

KARTA PRZEDMIOTU (SYLABUS)

Opis przedmiotu

Kod przedmiotu		Nazwa przedmiotu	Języki programowania	
RiSI/O/II/NST/A5			Programming Languages	
Język wykładowy		Polski		
Rok akademicki		2026/2027		
Kierunek		Robotyka i Sztuczna Inteligencja		
w zakresie		-		
Poziom studiów		studia drugiego stopnia		
Profil studiów		ogólnoakademicki		
Forma studiów		studia niestacjonarne		
Semestr / semestry		I		
Przynależność do grupy zajęć		Grupa zajęć podstawowych		
Status przedmiotu		Obowiązkowy		
Formy realizacji zajęć dydaktycznych, wymiar, punkty ECTS		Forma zajęć	Liczba godzin zajęć dydaktycznych	Liczba punktów ECTS
		Wykład	15	4
		Laboratorium	30	
Powiązanie przedmiotu	z profilem studiów	związany z prowadzoną działalnością naukową w dyscyplinie inżynieria mechaniczna, do której przyporządkowany jest kierunek studiów		0 ECTS
	z uprawnieniami	służy zdobywaniu przez studenta kompetencji inżynierskich		0 ECTS
	z dyscypliną	inżynieria mechaniczna		4 ECTS
Forma nauczania		Tradycyjna, zajęcia zorganizowane w Uczelni / zajęcia realizowane z wykorzystaniem metod i technik kształcenia na odległość / inne		
Wymagania wstępne		brak dodatkowych wymagań		
Jednostka prowadząca		Wydział Mechaniczny		
Koordynator		Marcin Migus, dr inż.		
Adres strony internetowej pjo		https://wm.uniwersytetradom.pl/		
Adres e-mail, telefon koordynatora		m.migus@urad.edu.pl		

EFEKTY UCZENIA SIĘ, TREŚCI PROGRAMOWE, REALIZACJA ZAJĘĆ
DYDAKTYCZNYCH, WERYFIKACJA EFEKTÓW UCZENIA SIĘ

Cel kształcenia:	Celem przedmiotu jest zapoznanie z podstawowymi zagadnieniami dotyczącymi języków programowania stosowanych w zadaniach inżynierskich, ze szczególnym uwzględnieniem programowania algorytmicznego, struktur danych, jakości kodu oraz korzystania z dokumentacji technicznej. Kolejnym celem jest także rozwinięcie umiejętności implementacji i testowania prostych programów w języku Python oraz analizy poprawności działania opracowanych rozwiązań.
Treści programowe:	<p>Wykład:</p> <ol style="list-style-type: none"> 1. wprowadzenie do języków programowania i ich roli w robotyce oraz sztucznej inteligencji, 2. podstawowe elementy składni i semantyki języka programowania, 3. typy danych, zmienne, operatory i instrukcje sterujące, 4. funkcje, moduły i organizacja kodu, 5. podstawowe struktury danych i operacje na danych, 6. wprowadzenie do algorytmów i analizy poprawności działania programu, 7. podstawy testowania, debugowania i optymalizacji kodu, 8. znaczenie dokumentacji technicznej, licencjonowania i zasad rzetelności w tworzeniu oprogramowania. <p>Laboratorium:</p> <ol style="list-style-type: none"> 1. przygotowanie środowiska programistycznego i uruchamianie prostych programów, 2. implementacja podstawowych operacji na danych w języku Python, 3. stosowanie instrukcji warunkowych, pętli i funkcji, 4. implementacja prostych struktur danych i algorytmów, 5. wczytywanie danych, przetwarzanie wyników i prezentacja rezultatów działania programu, 6. testowanie i poprawianie błędów w kodzie, 7. analiza jakości i czytelności kodu, 8. realizacja prostego zadania programistycznego z dokumentacją i prezentacją wyników.
Metody dydaktyczne (kształcenia):	<p>Wykład informacyjny i problemowy z prezentacją przykładów programistycznych oraz analizą poprawności i jakości rozwiązań.</p> <p>Laboratorium oparte o metodę praktycznych zadań programistycznych, samodzielną implementację wybranych algorytmów, analizę kodu, testowanie oraz prezentację wyników.</p>
Rygor zaliczenia, kryteria oceny osiągniętych efektów uczenia się, sposób obliczania oceny końcowej:	<p>Zaliczenie obejmuje pracę pisemną (test), projekt programistyczny oraz prezentację wyników laboratorium. Każdy element jest oceniany według ustalonych kryteriów, a suma punktowa decyduje o zaliczeniu.</p> <p>Test pisemny sprawdza wiedzę teoretyczną i umiejętność analizy algorytmicznej; projekt programistyczny ocenia praktyczne zastosowanie języków oraz optymalizację kodu; prezentacja laboratorium ocenia komunikację techniczną i zdolność do krytycznej oceny wyników.</p> <p>Ocena końcowa jest obliczana jako średnia ważona: test 30 %, projekt 40 %, prezentacja 30 %. Zaliczenie uzyskuje się przy średniej $\geq 2,0$ (w skali 1-5).</p>

