

KARTA PRZEDMIOTU OFEROWANEGO W SZKOLE DOKTORSKIEJ

Kod przedmiotu	4606-PS-0000000-0111	Nazwa przedmiotu	w j. polskim	Podstawy programowania w języku Python		
			w j. angielskim	Fundamentals of Python programming		
Przynależność do grupy przedmiotów	specjalnościowe					
Koordinator przedmiotu	dr hab. inż. Kamil Padaszyński					
Jednostka realizująca	Wydział Chemiczny, Katedra Chemii Fizycznej	Dyscyplina/y naukowa*	wszystkie			
Poziom kształcenia	Kształcenie doktorantów	Semestr	letni			
Język zajęć	polski					
Forma zaliczenia:	Zaliczenie na ocenę	Sumaryczna liczba godzin w semestrze	30	Sumaryczna liczba ECTS	3	
Minimalna liczba uczestników	10	Maksymalna liczba uczestników	18	Dostępność dla studentów	Nie	
Typ zajęć		Wykład	Ćwiczenia audytorjne	Ćwiczenia projektowe	Laboratorium	Seminarium
Liczba godzin zajęć	tygodniowo				2	
	łącznie w semestrze				30	

* nie dotyczy warsztatu badacza

1. Wymagania wstępne

Podstawy obsługi komputera osobistego (operacje na plikach, struktura katalogów) oraz matematyka i statystyka na podstawowym ogólno-akademickim (analiza matematyczna, algebra liniowa). Język angielski na poziomie umożliwiającym czytanie dokumentacji (główna pozycja zalecanej temperatury).

2. Cele przedmiotu

Python jest elastycznym, wydajnym i wszechstronnym językiem programowania o stałe rosnącej popularności i społeczności — nie tylko w branży IT, ale również wśród naukowców, dla których programowanie stało się w ostatnich latach podstawowym narzędziem warsztatu badawczego.

Celem zajęć jest zapoznanie słuchaczy z podstawą języka w stopniu umożliwiającym rozwiązywanie i zautomatyzowanie prostych i średnio zaawansowanych problemów związanych zarówno z prowadzonymi zadaniami naukowo-badawczymi, ale również z codziennymi (często żmudnymi) zadaniami wymagających wykonania operacji na plikach lub analizy danych zapisanych lokalnie lub udostępnianych przez sieć internet.

3. Treści programowe (dla każdego typu zajęć oddzielnie)

Laboratorium

1. Podstawowe informacje nt. języka Python. Instalacja oraz konfiguracja interpretera i środowiska programistycznego. Postawy pracy w powłoce systemowej. Pierwszy program „Hello world”.
2. Podstawy pracy w systemie kontroli wersji git oraz platformie GitHub.
3. Zmienne, podstawowe typy i struktury danych (int, float, string, NoneType), operatory i literały.
4. Struktura programu, moduły i pakiety. Środowiska wirtualne. Instalacja dodatkowych bibliotek z PyPI: menadżery pakietów pip i pipenv.
5. Dane tekstowe. Moduł i klasa string.
6. Typy iterowalne (listy, krotki, słowniki, zbiory) i ich podstawowe metody
7. Instrukcje warunkowe, pętle. Diagnostyka kodu (debugowanie) i obsługa wyjątków.
8. Funkcje, generatory, dekoratory.
9. Pliki tekstowe, operacje na plikach (biblioteki os i pathlib).
10. Interakcja programów z zasobami internetowymi, biblioteka requests.
11. Wstęp do programowania zorientowanego obiektowo. Klasy.
12. Przybornik programisty Pythona-inżyniera. Krótkie wprowadzenie do bibliotek numpy, pandas, scipy, matplotlib, sympy.

Każdemu z powyższych punktów towarzyszą przykłady omawiane na zajęciach oraz zadania domowe.

Z uwagi na zróżnicowaniem zainteresowań naukowych audytorium, treści programowe mogą ulec niewielkim modyfikacjom.

4. Efekty uczenia się

Rodzaj efektu	Opis efektu uczenia się	Odniesienie do efektów uczenia się w SD PW	Sposób weryfikacji efektów uczenia*
Wiedza			
W01	Absolwent zna podstawowe zagadnienia związane z programowaniem w językach wysokiego poziomu oraz świadomie interpretuje kiedy należy je stosować przy rozwiązywaniu własnych problemów naukowo-badawczych	SD_W3	kolokwium
Umiejętności			
U01	Absolwent potrafi zastosować zdobyte umiejętności w celu rozwiązania postawionego problemu naukowo-badawczego (lub jego wsparcia)	SD_U1	kolokwium prace domowe
U02	Absolwent potrafi przełożyć metodologię postawionego problemu na kod języka programowania wysokiego poziomu, napisać odpowiedni program oraz z czasem go utrzymywać i optymalizować	SD_U2	kolokwium prace domowe
Kompetencje społeczne			
K01	Absolwent jest gotów do krytycznej oceny proponowanych przez siebie rozwiązań, w tym własnego wkładu w rozwój swojej dyscypliny	SD_K1	kolokwium prace domowe

* dozwolone sposoby weryfikacji efektów uczenia się: egzamin; egzamin ustny; kolokwium pisemne; kolokwium ustne; ocena projektu; ocena sprawozdania; ocena raportu; ocena prezentacji; ocena aktywności w trakcie zajęć; prace domowe; test

5. Kryteria oceny

Aby uzyskać ocenę pozytywną należy uzyskać 50% punktów możliwych do zdobycia na kolokwium zaliczeniowym oraz z prac domowych. Każde dodatkowe 10 pkt. % skutkuje podniesieniem oceny, zgodnie ze skalą: 50-60% - 3,0; 61-70% - 3,5; 71-80% - 4,0; 81-90% - 4,5%; 91-100% - 5,0.

Dopuszczalną formą zaliczenia jest dodatkowo przygotowanie projektu programu prezentującego zastosowanie zdobytych umiejętności w obszarze badań doktoranta.

6. Literatura

Literatura podstawowa:

[1] Dokumentacja biblioteki standardowej Python i dodatkowych modułów (<https://docs.python.org/3/>).

Literatura uzupełniająca:

[1] Materiały oraz zasoby internetowe udostępnione przez prowadzącego.

7. Nakład pracy doktoranta niezbędny do osiągnięcia efektów uczenia się**

Lp.	Opis	Liczba godzin
1	godziny kontaktowe z nauczycielem akademickim wynikające z planu	30

2	Godziny kontaktowe z nauczycielem akademickim w ramach konsultacji, egzaminów, sprawdzianów itp.	5
3	Godziny pracy samodzielnej doktoranta w ramach przygotowania do zajęć oraz opracowania sprawozdań, projektów, prezentacji, raportów, prac domowych	30
4	godziny pracy samodzielnej doktoranta w ramach przygotowania do egzaminu, sprawdzianu, zaliczenia	20
Sumaryczny nakład pracy doktoranta		85
Liczba punktów ECTS		3

** 1 ECTS pracy = 25-30 godzin nakładu pracy doktoranta (np. 2 ECTS = 60 godzin; 4 ECTS = 110 godzin)