

KARTA OPISU MODUŁU KSZTAŁCENIA		
Nazwa modułu/przedmiotu Technologie rozwoju oprogramowania		Kod 1010512311010517176
Kierunek studiów Informatyka	Profil kształcenia (ogólnoakademicki, praktyczny) ogólnoakademicki	Rok / Semestr 1 / 1
Ścieżka obieralności/specjalność Software Engineering (Inżynieria)	Przedmiot oferowany w języku: polski	Kurs (obligatoryjny/obieralny) obligatoryjny
Stopień studiów: II stopień	Forma studiów (stacjonarna/niestacjonarna) stacjonarna	
Godziny Wykłady: 30 Ćwiczenia: - Laboratoria: 30 Projekty/seminaria: 15		Liczba punktów 6
Status przedmiotu w programie studiów (podstawowy, kierunkowy, inny) (ogólnouczelniany, z innego kierunku) kierunkowy z danego kierunku		
Obszar(y) kształcenia i dziedzina(y) nauki i sztuki nauki techniczne nauki techniczne		Podział ECTS (liczba i %) 6 100% 6 100%
Odpowiedzialny za przedmiot / wykładowca: Odpowiedzialny za przedmiot / wykładowca:		
Jerzy Nawrocki email: jerzy.nawrocki@put.poznan.pl tel. +48 61 6652980 Wydział Informatyki ul. Piotrowo 2, 60-965 Poznań		Michał Maćkowiak email: michal.mackowiak@put.poznan.pl tel. +48 61 6652944 Wydział Informatyki ul. Piotrowo 2, 60-965 Poznań
Wymagania wstępne w zakresie wiedzy, umiejętności, kompetencji społecznych:		
1	Wiedza:	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	Umiejętności:	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	Kompetencje społeczne	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.
Cel przedmiotu:		
<ol style="list-style-type: none"> 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 		
Efekty kształcenia i odniesienie do kierunkowych efektów kształcenia		
Wiedza:		
<ol style="list-style-type: none"> 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] 		
Umiejętności:		

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 - [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]
Kompetencje społeczne:
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]

Sposoby sprawdzenia efektów kształcenia
<p>Formative assessment:</p> <p>a) lectures:</p> <ul style="list-style-type: none">* based on the answers to the questions which test understanding of material presented on the lectures <p>b) laboratory classes / tutorials / projects / seminars:</p> <ul style="list-style-type: none">* based on the assessment of the tasks done during classes and as a homework <p>Summative assessment:</p> <p>a) verification of assumed learning objectives related to lectures:</p> <ul style="list-style-type: none">* 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:<ul style="list-style-type: none">- (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 <p>b) verification of assumed learning objectives related to laboratory classes / tutorials / projects / seminars:</p> <ul style="list-style-type: none">* 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 <p>Possibility to gain additional points by activity on classes:</p> <ul style="list-style-type: none">* 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
Treści programowe
<p>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.</p> <p>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.</p>
Literatura podstawowa:
<ol style="list-style-type: none">1. L. Bass, P. Clements, R. Kazman, "Software architecture in practice", WNT2. P. Kruchten, "The Rational Unified Process-An Introduction", Addison-Wesley3. A. Troelsen, P. Japikse, "C# 6.0 and the .NET 4.6 Framework", Apress4. D. Syme, A. Granicz, A. Cisternino, "Expert F# 4.0", Apress

Literatura uzupełniająca:		
Bilans nakładu pracy przeciętnego studenta		
Czynność	Czas (godz.)	
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 laboratory classes and projects,	25	
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), 250 pages	17	
7. preparing to and participating in exams	1	
8. discussing the results of the examination		
Obciążenie pracą studenta		
forma aktywności	godzin	ECTS
Łączny nakład pracy	145	6
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	78	3
Zajęcia o charakterze praktycznym	70	3