

Karta przedmiotu

Nazwa i kod przedmiotu	Wprowadzenie do programowania (Ćw. audytoryjne), PG_00150257						
Kierunek studiów	Filologia angielska (O)						
Data rozpoczęcia studiów	październik 2024 r.	Rok akademicki realizacji przedmiotu			2024/2025		
Poziom kształcenia	II stopnia	Grupa zajęć			Grupa zajęć obowiązkowych z zakresu kierunku studiów Grupa zajęć fakultatywnych Grupa zajęć powiązanych z prowadzonymi badaniami naukowymi w dziedzinie nauki związanej z kierunkiem - profil ogólnoakademicki		
Forma studiów	stacjonarne	Sposób realizacji			na uczelni		
Rok studiów	1	Język wykładowy			polski polski 90% angielski 10%		
Semestr studiów	2	Liczba punktów ECTS			4.0		
Profil kształcenia	ogólnoakademicki	Forma zaliczenia			zaliczenie		
Jednostka prowadząca	Rektor -> Wydział Filologiczny -> Instytut Anglistyki i Amerykanistyki -> Zakład Językoznawstwa Korpusowego i Glottodydaktyki						
Imię i nazwisko wykładowcy (wykładowców)	Odpowiedzialny za przedmiot		Kamil Soliwoda				
	Prowadzący zajęcia z przedmiotu		Kamil Soliwoda				
Formy zajęć	Forma zajęć	Wykład	Ćwiczenia	Laboratorium	Projekt	Seminarium	RAZEM
	Liczba godzin zajęć	0.0	60.0	0.0	0.0	0.0	60
	W tym liczba godzin zajęć na odległość: 0.0						
Aktywność studenta i liczba godzin pracy	Aktywność studenta	Udział w zajęciach dydaktycznych, objętych planem studiów		Udział w konsultacjach		Praca własna studenta	RAZEM
	Liczba godzin pracy studenta	60		0.0		36.0	96
Cel przedmiotu	Celem przedmiotu jest zaznajomienie studentów z podstawowymi technikami i metodami programowania, na przykładzie języka programowania Python.						

Efekty uczenia się przedmiotu	Efekt kierunkowy	Efekt z przedmiotu	Sposób weryfikacji i oceny efektu
	[FAMU2_K02] Jest gotów do uznawania znaczenia wiedzy i umiejętności z zakresu studiów anglistycznych w rozwiązywaniu problemów poznawczych i praktycznych oraz do zasięgania opinii opiekuna naukowego lub opiekuna w wybranym miejscu pracy w przypadku trudności z samodzielnym rozwiązywaniem problemów.	Student jest gotów zasygnalizować napotkane trudności poznawcze i praktyczne oraz korzysta z pomocy w przypadku problemów.	[SK2] prezentacja/projekt/referat/raport [SK5] realizacja zadania problemowego
	[FAMU2_W12] Zna i rozumie wybrane zagadnienia z zakresu nauk pomocniczych i pokrewnych filologii angielskiej oraz realioznawstwa krajów anglojęzycznych, niezbędne jako kontekst dla badań w zakresie anglistycznego językoznawstwa i literatur anglojęzycznych lub w działalności zawodowej filologa anglisty.	Student zna zastosowanie i sposoby wykorzystania nabytych umiejętności dotyczących podstaw programowania w codziennej pracy filologa anglisty	[SW4] test/egzamin - ustny lub pisemny [SW2] prezentacja/projekt/referat/raport [SW5] realizacja zadania problemowego
	[FAMU2_K01] Jest gotów do krytycznej oceny zakresu posiadanej przez siebie wiedzy i umiejętności, w szczególności z zakresu anglojęzycznego językoznawstwa i literaturoznawstwa oraz języka angielskiego.	Student ma świadomość ograniczenia własnej wiedzy oraz umiejętności i rozumie potrzebę uczenia się przez całe życie, a także pogłębiania i uzupełniania nabytej wiedzy i umiejętności; wyznacza kierunki własnego rozwoju i uczenia się.	[SK2] prezentacja/projekt/referat/raport [SK5] realizacja zadania problemowego
	[FAMU2_U04] Potrafi dobierać i stosować techniki informacyjno-komunikacyjne (ICT) podczas pozyskiwania i przetwarzania informacji dla celów badawczych i zawodowych w ramach filologii angielskiej.	Student umie analizować i interpretować zebrane dane za pomocą programów komputerowych (np. Excel, Google Sheets) oraz tworzyć i edytować dokumenty tekstowe, prezentacje multimedialne i raporty.	[SU2] prezentacja/projekt/referat/raport [SU4] test/egzamin - ustny lub pisemny [SU5] realizacja zadania problemowego
	[FAMU2_U10] Potrafi samodzielnie planować i realizować własne uczenie się przez całe życie i ukierunkowywać innych w tym zakresie w ramach filologii angielskiej i wybranej sfery działalności zawodowej.	Student posiada umiejętność planowania pracy własnej i systematycznego przygotowywania się do wyznaczonych przez prowadzącego zadań.	[SU2] prezentacja/projekt/referat/raport [SU4] test/egzamin - ustny lub pisemny [SU5] realizacja zadania problemowego
Treści przedmiotu	<ul style="list-style-type: none"> • Pojęcie programu. Instalacja i wprowadzenie do środowiska języka Python. • Podstawowe typy zmiennych. • Pojęcie funkcji w programowaniu. • Sterowanie programem za pomocą instrukcji warunkowych w programowaniu. • Struktury cykliczne w języku Python: pętle for oraz while. • Zaawansowane struktury danych: listy, krotki, zbiory oraz słowniki. • Operacje wejścia-wyjścia oraz na plikach w języku Python. • Pojęcie algorytmu i schematu blokowego. Podstawowe algorytmy wyszukiwania oraz sortowania. • Technika dzieli i zwyciężaj w programowaniu. • Podstawowe algorytmy tekstowe. • Pojęcie rekurencji w programowaniu. • Zarządzanie bibliotekami w języku Python. • Wyrażenia regularne w języku Python. • Obsługa plików tekstowych w języku Python. • Format JSON oraz ich użycie w języku Python • Zastosowanie bibliotek: numpy, nltk, matplotlib, seaborn, pygal. 		
Wymagania wstępne i dodatkowe			
Sposoby i kryteria oceniania osiągniętych efektów uczenia się	Sposób oceniania (składowe)	Próg zaliczeniowy	Składowa oceny końcowej
	Test	51.0%	60.0%
	Projekt	51.0%	20.0%
	Uczestnictwo	51.0%	20.0%

