



Karta przedmiotu

Nazwa i kod przedmiotu	Techniki programowania, PG_00047806						
Kierunek studiów	Informatyka						
Data rozpoczęcia studiów	październik 2020 r.	Rok akademicki realizacji przedmiotu			2020/2021		
Poziom kształcenia	I stopnia - inżynierskie	Grupa zajęć			Grupa zajęć fakultatywnych		
Forma studiów	niestacjonarne	Sposób realizacji			na uczelni		
Rok studiów	1	Język wykładowy			polski		
Semestr studiów	2	Liczba punktów ECTS			5.0		
Profil kształcenia	ogólnoakademicki	Forma zaliczenia			egzamin		
Jednostka prowadząca	Wydział Elektroniki, Telekomunikacji i Informatyki -> Katedra Systemów Mikroelektronicznych						
Imię i nazwisko wykładowcy (wykładowców)	Odpowiedzialny za przedmiot	dr inż. Maciej Kokot					
	Prowadzący zajęcia z przedmiotu	dr inż. Maciej Kokot dr hab. inż. Zbigniew Łubniewski					
Formy zajęć i metody nauczania	Forma zajęć	Wykład	Ćwiczenia	Laboratorium	Projekt	Seminarium	RAZEM
	Liczba godzin zajęć	15.0	0.0	0.0	15.0	0.0	30
	W tym liczba godzin zajęć na odległość: 0.0						
	Adresy na platformie eNauczanie:						
Aktywność studenta i liczba godzin pracy	Aktywność studenta	Udział w zajęciach dydaktycznych, objętych planem studiów	Udział w konsultacjach		Praca własna studenta		RAZEM
	Liczba godzin pracy studenta	30	5.0		90.0		125
Cel przedmiotu	Zapoznanie z technikami programowania. Nabycie umiejętności czytelnego i niezawodnego programowania. Znajomość narzędzi wspomagających programowanie - kompilator, linker, debugger. Poznanie zasad programowania strukturalnego i obiektowego, polimorfizm, enkapsulacja, dziedziczenie.						

