



## IV. Opis programu studiów

### 3. KARTA PRZEDMIOTU

Kod przedmiotu	<b>M#1-S1-IP-305</b>
Nazwa przedmiotu	<b>Język skryptowy</b>
Nazwa przedmiotu w języku angielskim	<b>Script language</b>
Obowiązuje od roku akademickiego	<b>2020/2021</b>

#### USYTUOWANIE MODUŁU W SYSTEMIE STUDIÓW

Kierunek studiów	<b>INFORMATYKA PRZEMYSŁOWA</b>
Poziom kształcenia	<b>I stopień</b>
Profil studiów	<b>ogólnoakademicki</b>
Forma i tryb prowadzenia studiów	<b>studia stacjonarne</b>
Zakres	<b>wszystkie</b>
Jednostka prowadząca przedmiot	<b>Katedra Technik Komputerowych i Uzbrojenia</b>
Koordynator przedmiotu	<b>dr inż. Dariusz Michalski</b>
Zatwierdził	

#### OGÓLNA CHARAKTERYSTYKA PRZEDMIOTU

Przynależność do grupy/bloku przedmiotów	<b>przedmiot podstawowy</b>
Status przedmiotu	<b>obowiązkowy</b>
Język prowadzenia zajęć	polski
Usytuowanie modułu w planie studiów - semestr	<b>semestr 3</b>
Wymagania wstępne	
Egzamin (TAK/NIE)	TAK
Liczba punktów ECTS	<b>4</b>

Forma prowadzenia zajęć	wykład	ćwiczenia	laboratorium	projekt	seminarium
Liczba godzin w semestrze	<b>15</b>		<b>30</b>		

## EFEKTY UCZENIA SIĘ

Kategoria	Symbol efektu	Efekty kształcenia	Odniesienie do efektów kierunkowych
Wiedza	W01	Student ma elementarną wiedzę w zakresie języka skryptowego Python,	IP1_W11
	W02	Student orientuje się w obecnym stanie oraz najnowszym trendach rozwojowych języków skryptowych w szczególności języka Python.	IP1_W19
	W03	Student zna i rozumie zasady projektowania, implementacji i procesy zachodzące w cyklu życia systemów informatycznych.	IP1_W20
Umiejętności	U01	Student potrafi pozyskiwać informacje z literatury, baz danych i innych źródeł; potrafi integrować uzyskane informacje, dokonywać ich interpretacji, a także wyciągać wnioski oraz formułować i uzasadniać opinie.	IP1_U01
	U02	Student potrafi pracować indywidualnie i w zespole; umie oszacować czas potrzebny na realizację zleconego zadania; potrafi opracować i zrealizować harmonogram prac zapewniający dotrzymanie terminów	IP1_U02
	U03	Student potrafi samodzielnie zaplanować samokształcenie i realizować uczenie się przez całe życie, porozumiewać się z wykorzystaniem różnych technik w środowisku zawodowym oraz podnosić kompetencje zawodowe.	IP1_U04
	U04	Student potrafi posługiwać się językiem obcym w obszarze słownictwa technicznego. Posługuje się językiem obcym w stopniu wystarczającym do czytania ze zrozumieniem podręczników obsługi narzędzi informatycznych oraz podobnych dokumentów w tym dotyczących języka Python.	IP1_U05
	U05	Potrafi tworzyć proste aplikacje w języku Python do zastosowań informatyki przemysłowej działające w różnych środowiskach sprzętowych i programowych wraz z zaprojektowaniem dla nich funkcjonalnego i użytecznego interfejsu użytkownika	IP1_U20
	U06	Student potrafi napisać program komputerowy w języku wysokiego poziomu Python.	P1_U25
	U07	Student potrafi implementować algorytmy z użyciem języka Python	IP1_U26
Kompetencje społeczne	K01	Rozumie potrzebę i zna możliwości ciągłego doskonalenia (studia II i III stopnia, studia podyplomowe, kursy) mającego na celu podnoszenie kompetencji zawodowych, osobistych i społecznych.	IP1_K01
	K02	Ma świadomość znaczenia i rozumie pozatechniczne aspekty i skutki działalności inżyniera informatyka przemysłowego, w aspekcie oddziaływania na środowisko i odpowiedzialności za podejmowane decyzje.	IP1_K02
	K03	Ma świadomość znaczenia profesjonalnego działania, przestrzegania zasad etyki zawodowej i poszanowania różnorodności poglądów i kultur i religii.	IP1_K03
	K04	Ma świadomość odpowiedzialności za pracę własną oraz gotowość podporządkowania się zasadom pracy w zespole i ponoszenia odpowiedzialności za wspólnie realizowane zadania.	IP1_K04