Zalecana lista lektur	Podstawowa lista lektur	<p>P. Broda, D. Smołuca, Informatyka. Podrecznik czesc II, Wyd. Operon, 2006.</p> <p>M. Dawson, Python dla kazdego. Podstawy programowania, Wyd. Helion, 2010.</p> <p>T. Gaddins, Python dla zupełnie poczatkujacych, Wyd. Helion, 2019.</p> <p>E. Grubiel, G. Hardt-Olejniczak, E. Kołczyk, H. Krupicka, M. M. Sysło, Informatyka czesc I, Wyd. WSiP, 2002.</p> <p>A. Sweigart, Automatyzacja nudnych zadan z Pythonem. Nauka programowania, Wyd. Helion, 2015.</p> <p>A. Szepietowski, Matematyka dyskretna, Wyd. UG 2004.</p> <p>P. Wroblewski, Algorytmy, struktury danych i techniki programowania, Wyd. Helion, 2010.</p>
	Uzupełniająca lista lektur	<p>T. H. Cormen, Ch. E. Leiserson, R. L. Rivest, Wprowadzenie do algorytmow, Wyd. Naukowe PWN, 2012.</p> <p>P. Krugiołka, Linux. Jak dostroic bestie do swoich potrzeb, Wyd. Helion, 2012.</p> <p>E. Matthes, Python. Instrukcje dla programisty, Wyd. Helion, 2016.</p>
	Adresy eZasobów	<p>Podstawowe</p> <p>http://www.pygal.org/en/stable/ - Dostęp: 15.07.2024</p> <p>https://edabit.com - Dostęp: 15.07.2024</p> <p>https://seaborn.pydata.org/ - Dostęp: 15.07.2024</p> <p>https://exercism.io/ - Dostęp: 15.07.2024</p> <p>https://matplotlib.org/stable/index.html - Dostęp: 15.07.2024</p> <p>https://www.codecademy.com - Dostęp: 15.07.2024</p> <p>Adresy na platformie eNauczanie:</p>

