



Akademia Nauk Stosowanych
im. Hipolita Cegielskiego w Gnieźnie Uczelnia Państwowa

SYLABUS

Pozycja przedmiotu w planie:		R.II/S.III – 6
1. OGÓLNY OPIS PRZEDMIOTU		
1	Nazwa modułu	Moduł treści kierunkowych
2	Nazwa przedmiotu	Programowanie skryptowe
3	Kierunek studiów	Informatyka
4	Poziom studiów	pierwszy
5	Forma studiów	stacjonarne
6	Profil studiów	praktyczny
7	Rok studiów	drugi
8	Semestr przedmiotu	trzeci
9	Jednostka prowadząca kierunek studiów	Instytut Nauk Technicznych
10	Liczba punktów ECTS	3
11	Sposób zaliczenia:	wykład, laboratorium, projekt: zaliczenie z oceną
12	Imię i nazwisko nauczyciela (li) akademickiego (ich), stopień lub tytuł naukowy, adres e-mail	mgr inż. Mateusz Leszek <i>m.leszek@ans-gniezno.edu.pl</i>
13	Imię i nazwisko koordynatora(ów) przedmiotu, stopień lub tytuł naukowy, adres e-mail	mgr inż. Mateusz Leszek <i>m.leszek@ans-gniezno.edu.pl</i>
14	Język wykładowy	polski
15	Tryb prowadzenia zajęć	mieszany
16	Sposób prowadzenia zajęć	synchroniczny
17	Narzędzia informatyczne wykorzystywane do prowadzenia zajęć, udostępniania materiałów i komunikacji ze studentami	Microsoft Teams / platforma Moodle
15	Przedmioty wprowadzające	- Wprowadzenie do programowania, - Programowanie obiektowe.
16	Wymagania wstępne	1. Podstawowa wiedza z programowania w języku C++, 2. Podstawowa wiedza z algorytmiki.
17	Cele przedmiotu:	
C1	Nabycie przez studentów wiedzy i umiejętności posługiwania się skryptowym językiem programowania, na przykładzie języka Python 3, LUA oraz RUBY.	
C2	Zaznajomienie z podstawowymi bibliotekami języka Python 3.	
C3	Przedstawienie przykładów zastosowania omówionych języków w dziedzinach zgodnych z wybranym kierunkiem studiów.	

18	Forma zajęć, liczba godzin wymagająca bezpośredniego udziału nauczyciela akademickiego, liczba godzin nakładu pracy studenta	
Forma zajęć		Liczba godzin
1. wykład		15
2. laboratorium		30
3. projekt		15
Suma godzin:		60
L.p.	Całkowity nakład pracy studenta	
1.	Nakład pracy związany z zajęciami wymagającymi bezpośredniego udziału nauczyciela akademickiego wynosi:	Godzinowe obciążenie studenta
	wykład: 15 godz.	60 godzin
	laboratorium: 30 godz.	
	projekt: 15 godz.	
Nakład pracy związany z zajęciami wymagającymi bezpośredniego udziału nauczyciela akademickiego wynosi 60 godzin , co odpowiada 2 punktom ECTS .		
2	Bilans nakładu pracy studenta: <ul style="list-style-type: none"> przygotowanie do laboratoriów: 5 godzin, przygotowanie projektu: 20 godzin. Łączny nakład pracy studenta wynosi 25 godzin , co odpowiada 1 punktom ECTS .	25 godzin
3	Łączny nakład pracy studenta (pozycja 1+2)	85 godzin
4	Punkty ECTS za przedmiot	3 ECTS
5	Liczba punktów ECTS, którą student musi osiągnąć w ramach zajęć o charakterze praktycznym w tym zajęć laboratoryjnych, warsztatowych, projektowych	2.5 ECTS
Efekty uczenia się - wiedza	K_W04: ma wiedzę w zakresie znajomości paradygmatu obiektowego, języków skryptowych oraz ich zastosowania, zna podstawowe konstrukcje programistyczne i zasady programowania w językach skryptowych.	
	K_W05: student ma ogólną wiedzę z zakresu teorii i metod formułowania, konstruowania i stosowania algorytmów; zna struktury danych w omawianych językach skryptowych	
	K_W24: ma wiedzę z zakresu informatyki w zakresie inżynierskim pozwalającym tworzyć i wykorzystywać oprogramowanie w obszarze podstawowych zastosowań informatyki.	
Efekty uczenia się - umiejętności	K_U01: potrafi samodzielnie pozyskiwać informacje z literatury, baz danych i innych źródeł i efektywnie pozyskiwać wiedzę, w tym w systemie kształcenia zdalnego (blended/e-learning); potrafi scalać i interpretować uzyskane informacje.	
	K_U13: potrafi pisać, uruchamiać, śledzić i testować programy w środowisku programistycznym PyCharm wykorzystując znajomość paradygmatów i metod programowania w omawianych językach skryptowych (Python, LUA, Ruby).	
	K_U14: ma umiejętności: implementowania algorytmów z wykorzystaniem podstawowych technik algorytmicznych.	
Efekty uczenia się – kompetencje społeczne	K_K01: rozumie potrzebę i zna możliwości ciągłego dokształcania się, krytycznie odnosi się do posiadanej wiedzy, podnoszenia kompetencji zawodowych, osobistych i społecznych.	
	K_K04: ma świadomość odpowiedzialności za pracę własną oraz gotowość podporządkowania się zasadom pracy w zespole i ponoszenia odpowiedzialności za wspólnie realizowane zadania, potrafi określić priorytety służące realizacji określonego przez siebie lub zespół zadania.	

