



KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu

Programowanie w języku Python [S1ETI2>PwJP]

Przedmiot

Kierunek studiów

Edukacja techniczno-informatyczna

Rok/Semestr

1/2

Studia w zakresie (specjalność)

–

Profil studiów

ogólnoakademicki

Poziom studiów

pierwszego stopnia

Język oferowanego przedmiotu

polski

Forma studiów

stacjonarne

Wymagalność

obligatoryjny

Liczba godzin

Wykład

0

Laboratorium

30

Inne

0

Ćwiczenia

0

Projekty/seminaria

0

Liczba punktów ECTS

3,00

Koordynatorzy

dr inż. Szymon Maćkowiak

szymon.mackowiak@put.poznan.pl

Wykładowcy

dr inż. Szymon Maćkowiak

szymon.mackowiak@put.poznan.pl

Wymagania wstępne

Wiedza z matematyki i informatyki na poziomie osiągniętym po 1 semestrze studiów na kierunku edukacja techniczno-informatyczna. Podstawowa umiejętność obsługi komputera z systemem Windows. Aktywna postawa podczas rozwiązywania problemów, zrozumienie konieczności poszerzania swoich kompetencji.

Cel przedmiotu

1. Zapoznanie studentów z podstawami języka Python umożliwiające tworzenie programów numerycznych rozwiązujących zagadnienia pojawiające się w fizyce i technice. 2. Zademonstrowanie użyteczności komputera jako narzędzia wspomagającego analizę i rozwiązywanie prostych problemów fizycznych i technicznych. 3. Wykształcenie u studenta umiejętności samodzielnego wykorzystywania komputera do analizy prostych problemów w dziedzinie informatyki i techniki.

Przedmiotowe efekty uczenia się

Wiedza:

1. Student zna aparat matematyczny niezbędny do opisu i analizy podstawowych zagadnień inżynierii materiałowej, mechaniki technicznej i informatyki, obejmujący podstawy rachunku różniczkowego i całkowego, algebrę liniową, geometrię analityczną, statystykę i metody numeryczne.

2. Student ma wiedzę w zakresie programowania proceduralnego, obiektowego, sztucznej inteligencji, baz danych oraz grafiki komputerowej.

Umiejętności:

1. Student potrafi pozyskiwać informacje z literatury, baz danych oraz innych źródeł, integrować je, dokonywać ich interpretacji oraz wyciągać wnioski, formułować i uzasadniać opinie, a także przygotować i przedstawić prezentację ustną i udokumentowane opracowanie dotyczące zagadnień z zakresu kierunku kształcenia w języku polskim i angielskim.
2. Student potrafi wykorzystać nabytą wiedzę matematyczną do opisu procesów, tworzenia modeli oraz innych działań w zakresie inżynierii materiałowej, mechaniki, konstrukcji maszyn, elektrotechniki, elektroniki i informatyki.
3. Student potrafi pracować indywidualnie i w zespole, zarządzać swoim czasem oraz komunikować się z wykorzystaniem zaawansowanych technik informacyjno-komunikacyjnych właściwych do realizacji zadań typowych dla działalności inżynierskiej.
4. Student potrafi identyfikować problem techniczny, określić jego stopień złożoności, a także zaproponować schemat jego analizy i rozwiązania.
5. Student ma umiejętność samokształcenia się i rozumie potrzebę uczenia się przez całe życie zawodowe.

Kompetencje społeczne:

1. Student rozumie potrzebę ciągłego doksztalcania się (np. poprzez uczestnictwo w kursach i studiach podyplomowych) w celu podnoszenia kompetencji zawodowych i społecznych oraz potrzebę myślenia i działania w sposób przedsiębiorczy i innowacyjny.
2. Student potrafi pracować nad wyznaczonym zadaniem samodzielnie oraz współpracować w zespole przyjmując w nim różne role, wykazuje się w tej pracy profesjonalizmem i odpowiedzialnością za podejmowane decyzje.
3. Student postępuje zgodnie z zasadami etyki zawodowej, jest odpowiedzialny za rzetelność wyników swoich prac.

Metody weryfikacji efektów uczenia się i kryteria oceny

Efekty uczenia się przedstawione wyżej weryfikowane są w następujący sposób:

1. 90 minutowe kolokwium (polega na rozwiązaniu krótkich zadań za pomocą poznanego środowiska programistycznego) (próg zaliczeniowy 50.1%).
2. Ocena ciągła na każdym laboratorium: prezentacja programu komputerowego (premiowanie aktywności).

Treści programowe

Podstawy programowania w języku Python, w tym środowiska programistyczne, obsługa wejścia i wyjścia, typy danych oraz struktury danych. Instrukcje warunkowe, pętle, rekurencję i algorytmy sortowania. Dodatkowo zasady programowania obiektowego, obejmujące klasy, dziedziczenie, polimorfizm i enkapsulację; praca z modułami oraz zastosowanie bibliotek Pythona w analizie danych naukowych.

Tematyka zajęć

A. Podstawy

1. Wprowadzenie do dystrybucji języka Python i wybranych edytorów kodu (Idle, Spyder, Jupyter, PyCharm, VS Code).
2. Obsługa standardowego wejścia wyjścia w języku Python.
3. Pojęcie zmiennej oraz podstawowe typy danych.
4. Przegląd złożonych struktur danych (listy, krotki, słowniki, łańcuchy znaków, zbiory)
5. Instrukcje warunkowe i obsługa wyjątków.
6. Pętle "for" i "while" w języku Python.
7. Rekurencja porównanie implementacji funkcji silnia(n) w ujęciu iteracyjnym i rekurencyjnym.
8. Algorytmy sortowania.
9. Tworzenie i obsługa funkcji.

B. Programowanie obiektowe.

1. Pojęcie klasy, konstruktory, atrybuty, metody.

2. Enkapsulacja.
3. Polimorfizm.
4. Dziedziczenie i wielodziedziczenie.
5. Praca z pakietami i modułami języka Python.
6. Tworzenie własnych modułów.
7. Zastosowanie wybranych bibliotek języka Python w analizie danych naukowych.

Metody dydaktyczne

1. Prezentacja multimedialna - wprowadzenie do realizowanego tematu ćwiczeń laboratoryjnych (komputerowych).
2. Ćwiczenia laboratoryjne (komputerowe) - wykonywanie specjalistycznych programów, praca indywidualna, dyskusja.
3. Indywidualna praca studenta, dyskusja.

Literatura

Podstawowa:

1. Amit Saha, "Matematyka w Pythonie", Helion, Gliwice 2015.
2. Al Sweigart, "Programowanie w Pythonie dla średnio zaawansowanych", Helion, Gliwice 2021.

Uzupełniająca:

literatura wskazana przez prowadzącego zajęcia

Bilans nakładu pracy przeciętnego studenta

	Godzin	ECTS
Łączny nakład pracy	75	3,00
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	30	1,50
Praca własna studenta (studia literaturowe, przygotowanie do zajęć laboratoryjnych/ćwiczeń, przygotowanie do kolokwium/egzaminu, wykonanie projektu)	45	1,50