Efekty uczenia się przedmiotu	Efekt kierunkowy	Efekt z przedmiotu	Sposób weryfikacji i oceny efektu
	[K6_U41] potrafi wytwarzać, testować lub oceniać oprogramowanie, wykorzystując nowoczesne platformy, narzędzia, języki i paradygmaty programowania różnych poziomów, a także posługiwać się pakietami oprogramowania wspierającymi naukowo-badawcze i biznesowe procesy decyzyjne oraz pracę zespołową	Student potrafi zaplanować przebieg prac potrzebnych do wytworzenia oprogramowania. Potrafi odpowiednio dobrać środowisko programistyczne. Potrafi wytworzyć oprogramowanie i je przetestować.	[SU4] Ocena umiejętności korzystania z metod i narzędzi [SU1] Ocena realizacji zadania
	[K6_U04] potrafi wykorzystywać posiadaną wiedzę z zakresu metod i technik programowania oraz dobrać i zastosować właściwe metody i narzędzia programistyczne w tworzeniu oprogramowania komputerów albo programowania urządzeń lub sterowników wykorzystujących mikroprocesory albo elementy lub układy programowalne, charakterystycznych dla danego kierunku studiów	Student zna różnice pomiędzy programowaniem strukturalnym a obiektowym, rozumie pojęcia polimorfizmu i dziedziczenia. Potrafi programować strukturalnie i obiektowo.	[SU4] Ocena umiejętności korzystania z metod i narzędzi [SU1] Ocena realizacji zadania
	[K6_U08] potrafi przy identyfikacji i formułowaniu specyfikacji zadań inżynierskich związanych z kierunkiem studiów oraz ich rozwiązywaniu: – wykorzystać metody analityczne, symulacyjne i eksperymentalne, – dostrzegać ich aspekty systemowe i pozatechniczne, – dokonać wstępnej oceny ekonomicznej proponowanych rozwiązań i podejmowanych działań inżynierskich	Student potrafi zaplanować przebieg prac potrzebnych do wytworzenia oprogramowania. Potrafi odpowiednio dobrać środowisko programistyczne. Potrafi wytworzyć oprogramowanie i je przetestować.	[SU2] Ocena umiejętności analizy informacji [SU1] Ocena realizacji zadania
	[K6_W42] zna i rozumie w zaawansowanym stopniu architektury, zasady projektowania oraz metody wsparcia sprzętowego i programowego dla lokalnych i rozproszonych systemów informatycznych, w tym systemów obliczeniowych, baz danych, sieci komputerowych i aplikacji informacyjnych, a także zasady współpracy człowieka z komputerem i wspomaganej komputerowo pracy zespołowej	Student zna różnorodne modele programowania. Zna ewolucję modeli i związanych z nimi języków programowania.	[SW1] Ocena wiedzy faktograficznej
	[K6_U03] potrafi zaprojektować, zgodnie z zadaną specyfikacją, oraz wykonać typowe dla kierunku studiów proste urządzenie, obiekt, system lub zrealizować proces, używając odpowiednio dobranych metod, technik, narzędzi i materiałów, korzystając ze standardów i norm inżynierskich, stosując właściwe dla kierunków studiów technologie i wykorzystując doświadczenie zdobyte w środowisku zajmującym się zawodowo działalnością inżynierską	Student zna różnice pomiędzy programowaniem strukturalnym a obiektowym, rozumie pojęcia polimorfizmu i dziedziczenia. Potrafi programować strukturalnie i obiektowo. Student potrafi zaplanować przebieg prac potrzebnych do wytworzenia oprogramowania. Potrafi odpowiednio dobrać środowisko programistyczne. Potrafi wytworzyć oprogramowanie i je przetestować.	[SU4] Ocena umiejętności korzystania z metod i narzędzi [SU2] Ocena umiejętności analizy informacji [SU1] Ocena realizacji zadania
Treści przedmiotu	Wprowadzenie. Własności programów, sposób ich osiągania. Reguły stylu programowania. Czytelność kodu. Programowanie strukturalne. Wyjątki niestrukturalne. Metody projektowania programów. Metoda analityczna i syntetyczna, ich zastosowania. Pseudokod Programowanie modułowe Graficzne przedstawianie programów – schemat blokowy i strukturogram. Przyczyny niestrukturalności i ich usuwanie. Narzędzia wspomagające programowanie: kompilator, linker, debugger. Pliki nagłówkowe i zasady ich stosowania. Narzędzie make i jego zastosowanie. Pliki Makefile – zmienne, cele, reguły domyślne. Zaawansowane możliwości i zastosowania make" a Programowanie hierarchiczne – wstęp do programowania obiektowego. Dziedziczenie. Enkapsulacja. Sekcje dostępu. Programowanie obiektowe. Polimorfizm. Konstruktory i destrukторы. Programowanie generyczne. Porównanie programowania strukturalnego z obiektowym. Najważniejsze różnice i zasady.		
Wymagania wstępne i dodatkowe			

Sposoby i kryteria oceniania osiągniętych efektów uczenia się	Sposób oceniania (składowe)	Próg zaliczeniowy	Składowa oceny końcowej
	Projekt	50.0%	50.0%
	Egzamin pisemny	66.0%	50.0%
Zalecana lista lektur	Podstawowa lista lektur	1. D. van Tassel: Praktyka programowania. WNT, Warszawa. 2. Kernighan, Ritchie: Język C. WNT, Warszawa. 3. Stallman, McGrath, Smith: GNU Make. www.gnu.org/software/make/manual/	
	Uzupełniająca lista lektur	1. C++ bez cholesterolu. www.intercon.pl/~sektor/cbx/cbx.html	
	Adresy eZasobów		
Przykładowe zagadnienia/ przykładowe pytania/ realizowane zadania			
Praktyki zawodowe w ramach przedmiotu	Nie dotyczy		