



KARTA PRZEDMIOTU

Kod przedmiotu	studia stacjonarne:	Z-ZIP2-U-231
	studia niestacjonarne:	Z-ZIPN2-U-231
Nazwa przedmiotu	Zaawansowane techniki programowania	
Nazwa przedmiotu w języku angielskim	Advanced Programming	
Obowiązuje od roku akademickiego	2019/2020	

USYTUOWANIE PRZEDMIOTU W SYSTEMIE STUDIÓW

Kierunek studiów	ZARZĄDZANIE I INŻYNIERIA PRODUKCJI
Poziom kształcenia	II stopień
Profil studiów	Ogólnoakademicki
Forma i tryb prowadzenia studiów	Studia stacjonarne
Zakres	Informatyka w zarządzaniu i modelowaniu
Jednostka prowadząca przedmiot	Katedra Technologii Informatycznych
Koordinator przedmiotu	dr inż. Sławomir Koczubiej
Zatwierdził	dr hab. inż. Artur Bartosik, prof. PŚk

OGÓLNA CHARAKTERYSTYKA PRZEDMIOTU

Przynależność do grupy/bloku przedmiotów	Przedmiot specjalnościowy	
Status przedmiotu	Obowiązkowy	
Język prowadzenia zajęć	Polski	
Usytuowanie w planie studiów - semestr	studia stacjonarne	Semestr II
	studia niestacjonarne	-
Wymagania wstępne	Brak	
Egzamin (TAK/NIE)	TAK	
Liczba punktów ECTS	3	

Forma prowadzenia zajęć		wykład	ćwiczenia	laboratorium	projekt	inne
Liczba godzin w semestrze	studia stacjonarne:	20		30		
	studia niestacjonarne:	-	-	-	-	-

EFEKTY UCZENIA SIĘ

Kategoria	Symbol efektu	Efekty uczenia się	Odniesienie do efektów kierunkowych
Wiedza	W01	Poznanie zaawansowanych modeli programowania obiektowego. Ma wiedzę dotyczącą różnych paradygmatów programowania.	ZIP2_W04
	W02	Posiada wiedzę dotyczącą budowania aplikacji wieloplatformowych.	ZIP2_W04
	W03	Student ma wiedzę dotyczącą modeli programowania i doboru ich do rozwiązania określonego zadania programistycznego.	ZIP2_W04
Umiejętności	U01	Student posiada umiejętność konstruowania algorytmów z wykorzystaniem zaawansowanych struktur danych służących rozwiązaniu różnych zadań programistycznych.	ZIP2_U11
	U02	Potrafi ocenić nowe technologie nowe narzędzia programistyczne, diagnostyczne i dokonać ich wyboru realizacji zadanych problemów.	ZIP2_U12
	U03	Posiada umiejętność wykorzystywania zasobów programistycznych i wykorzystać właściwe narzędzia programistyczne pozwalające na sprawne realizowanie projektów.	ZIP2_U11
Kompetencje społeczne	K01	Zdolność do kierowania zespołem ludzkim z uwzględnieniem czynników psychologicznych i społecznych członków zespołu.	ZIP2_K03

TREŚCI PROGRAMOWE

Forma zajęć	Treści programowe
wykład	<ol style="list-style-type: none">1. Metodologia budowania dużych projektów, kontrola wersji, zarządzanie projektem. Metodyka budowania testów. Biblioteki testujące. Optymalizacja kodu.2. Interfejsy i klasy ogólne. Metody anonimowe. Wielowątkowość, metody synchronizacji.3. Zaawansowane struktury danych i algorytmy. Wyrażenia regularne.4. Zaawansowane programowanie obiektowe. Wzorce projektowe.
laboratorium	<ol style="list-style-type: none">1. Zarządzanie projektem, kontrola wersji i zarządzanie kodem aplikacji. Pluginy, programowanie dynamiczne.2. Serializacja i klonowanie. Aplikacje wielowątkowe, metody synchronizacji wątków. Programowanie asynchroniczne.3. Zaawansowane biblioteki struktur danych. Aplikacje wieloplatformowe.4. Interfejsy i klasy ogólne, uniwersalne implementacje algorytmów. Kreatywne, strukturalne i czynnościowe wzorce projektowe.

METODY WERYFIKACJI EFEKTÓW UCZENIA SIĘ

Symbol efektu	Metody sprawdzania efektów kształcenia (zaznaczyć X)					
	Egzamin ustny	Egzamin pisemny	Kolokwium	Projekt	Sprawozdanie	Inne
W01		X	X			
W02		X	X			
W03		X	X			
U01		X	X		X	
U02		X	X		X	
U03		X	X		X	
K01					X	

FORMA I WARUNKI ZALICZENIA

Forma zajęć	Forma zaliczenia	Warunki zaliczenia
wykład	egzamin	Uzyskanie co najmniej 50% punktów z egzaminu, komentarze na wykładach.
laboratorium	zaliczenie z oceną	Uzyskanie co najmniej 50% z kolokwίων, aktywność na zajęciach laboratoryjnych.

NAKŁAD PRACY STUDENTA

Bilans punktów ECTS												
Lp.	Rodzaj aktywności	Obciążenie studenta										Jednostka
		studia stacjonarne					studia niestacjonarne					
		W	C	L	P	S	W	C	L	P	S	
1.	Udział w zajęciach zgodnie z planem studiów	20		30			-	-	-	-	-	h
2.	Inne (konsultacje, egzamin)	4		2			-	-	-	-	-	h
3.	Razem przy bezpośrednim udziale nauczyciela akademickiego	56					-					h
4.	Liczba punktów ECTS, którą student uzyskuje przy bezpośrednim udziale nauczyciela akademickiego	2,2					-					ECTS
5.	Liczba godzin samodzielnej pracy studenta	19					-					h
6.	Liczba punktów ECTS, którą student uzyskuje w ramach samodzielnej pracy	0,8					-					ECTS
7.	Nakład pracy związany z zajęciami o charakterze praktycznym	45					-					h
8.	Liczba punktów ECTS, którą student uzyskuje w ramach zajęć o charakterze praktycznym	1,8					-					ECTS
9.	Sumaryczne obciążenie pracą studenta	75					-					h
10.	Punkty ECTS za moduł <i>1 punkt ECTS=25 godzin obciążenia studenta</i>	3										ECTS

LITERATURA

1. Galowicz J., C++ STL. Receptury. Wydawnictwo Helion, Gliwice 2018.
2. Ezust A., Ezust P., C++ i Qt. Wprowadzenie do wzorców projektowych, Helion, Gliwice 2014.
3. Gamma E., Helm R., Johnson R., Vlissides J., Wzorce projektowe. Elementy oprogramowania obiektowego wielokrotnego użytku. Wydawnictwo Helion, Gliwice 2010.