



KARTA PRZEDMIOTU

Kod przedmiotu	studia stacjonarne:	Z-IB-306
	studia niestacjonarne:	Z-IBN-306
Nazwa przedmiotu	Programowanie komputerów	
Nazwa przedmiotu w języku angielskim	Computer programming	
Obowiązuje od roku akademickiego	2026/2027	

USYTUOWANIE PRZEDMIOTU W SYSTEMIE STUDIÓW

Kierunek studiów	INŻYNIERIA BIOMEDYCZNA	
Poziom kształcenia	I stopień	
Profil studiów	Praktyczny	
Forma i tryb prowadzenia studiów	Studia stacjonarne i niestacjonarne	
Zakres	Wszystkie zakresy	
Jednostka prowadząca przedmiot	Uczelnia	Politechnika Świętokrzyska
	Jednostka	Katedra Technologii Informatycznych
Koordinator przedmiotu	dr hab. Marzena Nowakowska, prof. PŚk	
Zatwierdził	dr hab. inż. Dariusz Bojczuk, prof. PŚk	

OGÓLNA CHARAKTERYSTYKA PRZEDMIOTU

Przynależność do grupy/bloku przedmiotów	Przedmiot kierunkowy	
Status przedmiotu	Obowiązkowy	
Język prowadzenia zajęć	Polski	
Usytuowanie w planie studiów - semestr	studia stacjonarne	Semestr III
	studia niestacjonarne	Semestr III
Wymagania wstępne	Technologie informacyjne Podstawy informatyki	
Egzamin (TAK/NIE)	TAK	
Liczba punktów ECTS	3	

Forma prowadzenia zajęć		wykład	ćwiczenia	laboratorium	projekt	inne
Liczba godzin w semestrze	studia stacjonarne:	15		30		
	studia niestacjonarne:	9		18		

EFEKTY UCZENIA SIĘ

Kategoria	Symbol efektu	Efekty kształcenia	Odniesienie do efektów kierunkowych
Wiedza	W01	Zna podstawowe struktury danych i ich właściwości w kontekście tworzenia i budowy programów komputerowych.	IB1_W04
	W02	Ma wiedzę nt. programowania strukturalnego i zorientowanego obiektowego. Rozumie zasady korzystania z obiektów własnych oraz wbudowanych (dostarczanych w bibliotekach języka programowania).	IB1_W04 IB1_W13
Umiejętności	U01	Potrafi formułować algorytmy w języku programowania i dobrać odpowiednie struktury danych.	IB1_U01 IB1_U08 IB1_U15
	U02	Posiada umiejętności korzystania z gotowych narzędzi programistycznych we własnych programach.	IB1_U01 IB1_U12
Kompetencje społeczne	K01	Rozumie znaczenie doskonalenia nabytej wiedzy i umiejętności z zakresu struktur danych i programach operujących na tych strukturach.	IB1_K01
	K02	Rozumie potrzebę stałego uzupełniania wiedzy z obszaru nowoczesnych narzędzi i idei informatyki.	IB1_K01 IB1_K04

TREŚCI PROGRAMOWE

Forma zajęć	Treści programowe
wykład	<p>Charakterystyka języka programowania wysokiego poziomu wybranego do nauki programowania oraz środowisko IDE dla tego języka. Translacja programu komputerowego.</p> <p>Paradygmat programowania strukturalnego; struktury kontrolne programu, podprogramy, bloki. Koncepcja algorytmu i jego zapisu w języku programowania komputerów – przetwarzanie prostych i złożonych struktur danych.</p> <p>Paradygmat programowania zorientowanego obiektowo (OOP) – podstawy; definiowanie klasy i jej składowych, tworzenie obiektów, wykorzystanie atrybutów i metod do definiowania właściwości obiektów i czynności z nimi związanych.</p> <p>Współpraca programu z plikiem dyskowym; operacje odczytu i zapisu z wykorzystaniem wbudowanych narzędzi programistycznych.</p> <p>Specjalistyczne biblioteki programistyczne dla wybranego języka programowania; zasady dostępu i korzystania z ich zasobów.</p> <p>Programowanie wybranych zagadnień analityki i wizualizacji danych z zastosowaniem wbudowanych funkcji i obiektów bibliotek specjalistycznych.</p>
laboratorium	<p>Organizacja środowiska programistycznego IDE dla wybranego języka programowania.</p> <p>Programowanie strukturalne; implementacja algorytmów przetwarzania danych liczbowych i tekstowych, uruchomienie i wykonanie programu, testowanie programu. Rodzaje błędów w programie komputerowym i ich eliminacja.</p> <p>Programowanie zorientowane obiektowo; przygotowanie koncepcji klas oraz opracowanie programu do przetwarzania danych z wykorzystaniem obiektów zdefiniowanych wg tych klas. Sprawdzenie poprawności działania programu.</p> <p>Programowanie obsługi plików dyskowych w formatach: txt, csv, tsv oraz xlsx.</p> <p>Dostęp do wybranych tematycznych bibliotek programistycznych; wykorzystanie wbudowanych obiektów tych bibliotek do eksploracji danych i wizualizacji wyników obliczeń.</p> <p>Zasady poszukiwania i korzystania ze źródeł internetowych w zakresie rozwiązywania problemów programistycznych.</p>