<p>Przykładowe zagadnienia/ przykładowe pytania/ realizowane zadania</p>	<ol style="list-style-type: none"> 1. Pojęcie programu. Instalacja i wprowadzenie do środowiska języka Python: <ul style="list-style-type: none"> • Co to jest program komputerowy? • Jakie są kroki instalacji środowiska Python na różnych systemach operacyjnych (Windows, macOS, Linux)? • Wprowadzenie do IDE (np. PyCharm, VS Code, Jupyter Notebook). 1. Podstawowe typy zmiennych: <ul style="list-style-type: none"> • Jakie są podstawowe typy danych w Pythonie (int, float, str, bool)? • Jakie operacje można wykonywać na różnych typach danych? • Ćwiczenie: Napisz program, który przyjmuje liczby całkowite od użytkownika i wykonuje podstawowe operacje arytmetyczne. 1. Pojęcie funkcji w programowaniu: <ul style="list-style-type: none"> • Jak definiować funkcje w Pythonie? • Co to są argumenty i wartości zwracane przez funkcję? • Ćwiczenie: Napisz funkcję, która oblicza silnię liczby. 1. Sterowanie programem za pomocą instrukcji warunkowych w programowaniu: <ul style="list-style-type: none"> • Jak działają instrukcje warunkowe if, elif, else? • Ćwiczenie: Napisz program, który sprawdza, czy liczba jest parzysta czy nieparzysta. 1. Struktury cykliczne w języku Python: pętle for oraz while: <ul style="list-style-type: none"> • Jak działa pętla for i while? • Ćwiczenie: Napisz program, który wyświetla liczby od 1 do 10 przy użyciu obu typów pętli. 1. Zaawansowane struktury danych: listy, krotki, zbiory oraz słowniki: <ul style="list-style-type: none"> • Jakie są różnice między listami, krotkami, zbiorami a słownikami? • Ćwiczenie: Napisz program, który sortuje listę liczb i usuwa duplikaty. 1. Operacje wejścia-wyjścia oraz na plikach w języku Python: <ul style="list-style-type: none"> • Jak otwierać, czytać i zapisywać pliki tekstowe w Pythonie? • Ćwiczenie: Napisz program, który zapisuje dane użytkownika do pliku tekstowego. 1. Pojęcie algorytmu i schematu blokowego. Podstawowe algorytmy wyszukiwania oraz sortowania: <ul style="list-style-type: none"> • Co to jest algorytm i jak tworzyć schematy blokowe? • Jak działają algorytmy sortowania (np. sortowanie bąbelkowe, szybkie sortowanie) i wyszukiwania (np. wyszukiwanie binarne)? • Ćwiczenie: Napisz program implementujący sortowanie bąbelkowe. 1. Technika dziel i zwyciężaj w programowaniu: <ul style="list-style-type: none"> • Jakie są podstawowe zasady techniki dziel i zwyciężaj? • Ćwiczenie: Napisz program, który realizuje algorytm sortowania szybkiego (QuickSort). 1. Podstawowe algorytmy tekstowe: <ul style="list-style-type: none"> • Jak działają podstawowe algorytmy przetwarzania tekstu (np. wyszukiwanie wzorca w tekście)? • Ćwiczenie: Napisz program, który znajduje wszystkie wystąpienia danego wzorca w tekście. 1. Pojęcie rekurencji w programowaniu: <ul style="list-style-type: none"> • Co to jest rekurencja i jak ją stosować w programowaniu? • Ćwiczenie: Napisz funkcję rekurencyjną obliczającą n-ty wyraz ciągu Fibonacciego. 1. Zarządzanie bibliotekami w języku Python: <ul style="list-style-type: none"> • Jakie są metody zarządzania bibliotekami w Pythonie (pip, virtualenv)? • Ćwiczenie: Zainstaluj i zaimportuj bibliotekę numpy w swoim projekcie. 1. Wyrażenia regularne w języku Python: <ul style="list-style-type: none"> • Jak definiować i używać wyrażeń regularnych w Pythonie? • Ćwiczenie: Napisz program, który sprawdza, czy podany przez użytkownika adres e-mail jest poprawny. 1. Obsługa plików tekstowych w języku Python: <ul style="list-style-type: none"> • Jakie są metody odczytu i zapisu plików tekstowych? • Ćwiczenie: Napisz program, który czyta dane z pliku tekstowego i wyświetla je na ekranie. 1. Format JSON oraz ich użycie w języku Python: <ul style="list-style-type: none"> • Jak pracować z plikami JSON w Pythonie (odczyt, zapis, modyfikacja)? • Ćwiczenie: Napisz program, który odczytuje dane z pliku JSON i wyświetla je w formie tabeli. 1. Zastosowanie bibliotek: numpy, nltk, matplotlib, seaborn, pygal: <ul style="list-style-type: none"> • Jakie są zastosowania bibliotek numpy, nltk, matplotlib, seaborn, pygal? • Ćwiczenie: Napisz program, który wykorzystuje bibliotekę numpy do przetwarzania danych numerycznych oraz bibliotekę matplotlib do tworzenia wykresów. <p>Przykładowe projekty:</p> <ul style="list-style-type: none"> • Projekt 1: Napisz program do zarządzania kontaktami, który pozwala na dodawanie, usuwanie i wyszukiwanie kontaktów. Program powinien zapisywać dane do pliku tekstowego. • Projekt 2: Stwórz aplikację analizującą tekst, która liczy słowa, zdania i litery w podanym tekście oraz wyświetla najczęściej występujące słowa.
<p>Praktyki zawodowe w ramach przedmiotu</p>	<p>Nie dotyczy</p>

Dokument wygenerowany elektronicznie. Nie wymaga pieczęci ani podpisu.