



Karta przedmiotu

Nazwa i kod przedmiotu	Object programming, PG_00045295						
Kierunek studiów	Inżynieria danych, Inżynieria danych						
Data rozpoczęcia studiów	październik 2026 r.	Rok akademicki realizacji przedmiotu			2026/2027		
Poziom kształcenia	I stopnia - inżynierskie	Grupa zajęć			Grupa zajęć obowiązkowych z zakresu kierunku studiów Grupa zajęć powiązanych z prowadzonymi badaniami naukowymi w dziedzinie nauki związanej z kierunkiem - profil ogólnoakademicki		
Forma studiów	stacjonarne	Sposób realizacji			na uczelni		
Rok studiów	1	Język wykładowy			polski		
Semestr studiów	2	Liczba punktów ECTS			3.0		
Profil kształcenia	ogólnoakademicki	Forma zaliczenia			egzamin		
Jednostka prowadząca	Wydział Politechniki Gdańskiej -> Wydział Elektroniki, Telekomunikacji i Informatyki -> Katedra Systemów Geoinformatycznych						
Imię i nazwisko wykładowcy (wykładowców)	Odpowiedzialny za przedmiot	dr hab. inż. Marek Moszyński					
	Prowadzący zajęcia z przedmiotu	dr hab. inż. Marek Moszyński mgr inż. Tomasz Idzi mgr inż. Tomasz Bieliński					
Formy zajęć	Forma zajęć	Wykład	Ćwiczenia	Laboratorium	Projekt	Seminarium	RAZEM
	Liczba godzin zajęć	15.0	0.0	11.0	30.0	0.0	56
	W tym liczba godzin zajęć na odległość: 0.0						
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	56	6.0		13.0		75
Cel przedmiotu	Nabycie wiedzy z zakresu teorii programowania obiektowego oraz umiejętności programowania z wykorzystaniem tego typu podejścia.						
Efekty uczenia się przedmiotu	Efekt kierunkowy		Efekt z przedmiotu		Sposób weryfikacji i oceny efektu		
	[K6_W06] klasyfikuje pozyskiwane informacje, oceniając ich przydatność do rozwiązania sformułowanych problemów		Student zdobywa umiejętności praktyczne wykonując przykładowe zadania w kilku językach programowania obiektowego.		[SW3] Ocena wiedzy zawartej w opracowaniu tekstowym i projektowym		
	[K6_U02] przygotowuje i przedstawia w sposób przekonujący profesjonalne prezentacje wyników swoich działań, z ich zaawansowaną interpretacją		Student potrafi zaprezentować rozwiązania stosowane w zrealizowanych przez siebie zadaniach.		[SU5] Ocena umiejętności zaprezentowania wyników realizacji zadania [SU4] Ocena umiejętności korzystania z metod i narzędzi		
	[K6_U06] zdobywa nową wiedzę, planując własny rozwój sprzyjający osiągnięciu wyznaczonych celów		Student nabywa umiejętności z zakresu podstaw programowania obiektowego z wykorzystaniem następujących języków programowania: C++, Java, C#, Python i Javascript.		[SU2] Ocena umiejętności analizy informacji [SU1] Ocena realizacji zadania		

<p>Treści przedmiotu</p>	<p>Treści przedmiotu - wykład</p> <ol style="list-style-type: none"> 1. Paradygmaty programowania ze szczególnym uwzględnieniem paradygmatu obiektowego 2. Implementacja enkapsulacji, dziedziczenia, abstrakcji i polimorfizmu w języku C++ 3. Specyfika implementacji obiektowości w języku C++ 4. Język Java i jego porównanie z językiem C++ 5. Język C# i jako następcą języka C i porównanie z językiem Java 6. Język Python jako przedstawiciel skryptowych języków programowania obiektowego <hr/> <p>Treści przedmiotu - laboratoria</p> <ol style="list-style-type: none"> 1. Wprowadzenie do zajęć laboratoryjnych 2. Wstęp do programowania obiektowego w C++ Omówienie zagadnień dotyczących programowania obiektowego i obejmujących: definiowanie klas i obiektów, hermetyzację, dziedziczenie, polimorfizm oraz abstrakcję, a także budowanie hierarchii klas i wykorzystanie metod wirtualnych. Zagadnienia związane z zarządzaniem pamięcią w klasach, w tym dynamiczną alokacją, destruktorami oraz odpowiedzialnością za zasoby. 3. Zarządzanie cyklem życia obiektów oraz mechanizmy operacji na obiektach w C++. Wprowadzenie do mechanizmów kopiowania obiektów, konstruktor kopiujący, operatory przypisania, przekazywanie i zwracanie obiektów przez wartość i referencję, optymalizacja operacji na obiektach. 4. Zagadnienia dotyczące programowania obiektowego w języku Java, obejmujące definiowanie klas i obiektów, hermetyzację, dziedziczenie, polimorfizm oraz abstrakcję, a także wykorzystanie kolekcji i typów generycznych. 5. Interfejsy funkcjonalne oraz programowanie zdarzeniowe w języku Java, z wykorzystaniem wyrażeń lambda. Projektowanie i rozwój aplikacji obiektowej, w tym rozbudowa hierarchii klas i implementacja mechanizmów obsługi zdarzeń. <hr/> <p>Treści przedmiotu - projekt</p> <ol style="list-style-type: none"> 1. Samodzielna realizacja zadania w języku C++ 2. Samodzielna realizacja zadania w języku Java 3. Samodzielna realizacja zadania w języku Python <ol style="list-style-type: none"> 1. Samodzielna realizacja zadania w języku C++ 2. Samodzielna realizacja zadania w języku Java 3. Samodzielna realizacja zadania w języku Python <ol style="list-style-type: none"> 1. Samodzielna realizacja zadania w języku C++ 2. Samodzielna realizacja zadania w języku Java 3. Samodzielna realizacja zadania w języku Python
<p>Wymagania wstępne i dodatkowe</p>	<p>Znajomość dowolnego języka programowania nieobiektywnego np języka C</p>

Sposoby i kryteria oceniania osiągniętych efektów uczenia się	Sposób oceniania (składowe)	Próg zaliczeniowy	Składowa oceny końcowej
	wykład	60.0%	34.0%
	projekt	60.0%	33.0%
	laboratorium	60.0%	33.0%
Zalecana lista lektur	Podstawowa lista lektur	Bjarne Stroustrup - The C++ programming language Bruce Eckel - Thinking in Java Andy Harris - Microsoft C# for absolute beginner Mark Lutz - Programming Python	
	Uzupełniająca lista lektur	John Hunt - Smalltalk and Object Orientation	
	Adresy eZasobów		
Przykładowe zagadnienia/ przykładowe pytania/ realizowane zadania	Przykładowe pytanie: w jakim kierunku rozwija się język C++? Przykładowe zadanie: Implementacja prostego programu obiektowego z zastosowaniem paradygmatów programowania obiektowego w różnych językach programowania.		
Zajęcia praktyczne w ramach przedmiotu	Nie dotyczy		

Dokument wygenerowany elektronicznie. Nie wymaga pieczęci ani podpisu.