## TREŚCI PROGRAMOWE

Forma zajęć*	Treści programowe
wykład	1. Środowiska do programowania w Pythonie.
	2. Podstawowe typy danych, literały, operatory i wyrażenia.
	3. Wyrażenia warunkowe i pętle.
	4. Funkcje, skrypty i moduły.
	5. Projektowanie odgórne, testowanie i debugowanie.
	6. Listy, krotki, zbiory i słowniki.
	7. Programowanie obiektowe: klasy i obiekty.
	8. Grafika.
	9. Operacje na plikach.
laboratorium	1. Instalacja i konfiguracja środowiska do programowania Python
	2. Python w telegraficznym skrócie. Podstawowe składnie języka. Podstawowe typy danych i instrukcje sterujące
	3. Listy, krotki i ciągi znaków, złożone struktury danych, operacje na ciągach znaków
	4. Słowniki, słownik w Pythonie, wykorzystywanie słowników
	5. Funkcje, przekazywanie parametrów do funkcji, zwracanie wyników działania funkcji, tworzenie i wywoływanie funkcję o zmiennej liczbie parametrów zasady zasięgu zmiennych
	6. Funkcje wbudowane, podstawowe funkcje wbudowane języka Python funkcje anonimowe
	7. Generatory, czym są generatory, tworzenie i wykorzystanie generatorów wyrażenia generatora
	8. Operacje na plikach, wykonywanie operacji na plikach
	9. Moduły i pakiety. Co to jest moduł ? Co to jest pakiet ? Tworzenie i wykorzystanie własnych modułów i pakietów.
	10. Klasy, tworzenie hierarchii klas, wykorzystanie utworzonych klas
	11. Wyjątki, czym są wyjątki ? Zasady obsługi wyjątków w Pythonie
	12. Iteratory, czym jest iterator ? Tworzenie i wykorzystanie iteratorów
	13. Obliczenia numeryczne w Pythonie, dostępne moduły numeryczne, wykorzystanie modułów numerycznych do obliczeń
	14. Obliczenia symboliczne w Pythonie pakiet SAGE pakiet sympy
	15. Tworzenia aplikacji okienkowych wykorzystanie wieloplatformowej biblioteki GTK+

## METODY WERYFIKACJI EFEKTÓW UCZENIA SIĘ

Symbol efektu	Metody sprawdzania efektów kształcenia (zaznaczyć X)					
	Egzamin ustny	Egzamin pisemny	Kolokwium	Projekt	Sprawozdanie	Inne
W01		X	X			
W02		X			X	
W03		X			X	
U01		X	X		X	
U02			X		X	
U03			X		X	
U04			X		X	
U05		X	X		X	
U06		X	X		X	
U07		X	X		X	
K01						X
K02						X