2. TREŚCI PROGRAMOWE ODNIESIONE DO EFEKTÓW UCZENIA SIĘ		
	Treści programowe	liczba godzin
Forma: wykład		
W1	Wprowadzenie do skryptowych języków programowania	2
W2	Wstęp do języka Python 3	2
W3	Klasy i obiekty	2
W4	Najpopularniejsze biblioteki do Python'a	2
W5	PyQT – tworzenie GUI dla skryptów Pythona	2
W6	Podstawy języka Ruby	2
W7	Podstawy języka LUA	2
Forma: laboratoria		
L1	Podstawowa składnia Pythona 3	3
L2	Struktury danych w języku Python3 - listy, słowniki, krotki	3
L3	Obliczenia matematyczne i wykresy – biblioteka NumPy, SciPy, Matplotlib	3
L4	Klasy i obiekty w języku Python	3
L5	Operacje na systemie plików przy pomocy Python'a	3
L6	PyQT5 – tworzenie GUI dla aplikacji Python'owych	3
L7	Podstawowa składnia języka Ruby	3
L8	Struktury danych w Ruby	3
L9	Podstawy języka LUA	3
L10	Struktury danych w LUA	3
Forma: projekt		
P1	Omówienie założeń projektowych.	1,5
P2	Przedstawienie wybranych tematów i określenie założeń i celów projektowych	1,5
P3	Realizacja projektu	10
P4	Przedstawienie projektu i zaliczenie	1,5

3. Literatura	
Literatura podstawowa	1. Lutz M., Python. Wprowadzenie, Wyd. V, Helion, 2020. - https://biblioteka.ans-gniezno.edu.pl/?det&oid=9612&dt=0 2. McKinney W., Python w analizie danych : przetwarzanie danych za pomocą pakietów Pandas i NumPy oraz środowiska IPython, Helion, 2018. - https://biblioteka.ans-gniezno.edu.pl/?det&oid=9627&dt=0 3. Johansson R., Matematyczny Python. Obliczenia naukowe i analiza danych z użyciem NumPy, SciPy i Matplotlib, Helion, 2021 - https://biblioteka.ans-gniezno.edu.pl/?det&oid=12530&dt=0
Literatura uzupełniająca	1. https://matplotlib.org/contents.html - dokumentacja biblioteki Matplotlib 2. https://www.lua.org/docs.html - dokumentacja języka LUA 3. https://pandas.pydata.org/docs/ - dokumentacja biblioteki Pandas 4. https://www.riverbankcomputing.com/static/Docs/PyQt5/ - dokumentacja biblioteki PyQt5 5. https://www.ruby-lang.org/pl/ - strona poświęcona językowi Ruby

4. Metody dydaktyczne	
Forma:	Metody dydaktyczne:
wykład	wykład informacyjny – prezentacja multimedialna, opowiadanie.
laboratoria	metody ćwiczeniowo- praktyczne, metoda doświadczeniowa – prezentacja multimedialna
projekt	dyskusja

5. Metody i kryteria oceniania	
Forma zajęć: wykład	Forma zaliczenia: zaliczenie z oceną
Uzyskane punkty są przeliczane na oceny według następującej skali: Procent punktów: Ocena: 91-100% Bardzo dobry 85-90% Dobry plus 76-84% Dobry 66-75% Dostateczny plus 51-65% Dostateczny 0-50% Niedostateczny	
Opis: Zaliczenie w formie testu jedнокrotnego wyboru na platformie Moodle ze znajomości składni języka Python oraz omawianych bibliotek.	
Forma zajęć: laboratoria	Forma zaliczenia: zaliczenie z oceną
Uzyskane punkty są przeliczane na oceny według następującej skali: Procent punktów: Ocena: 91-100% Bardzo dobry 85-90% Dobry plus 76-84% Dobry 66-75% Dostateczny plus 51-65% Dostateczny 0-50% Niedostateczny	
Opis: Średnia ocen ze sprawozdań ocenianych w toku zajęć laboratoryjnych.	
Forma zajęć: projekt	Forma zaliczenia: zaliczenie z oceną
Uzyskane punkty są przeliczane na oceny według następującej skali: Procent punktów: Ocena: 91-100% Bardzo dobry 85-90% Dobry plus 76-84% Dobry 66-75% Dostateczny plus 51-65% Dostateczny 0-50% Niedostateczny	
Opis: Realizacja założeń projektowych we wskazanym przez studenta temacie.	
Warunkiem zaliczenia przedmiotu jest uzyskanie oceny pozytywnej ze wszystkich form zajęć.	

	Zatwierdzenie karty opisu zajęć	
	Stanowisko Tytuł/stopień naukowy, imię nazwisko	Podpis
Opracował	mgr inż. Mateusz Leszek	
Zatwierdził	Dyrektor Instytutu Nauk Technicznych	