development [S2Inf1E-IO>TRO]

Course

Field of study
Computing

Year/Semester
1/1

Area of study (specialization)
Software Engineering

Profile of study
general academic

Level of study
second-cycle

Course offered in
English

Form of study
full-time

Requirements
compulsory

Number of hours

Lecture
30

Laboratory classes
0

Other
0

Tutorials
0

Projects/seminars
30

Number of credit points

6,00

Coordinators

dr inż. Sylwia Kopczyńska
sylwia.kopczynska@put.poznan.pl

mgr inż. Michał Maćkowiak
michal.mackowiak@put.poznan.pl

Lecturers

Prerequisites

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. 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. Student should understand the need to extend his/her competences / has the willingness to work in a team.

Course objective

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

Course-related learning outcomes

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

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 it,
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,
3. is able to design and develop a web application using a database,
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,
6. is able to work in a group, performing a role of developer.

Social competences:
1. understands that knowledge and skills related to computer science quickly become obsolete,
2. knows how new development technologies and tools could be helpful to solve practical problems like developing a web application.

Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

Learning outcomes presented above are verified as follows:
1. Formative assessment:
a) lectures: based on the answers to the questions which test understanding of material presented on the lectures
b) project: based on the assessment of the tasks done during classes and as a homework
2. Summative assessment:
a) verification of assumed learning objectives related to lectures within a written test. 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.
b) verification of assumed learning objectives related to the project is based on:
- verification of the tasks and projects to be done individually or in a team
- project output, and approach to realize it, including the skills to work in team.
Student can score maximum 100 points, and 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.

Programme content

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.

Course topics

1. Organizational meeting – presentation of the rules and requirements for completing the course.

2. A series of lectures and workshops on selected technologies used in software development. Each lecture presents theoretical aspects as well as practical ways of applying the technology. During the workshops, students complete tasks related to the discussed technologies.

3. Team-based project implementation – enables students to learn new technologies and apply those they already know. The project is developed in increments; after each increment, students present their results and share the experience they have gained.

Teaching methods

Lectures: multimedia presentation with the examples presented on a whiteboard
Project: multimedia presentation, examples presented on a whiteboard, tasks given by the tutor to the students

Bibliography

Basic
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

Breakdown of average student's workload

	Hours	ECTS
Total workload	150	6,00
Classes requiring direct contact with the teacher	60	2,50
Student's own work (literature studies, preparation for laboratory classes/ tutorials, preparation for tests/exam, project preparation)	90	3,50