K03					X	X
K04					X	X

### FORMA I WARUNKI ZALICZENIA

Forma zajęć*	Forma zaliczenia	Warunki zaliczenia
wykład	egzamin	Uzyskanie co najmniej 50% punktów z rozwiązania zadań egzaminacyjnych. Wysokość oceny końcowej będzie ustalana zgodnie ze skalą ocen obowiązującą w regulaminie PŚK.
laboratorium	zaliczenie z oceną	Każde zajęcia laboratoryjne rozpoczynają się krótką kartkówką z materiału wcześniejszego, za które można uzyskać 20% ogółu punktów. 60 % ogółu punktów można uzyskać za programy napisany w czasie zajęć. Pozostałe 20% ogółu punktów można otrzymać za programy napisane podczas samodzielnej pracy studenta w grupie oddane na następnym zajęciu wraz z dokumentacją. Przy obliczaniu oceny końcowej z laboratorium anulowane są po jednej najsłabszej ocenie z: kartkówek, programów pisanych na zajęcia i w ramach pracy samodzielnej. Wysokość oceny końcowej będzie ustalana zgodnie ze skalą ocen obowiązującą w regulaminie PŚK.

### NAKŁAD PRACY STUDENTA

Bilans punktów ECTS							
Lp.	Rodzaj aktywności	Obciążenie studenta					Jednostka
		W	C	L	P	S	
1.	Udział w zajęciach zgodnie z planem studiów	15		30			h
2.	Inne (konsultacje, egzamin)	4		2			h
3.	<b>Razem przy bezpośrednim udziale nauczyciela akademickiego</b>	<b>51</b>					h
4.	<b>Liczba punktów ECTS, którą student uzyskuje przy bezpośrednim udziale nauczyciela akademickiego</b>	<b>2,0</b>					ECTS
5.	<b>Liczba godzin samodzielnej pracy studenta</b>	<b>49</b>					h
6.	<b>Liczba punktów ECTS, którą student uzyskuje w ramach samodzielnej pracy</b>	<b>2,0</b>					ECTS
7.	<b>Nakład pracy związany z zajęciami o charakterze praktycznym</b>	<b>67</b>					h
8.	<b>Liczba punktów ECTS, którą student uzyskuje w ramach zajęć o charakterze praktycznym</b>	<b>2,7</b>					ECTS
9.	<b>Sumaryczne obciążenie pracą studenta</b>	<b>100</b>					h
10.	<b>Punkty ECTS za moduł</b> <i>1 punkt ECTS=25 godzin obciążenia studenta</i>	<b>4</b>					ECTS

### LITERATURA

1. Allen Downey, Jeffrey Elkner, Chris Meyers, "Think Python. How to Think Like a Computer Scientist", GreenTea Press Wellesley, Massachusetts 2002  
(<http://www.greenteapress.com/thinkpython/>)
2. M. Pilgrim, "Dive into Python 3", (<http://www.diveintopython.net/>)
3. Edition Peter Wentworth, Jeffrey Elkner, Allen B. Downey and Chris Meyers "How to Think Like a Computer Scientist: Learning with Python 3" Documentation Release 3rd Edition
4. Swaroop C H A "Byte of Python"  
[https://www.ibiblio.org/swaroopch/byteofpython/files/120/byteofpython\\_120.pdf](https://www.ibiblio.org/swaroopch/byteofpython/files/120/byteofpython_120.pdf)
5. David Beazley and Brian K. Jones "Python Cookbook, Third Edition"  
[https://d.cxcore.net/Python/Python\\_Cookbook\\_3rd\\_Edition.pdf](https://d.cxcore.net/Python/Python_Cookbook_3rd_Edition.pdf)
6. Mark Lutz Learning "Python FOURTH EDITION" O'Reilly Media 2009  
[https://cfm.ehu.es/ricardo/docs/python/Learning\\_Python.pdf](https://cfm.ehu.es/ricardo/docs/python/Learning_Python.pdf)
7. Peter Norton, „Python: od podstaw“, Helion, 2006.
8. Langtangen, Hans Petter. "Python scripting for computational science" Berlin; Heidelberg: Springer, cop. 2009.
9. Sandro Tosi "Matplotlib for Python Developers" Packt Publishing, 2009.
10. Wes McKinney, "Python for Data Analysis", O'Reilly Media, Inc., 2012.