



Karta przedmiotu  
**Języki programowania**

**1. Informacje podstawowe**

<b>Kierunek studiów</b> mechatronika	<b>Cykl kształcenia (nabór)</b> 2024/25	
<b>Specjalność</b> -	<b>Kod przedmiotu</b> 03MCHS.PI6C.0085.24	
<b>Jednostka zarządzająca kierunkiem studiów</b> Wydział Inżynierii Mechanicznej	<b>Języki wykładowe</b> polski	
<b>Poziom studiów</b> pierwszego stopnia (inż.)	<b>Obligatoryjność</b> Obowiązkowy	
<b>Profil studiów</b> Profil ogólnoakademicki	<b>Blok zajęciowy</b> Przedmioty kierunkowe	
<b>Forma studiów</b> studia stacjonarne		
<b>Wymagania wstępne</b> -		
<b>Przedmioty wprowadzające</b>	technologia informacyjna	
<b>Koordinator</b>	Krzysztof Nowicki	
<b>Okres</b> Semestr 2	<b>Forma i godziny zajęć</b> • Wykład: 30, Zaliczenie na ocenę • Ćwiczenia laboratoryjne: 30, Zaliczenie na ocenę	<b>Liczba punktów ECTS</b> 4
<b>Okres</b> Semestr 3	<b>Forma i godziny zajęć</b> • Ćwiczenia laboratoryjne: 30, Zaliczenie na ocenę	<b>Liczba punktów ECTS</b> 2

**2. Efekty uczenia się dla przedmiotu**

Kod	Opis efektów uczenia się	Odniesienie do kierunkowych efektów uczenia się	Odniesienie do charakterystyk PRK
<b>Wiedza:</b>			
W1	Posiada wiedzę z zakresu teorii języków sterowania oraz stosowanych w nich metod gromadzenia, przechowywania, przetwarzania danych	MCH_O1_K_W10	P6S_WG P6S_WK P6S_WG_inż P6S_WK_inż
<b>Umiejętności:</b>			
U1	Potrafi z wykorzystaniem języka programowania Python zaprojektować oraz wykonać aplikację komputerową	MCH_O1_K_U04	P6S_UW P6S_UK P6S_UW_inż
U2	Potrafi omawiać kwestie związane z programowaniem w języku Python z użyciem specjalistycznej terminologii.	MCH_O1_K_U12	P6S_UW P6S_UK P6S_UO P6S_UW_inż
<b>Kompetencje społeczne:</b>			
K1	Potrafi wykonać projekt z wykorzystaniem języka programowania samodzielnie lub w grupie.	MCH_O1_K_K01	P6S_KO P6S_KR

### 3. Treści programowe

Lp.	Treści programowe	Formy zajęć	Efekty uczenia się dla przedmiotu
1.	Paradygmaty przetwarzania informacji. Paradygmaty programowania. Semantyka zmiennych. Typy danych. Typy abstrakcyjne. Programowanie strukturalne. Programowanie obiektowe. Programowanie funkcyjne. Testy jednostkowe.	Wykład	W1, U2
2.	Wprowadzenie do narzędzia JupyterLab. Praca z językiem Python w trybie interaktywnym. Wartości liczbowe i operacje na nich. Ciągi znaków i operacje na nich. Podstawy struktury języka Python. Obsługa wyjątków. Obsługa plików tekstowych i binarnych. Moduły. Biblioteka standardowa. Biblioteki zewnętrzne. Programowanie strukturalne - procedury i funkcje. Programowanie obiektowe - atrybuty, metody, klasy, obiekty. Testy jednostkowe.	Ćwiczenia laboratoryjne	U1, K1

### 4. Metody prowadzenia zajęć, weryfikacji efektów uczenia się i warunki zaliczenia

#### Semestr 2

Forma zajęć			
Wykład	<b>Metody prowadzenia zajęć:</b>		
	Wykład, Case study		
	<b>Metody (sposoby) weryfikacji:</b>		<b>Udział:</b>
	Kolokwium		100%
	<b>Warunki zaliczenia przedmiotu:</b>		
Uzyskanie 50% punktów z kolokwium pisemnego			

Ćwiczenia laboratoryjne	<b>Metody prowadzenia zajęć:</b>	
	Ćwiczenia laboratoryjne	
	<b>Metody (sposoby) weryfikacji:</b>	<b>Udział:</b>
	Sprawdzian zdolności metodycznych i umiejętności technicznych	100%
	<b>Warunki zaliczenia przedmiotu:</b>	
Rozwiązanie do końca semestru zestawu ćwiczeń laboratoryjnych, w tym zadań poprawnie rozwiązanych 50%.		

### Semestr 3

Forma zajęć		
Ćwiczenia laboratoryjne	<b>Metody prowadzenia zajęć:</b>	
	Ćwiczenia laboratoryjne	
	<b>Metody (sposoby) weryfikacji:</b>	<b>Udział:</b>
	Projekt	100%
	<b>Warunki zaliczenia przedmiotu:</b>	
Przedstawienie działającej aplikacji opracowanej w języku programowania Python w oparciu o samodzielnie zdefiniowany lub narzucony temat.		

Efekt uczenia się dla przedmiotu	Metody (sposoby) weryfikacji		
	Kolokwium	Sprawdzian zdolności metodycznych i umiejętności technicznych	Projekt
W1	x		
U1		x	x
U2		x	
K1			x

## 5. Literatura

### Literatura podstawowa

1. Mark Lutz M., Learning Python, O'Reilly Media, 2013
2. Pilgrim M., Dive into Python 3, Apress, 2009

### Literatura uzupełniająca

1. Nagar S., Introduction to Python for Engineers and Scientists: Open Source Solutions for Numerical Computation, Apress, 2017

## 6. Nakład pracy studenta - bilans godzin i punktów ECTS

Aktywność studenta		Obciążenie studenta Liczba godzin
Zajęcia prowadzone z bezpośrednim udziałem nauczyciela akademickiego lub innych osób prowadzących zajęcia	Wykład	30
	Ćwiczenia laboratoryjne	60
Praca własna studenta	Konsultacje	10
	Przygotowanie do zajęć	10
	Studiowanie literatury	10
	Przygotowanie do zaliczenia	10
	Przygotowanie projektu	50
<b>Łączny nakład pracy studenta</b>		180
<b>Liczba punktów ECTS</b>		6

\* Godzina (dydaktyczna) oznacza 45 minut