



KARTA PRZEDMIOTU

Kod przedmiotu	studia stacjonarne:	
	studia niestacjonarne:	
Nazwa przedmiotu	Zaawansowane programowanie w języku Python	
Nazwa przedmiotu w języku angielskim	Advanced Python Programming	
Obowiązuje od roku akademickiego	2022/23	

USYTUOWANIE PRZEDMIOTU W SYSTEMIE STUDIÓW

Kierunek studiów	Informatyka
Poziom kształcenia	I stopień
Profil studiów	ogólnoakademicki
Forma i tryb prowadzenia studiów	studia stacjonarne i niestacjonarne
Zakres	wszystkie specjalności
Jednostka prowadząca przedmiot	Katedra Systemów Informatycznych
Koordinator przedmiotu	dr inż. Adam Krechowicz, mgr inż. Paweł Pięta
Zatwierdził	Dziekan Wydziału Elektrotechniki, Automatyki i Informatyki dr hab. inż. Roman Deniziak, prof. PŚk

OGÓLNA CHARAKTERYSTYKA PRZEDMIOTU

Przynależność do grupy/bloku przedmiotów	przedmiot specjalnościowy	
Status przedmiotu	wybieralny	
Język prowadzenia zajęć	polski	
Usytuowanie w planie studiów - semestr	studia stacjonarne	semestr VII
	studia niestacjonarne	semestr VIII
Wymagania wstępne	Języki skryptowe, Bazy danych, Podstawy grafiki komputerowej 1 i 2, Metody obliczeniowe, Programowanie współbieżne	
Egzamin (TAK/NIE)	NIE	
Liczba punktów ECTS	6	

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

EFEKTY UCZENIA SIĘ

Kategoria	Symbol efektu	Efekty uczenia się	Odniesienie do efektów kierunkowych
Wiedza	W01	Student zna i rozumie zastosowanie wybranych zaawansowanych pakietów języka Python.	INF1_W18
	W02	Student zna i rozumie pojęcia i zagadnienia związane z tworzeniem graficznych interfejsów użytkownika w języku Python.	INF1_W18
	W03	Student zna i rozumie zaawansowane pojęcia i zagadnienia związane z programowaniem obiektowym i funkcyjnym w języku Python.	INF1_W18
	W04	Student zna i rozumie znaczenie Python Enhancement Proposals.	INF1_W18
Umiejętności	U01	Student potrafi korzystać z wybranych zaawansowanych pakietów języka Python.	INF1_U18
	U02	Student potrafi tworzyć graficzne interfejsy użytkownika w języku Python.	INF1_U18
	U03	Student potrafi tworzyć zaawansowane programy w języku Python w technice obiektowej i funkcyjnej.	INF1_U18
	U04	Student potrafi Potrafi pracować w zespole podczas rozwiązywania problemów programistycznych i dzielić je na mniejsze zadania. Potrafi opracować harmonogram prac zapewniający dotrzymanie terminów.	INF1_U18
	U05	Student potrafi Potrafi opracować sprawozdanie i dokumentację. Potrafi zaprezentować wyniki realizacji zadania.	INF1_U18
Kompetencje społeczne	K01	Student jest gotów do ciągłego dokształcania się, podnoszenia swoich kompetencji zawodowych i ich krytycznej oceny.	INF1_K01 INF1_K02

TREŚCI PROGRAMOWE

Forma zajęć	Treści programowe
wykład	<ol style="list-style-type: none"> 1. Interakcja z bazami danych. 2. Logowanie zdarzeń. 3. Parsowanie plików konfiguracyjnych. 4. Wybrane zaawansowane pakiety języka. 5. Serializacja obiektów. 6. Programowanie współbieżne. 7. Tworzenie graficznych interfejsów użytkownika i aplikacji multimedialnych. 8. Zaawansowane techniki programowania obiektowego i funkcyjnego. 9. Metaprogramowanie. Python Enhancement Proposals (PEP).
laboratorium	<ol style="list-style-type: none"> 1. Interakcja z bazami danych. 2. Logowanie zdarzeń. 3. Parsowanie plików konfiguracyjnych. 4. Wybrane zaawansowane pakiety języka. 5. Serializacja obiektów. 6. Programowanie współbieżne. 7. Tworzenie graficznych interfejsów użytkownika i aplikacji multimedialnych. 8. Zaawansowane techniki programowania obiektowego i funkcyjnego. 9. Metaprogramowanie: dekoratory funkcji i klas. Metaprogramowanie: metaklasy.
projekt	Zadania projektowe wykonywane w zespołach, polegające na zaimplementowaniu aplikacji w języku Python z zastosowaniem paradygmatów programowania obiektowego i funkcyjnego. Powstały program powinien wykorzystywać wybrane zaawansowane pakiety języka. Aplikacja powinna posiadać intuicyjny interfejs graficzny.

