

SYLABUS ZAJ /GRUPY ZAJ

Dane ogólne:

Jednostka organizacyjna:	Katedra Informatyki				
Kierunek studiów:	Informatyka				
Specjalno /Specjalizacja:	Inżynieria oprogramowania				
Nazwa zaj / grupy zaj :	Testowanie i jako oprogramowania				
Course / group of courses:	Software Testing Quality				
Forma studiów:	stacjonarne				
Nazwa katalogu:	WP-IN-I-20/21Z-IO				
Nazwa bloku zaj :					
Kod zaj /grupy zaj :	105914	Kod Erasmus:			
Punkty ECTS:	3	Rodzaj zaj :		obowi zkowy	
Rok studiów:	3	Semestr:		6	
Rok	Semestr	Forma zaj	Liczba godzin	Forma zaliczenia	ECTS
3	6	LO	30	Zaliczenie z ocen	2
		W	15	Zaliczenie z ocen	1
Razem			45		3
Koordinator:	magister Tomasz G dek				
Prowadz cy zaj cia:					
J zyk wykładowy:	semestr: 6 - j zyk polski				

Obja nienia:

Rodzaj zaj : obowi zkowe, do wyboru.

Forma prowadzenia zaj : W - wykład, - wiczenia audytoryjne, L - lektorat, S – seminarium/ zaj cia seminaryjne, P - wiczenia praktyczne (w tym zaj cia wf), M - wiczenia specjalistyczne (medyczne/ kliniczne), LO – wiczenia laboratoryjne, LI - laboratorium informatyczne, ZTI - zaj cia z technologii informacyjnych, P – wiczenia projektowe, ZT – zaj cia terenowe, T - wiczenia terenowe na obozach programowych, SK - samokształcenie (i inne), PR - praktyka zawodowa

Dane merytoryczne

Wymagania wst pne:			
Znajomo paradygmatów programowania, umiej tno programowania obiektowego, znajomo podstawowych technologii programowania aplikacji webowych, znajomo baz danych.			
Szczegółowe efekty uczenia si			
Lp.	Student, który zaliczył zaj cia zna i rozumie/potrafi/jest gotowy do:	Kod efektu dla kierunku studiów	Sposób weryfikacji efektu uczenia si
1	Zna, rozumie koncepcje i dobre praktyki testowania jednostkowego.	IN1_W03	kolokwium
2	Potrafi zweryfikowa jako pokrycia kodu testami.	IN1_W03	kolokwium
3	Potrafi wskaza ró nic pomi dzy testami jednostkowymi a integracyjnymi.	IN1_W04	kolokwium
4	Potrafi tworzy kod dobrej jako zgodnie z przyj tymi standardami.	IN1_W07, IN1_W04	kolokwium

5	Zna i rozumie zagadnienie ci głęj integracji.	IN1_W08	kolokwium
6	Zna i rozumie zagadnienie ci głęj integracji.	IN1_U02	kolokwium
7	Potrafi tworzy kod dobrej jako zgodnie z przyj tymi standardami.	IN1_U05	kolokwium
8	Zna, rozumie koncepcje i dobre praktyki testowania jednostkowego.	IN1_K01	kolokwium
9	Potrafi tworzy kod dobrej jako zgodnie z przyj tymi standardami.	IN1_K01	kolokwium
Stosowane metody osi gania zakładanych efektów uczenia si (metody dydaktyczne)			
metody podaj ce (Przedstawienie tre ci kształcenia w oparciu o prezentacje multimedialn oraz analiz i omawianie praktycznych przykładów.), metody praktyczne (wiczenia laboratoryjne prowadzone w oparciu o przygotowane instrukcje (udost pnione w sposób elektroniczny) z przykładami i zadaniami do samodzielnej realizacji, które podlegaj ocenie.)			
Kryteria oceny i weryfikacji efektów uczenia si			
wiedza: ocena kolokwium (ocena kolokwium) umiejętności: ocena kolokwium (ocena kolokwium) kompetencje społeczne: ocena kolokwium (ocena kolokwium)			
Warunki zaliczenia			
Wykład: Zaliczenie na podstawie kolokwium. Laboratorium: Zaliczenie z ocen wystawion na podstawie pozytywnego zaliczenia kolokwium z zada , aktywno na zaj ciach (rozwi zywanie zada i problemów).			
Tre ci programowe (opis skrócony)			
1. Wprowadzenie do wersjonowania kodu. 2. Jako oprogramowania. 3. Testowanie oprogramowania.			
Content of the study programme (short version)			
1. Introduction to code versioning. 2. Software quality. 3. Software testing.			
Tre ci programowe			
			Liczba godzin
Semestr: 6			
Forma zaj : wykład			
1. Wprowadzenie do wersjonowania kodu: Rozproszony system kontroli wersji GIT (gitflow, branche, podstawowe i zaawansowane polecenia, rozwi zywanie konfliktów, współpraca zespołowa) 2. Jako oprogramowania: Code review, code style, dobre praktyki, SOLID, wzorce projektowe. 3. Testowanie oprogramowania: Wprowadzenie do testowania, testy jednostkowe, testy integracyjne, testy end to end, testy mutacyjne, statyczna analiza kodu, testy dynamiczne – testy statyczne, pokrycie kodu, TDD.			15
Forma zaj : wiczenia laboratoryjne			
1. Wprowadzenie do wersjonowania kodu: Rozproszony system kontroli wersji GIT (gitflow, branche, podstawowe i zaawansowane polecenia, rozwi zywanie konfliktów, współpraca zespołowa) 2. Jako oprogramowania: Code review, code style, dobre praktyki, SOLID, wzorce projektowe. 3. Testowanie oprogramowania:			30

Wprowadzenie do testowania, testy jednostkowe, testy integracyjne, testy end to end, testy mutacyjne, statyczna analiza kodu, testy dynamiczne – testy statyczne, pokrycie kodu, TDD.	30
Literatura	
Podstawowa	
Farcic V., Garcia A, TDD Programowanie w Javie sterowane testami., Helion 2016	
J. Bloch, Java. Efektywne programowanie, Helion 2018	
Robert. C. Martin, Czysty kod. Podręcznik dobrego programisty., Helion 2014	
Uzupełniająca	

Dane jako ciowe

Przyporządkowanie zajęć /grup zajęć do dyscypliny naukowej/artystycznej	informatyka techniczna i telekomunikacja	
Sposób określenia liczby punktów ECTS		
Forma nakładu pracy studenta (udział w zajęciach, aktywność, przygotowanie sprawozdania, itp.)	Obciążenie studenta [w godz.]	
Udział w zajęciach	45	
Konsultacje z prowadzącym	0	
Udział w egzaminie	0	
Bezpośredni kontakt z nauczycielem - inne	0	
Przygotowanie do laboratorium, ćwiczeń, zajęć	15	
Przygotowanie do kolokwium i egzaminu	15	
Indywidualna praca własna studenta z literatury, wykładami itp.	0	
Inne	0	
Sumaryczne obciążenie prac studenta	75	
Liczba punktów ECTS		
Liczba punktów ECTS	3	
Zajęcia wymagające bezpośredniego udziału nauczyciela akademickiego	L. godzin	ECTS
	45	1,8
Zajęcia o charakterze praktycznym	L. godzin	ECTS
	60	2,4

1 godz = 45 minut; 1 punkt ECTS = 25-30 godzin

W sekcji 'Liczba punktów ECTS' suma punktów ECTS zajęć wymagających bezpośredniego udziału nauczyciela akademickiego i o charakterze praktycznym może się różnić od łącznej liczby punktów ECTS dla zajęć/grup zajęć.