METODY WERYFIKACJI EFEKTÓW UCZENIA SIĘ

Symbol efektu	Metody sprawdzania efektów kształcenia					
	Egzamin ustny	Egzamin pisemny	Kolokwium	Projekt	Sprawozdanie	Inne (obserwacja, dyskusja)
W01		X				
W02		X				
U01		X	X			
U02		X	X			
K01						X
K02						X

FORMA I WARUNKI ZALICZENIA

Forma zajęć	Forma zaliczenia	Warunki zaliczenia
wykład	egzamin	Uzyskanie co najmniej 50% możliwych punktów z testu pisemnego. Obecność na wykładach jest premiowana dodatkowymi punktami.
laboratorium	zaliczenie z oceną	Uzyskanie co najmniej 50% punktów z kolokwium przy komputerze w trakcie laboratoriów. Aktywny udział w zajęciach jest premiowany dodatkowymi punktami.

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	15		30			9		18					h
2.	Inne (konsultacje, egzamin)	1		2			1		2					h
3.	Razem przy bezpośrednim udziale nauczyciela akademickiego	48					30					h		
4.	Liczba punktów ECTS, którą student uzyskuje przy bezpośrednim udziale nauczyciela akademickiego	1,9					1,2					ECTS		
5.	Liczba godzin samodzielnej pracy studenta	27					45					h		
6.	Liczba punktów ECTS, którą student uzyskuje w ramach samodzielnej pracy	1,1					1,8					ECTS		
7.	Nakład pracy związany z zajęciami o charakterze praktycznym	50					50					h		
8.	Liczba punktów ECTS, którą student uzyskuje w ramach zajęć o charakterze praktycznym	2,0					2,0					ECTS		
9.	Sumaryczne obciążenie pracą studenta	75					75					h		
10.	Punkty ECTS za moduł <i>1 punkt ECTS=25 godzin obciążenia studenta</i>	3										ECTS		

LITERATURA

1. Althoff C. (2017), *Programista samouk. Profesjonalny przewodnik do samodzielnej nauki kodowania*, Helion, Gliwice.
2. Bhargava A. (2017), *Algorytmy. Ilustrowany przewodnik*, Helion, Gliwice.
3. Cormen T. H., Leiserson C. E., Rivest R. L., Clifford S. (2021), *Wprowadzenie do algorytmów*, Wydawnictwo Naukowe PWN S.A, Warszawa.
4. Diks K., Banachowski L., Rytter W. (2017). *Algorytmy i struktury danych*, wydanie nowe poprawione, PWN, Warszawa
5. Lubanovic B. (2020), *Python. Nowoczesne programowanie w prostych krokach*, Helion, Gliwice.
6. Matthes E. (2020), *Python. Instrukcje dla programisty*, Helion, Gliwice.
7. <http://pl.python.org> 4.
8. <https://docs.python.org/3/>
9. Dostępne w Internecie: materiały dydaktyczne, szkolenia on-line, grupy dyskusyjne, po wpisaniu w przeglądarkę hasła: *programowanie*, *programowanie w Pythonie* lub określonej tematyki interesującej studenta.