METODY WERYFIKACJI EFEKTÓW UCZENIA SIĘ

Symbol efektu	Metody sprawdzania efektów uczenia się					
	Egzamin ustny	Egzamin pisemny	Kolokwium	Projekt	Sprawozdanie	Inne
W01			X			
W02			X			
W03			X			
W04			X			
U01			X	X	X	X
U02			X	X	X	X
U03			X	X	X	X
U04				X	X	X
U05				X	X	X
K01				X	X	X

FORMA I WARUNKI ZALICZENIA

Forma zajęć	Forma zaliczenia	Warunki zaliczenia
wykład	zaliczenie z oceną	Uzyskanie co najmniej 50% punktów.
laboratorium	zaliczenie z oceną	Uzyskanie co najmniej 50% punktów.
projekt	zaliczenie z oceną	Dostateczne wykonanie projektu (wraz ze sprawozdaniem i dokumentacją kodu źródłowego) i jego pozytywna obrona.

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	30		30	15		18		18	9		h
2.	Inne (konsultacje, egzamin)	2		2	1		2		2	1		h
3.	Razem przy bezpośrednim udziale nauczyciela akademickiego	80					50					h
4.	Liczba punktów ECTS, którą student uzyskuje przy bezpośrednim udziale nauczyciela akademickiego	3,2					2,0					ECTS
5.	Liczba godzin samodzielnej pracy studenta	70					100					h
6.	Liczba punktów ECTS, którą student uzyskuje w ramach samodzielnej pracy	2,8					4,0					ECTS
7.	Nakład pracy związany z zajęciami o charakterze praktycznym	45					27					h
8.	Liczba punktów ECTS, którą student uzyskuje w ramach zajęć o charakterze praktycznym	1,8					1,08					ECTS
9.	Sumaryczne obciążenie pracą studenta	150					150					h
10.	Punkty ECTS za moduł <i>1 punkt ECTS=25 godzin obciążenia studenta</i>	6										ECTS

LITERATURA

1. Allen B. Downey, *Think Python*, 2nd Edition, 2015.
2. Peter Wentworth, Jeffrey Elkner, Allen B. Downey and Chris Meyers, *How to Think Like a Computer Scientist: Learning with Python 3 Documentation*, 3rd Edition, 2020.
3. Eric Matthes, *Python Crash Course*, 2nd Edition, 2019.
4. Slatkin Brett, *Effective Python*, 2nd Edition, 2019.
5. Luciano Ramalho, *Fluent Python*, 2nd Edition, 2022.
6. Michał Jaworski and Tarek Ziadé, *Expert Python Programming*, 4th Edition, 2021.
7. Rick van Hattem, *Mastering Python*, 2nd Edition, 2022.
8. Wes McKinney, *Python for Data Analysis*, 2nd Edition, 2017.
9. Christian Hill, *Learning Scientific Programming with Python*, 2nd Edition, 2020.
10. Micha Gorelick and Ian Ozsvald, *High Performance Python*, 2nd Edition, 2020.
11. Quan Nguyen, *Advanced Python Programming*, 2nd Edition, 2022.
12. Sandipan Dey, *Image Processing Masterclass with Python*, 2021.
13. Alan D. Moore, *Python GUI Programming with Tkinter*, 2nd Edition, 2021.
14. Martin Fitzpatrick, *Create GUI Applications with Python & Qt6*, 2021.
15. Dokumentacja języka Python (<https://docs.python.org/3/>).
16. Dokumentacja Qt for Python (<https://www.qt.io/qt-for-python>).