Efekty uczenia się dla przedmiotu w odniesieniu do efektów kierunkowych i formy zajęć				Metody weryfikacji efektów uczenia się	
Numer efektu uczenia się	Opis efektów uczenia się dla przedmiotu (PEU) Student, który zaliczył	Kierunkowy efekt uczenia się (KEU)	Forma zajęć	Forma weryfikacji (zaliczeń)	Metody sprawdzania i oceny

	przedmiot (W) zna i rozumie / (U) potrafi / (K) jest gotów do:				
W1	Student zna i rozumie zaawansowane metody obliczeniowe oraz zasady zarządzania oprogramowaniem i zasobami cyfrowymi, w tym aspekty ochrony własności intelektualnej i prawa autorskiego.	K_WG01, K_WK15	wykład	test pisemny	analiza kodu, test teoretyczny
U1	Student potrafi implementować modele matematyczne i algorytmy w językach Python i/lub C i/lub Matlab, korzystając przy tym z dokumentacji technicznej i baz danych.	K_UW01, K_UK11	laboratorium	projekt programistyczny	kodowanie, recenzja kodu, test funkcjonalności
K1	Student jest gotów do odpowiedzialnego pełnienia ról zawodowych związanych z programowaniem oraz do przestrzegania zasad etyki i rzetelności w tworzeniu kodu.	K_KR05	laboratorium	prezentacja wyników laboratoryjnych	ocena prezentacji, analiza przypadków

Literatura i pomoce naukowe

Literatura podstawowa:

1. Matthes E., *Python Crash Course*, 3rd ed., No Starch Press, 2023.
2. Python Software Foundation, *The Python Tutorial*, aktualna dokumentacja Python 3.14.
3. MathWorks, *MATLAB Documentation* oraz *Language Fundamentals*,

Literatura uzupełniająca:

1. Python Software Foundation, *Python Tutorial – dokumentacja polska*, aktualna wersja.
2. MathWorks Educator Content Development Team, *Fundamentals of Programming*, MathWorks / MATLAB Central / GitHub, wersja 2025.
3. MathWorks Educator Content Development Team, *Programming: Structuring Code*, MathWorks / MATLAB Central, aktualizacja 2024.
4. Matthes E., *Python. Instrukcje dla programisty. Wydanie III*, 2023

Nakład pracy studenta potrzebny do osiągnięcia zakładanych efektów uczenia się – bilans punktów ECTS

Udział w zajęciach/aktywność	Obciążenie studenta [h]	
	Zajęcia bez nauczyciela-praca własna studenta (ZBN)	Zajęcia dydaktyczne
Udział w wykładach/ćwiczeniach/laboratoriach	X	45 h
Przygotowanie do wykładów/ćwiczeń/lab	55 h	X
Sumaryczne obciążenie pracą studenta	55 h / 2,2 ECTS	45 h / 1,8 ECTS
Punkty ECTS za przedmiot	4 ECTS	

Informacje dodatkowe, uwagi

W przypadku studentów ze szczególnymi potrzebami, w tym: z niepełnosprawnością, przewlekłe chorych, określone powyżej (w karcie) metody i formy weryfikacji efektów uczenia się dostosowuje się odpowiednio do indywidualnych potrzeb tych studentów.

Szczegółowe zasady i formy wsparcia studentów ze szczególnymi potrzebami: w tym z niepełnosprawnością, przewlekłe chorych podczas zajęć, zaliczeń i egzaminów określono w: Regulaminie Studiów, Zasadach Studiowania, Procedurze dotyczącej zapewnienia dostępności procesu kształcenia studentom ze szczególnymi potrzebami, w tym: z niepełnosprawnością, przewlekłe chorych.

