

Nazwa w języku polskim: Programowanie w języku Python dla inżynierów

Nazwa w jęz. angielskim: Programming in Python for engineers

Dane dotyczące zajęć:

Information on course:

Jednostka oferująca: Instytut Fizyki - CND // dr hab. inż. Grzegorz Adamiec , prof. PŚ

Course offered by: Institute of Physics – CSE // dr hab. inż. Grzegorz Adamiec , prof. PŚ

Język wykładowy:
polski
Language:
Polish
Strona WWW: Course homepage:
Skrócony opis:
W ramach przedmiotu studenci zapoznają się z najpopularniejszymi bibliotekami języka Python pozwalającymi na proste tworzenie bardzo zaawansowanych i wydajnych aplikacji. Celem przedmiotu jest zapoznanie Studentów z pojęciami dotyczącymi reprezentacji danych, projektowania i analizy algorytmów oraz z podstawami poprawnego programowania w języku Python w zastosowaniach inżynierskich. Studenci mają zostać przygotowani do samodzielnej realizacji prostych algorytmów zarówno numerycznych jak i tekstowych. Zestaw omawianych zagadnień stanowi podstawę do dalszej nauki programowania.
Short description:
During the course, students will become acquainted with the most popular Python libraries that allow for the simple creation of very advanced and efficient applications. The aim of the course is to familiarize students with concepts related to data representation, design and analysis of algorithms, and the basics of correct programming in Python for engineering applications. Students are to be prepared for the independent implementation of simple numerical and textual algorithms. The set of topics discussed forms a basis for further learning in programming.
Opis:
Treści programowe Wykład 1. Pojęcie zmiennej, typy int, float, str, bool, complex 2. Operacje na łańcuchach 3. Operacje arytmetyczne 4. Operatory logiczne 5. Instrukcja warunkowa if elif else 6. Typ złożony: lista 7. Pętle for i while .. else 8. Instrukcje break i continue 9. Liczby binarne i szesnastkowe (inaczej heksadecymalne) 10. Pojęcie algorytmu 11. Wartość None 12. Kolekcje: list, tuple, dict (listy, tuple, słowniki) 13. Operacje na kolekcjach, iteracja po elementach kolekcji 14. Paradygmaty programowania. Pojęcie funkcji 15. Operacje na plikach, wczytywanie plików tekstowych 16. Moduł os 17. Zasięg zmiennych 18. Wyjątki. Try except 19. Docstrings 20. Funkcje ze zmienną liczbą parametrów 21. Typ set (zbiory)

22. Pojęcie klasy i obiektu
23. Metody i atrybuty
24. Metoda magiczna `__init__`
25. Założenia paradygmatu obiektowego
26. Dziedziczenie
27. Metoda `__super__()`
28. Listy składane (listy comprehensions)
29. Operator trójargumentowy (ternary operator)
30. Identyfikacja obiektów. Operator `is`. Funkcja `id`
31. Funkcje jako pełnoprawne obiekty w Pythonie
32. Wyrażenie `lambda`
33. Podstawy dekoratorów
34. Zmienna liczba argumentów funkcji `*args` i `**kwargs`
35. Menadżer kontekstu dla pliku (`with ... as...:`)
36. Moduł `numpy`
37. Wyrażenia regularne i moduł `re`
38. Moduł `numpy`
39. `Matplotlib`, wykresy 3D, kreślenie wektorów
40. Wykresy animowane
41. Iteratory
42. Generatory

Wykład:

- **stacjonarne: 30 h**
- **niestacjonarne: 18 h**

Liczba punktów ECTS: 2

Description:

Lecture

1. Concept of a variable, types `int`, `float`, `str`, `bool`, `complex`
2. String operations
3. Arithmetic operations
4. Logical operators
5. Conditional statement `if` `elif` `else`
6. Composite type: `list`
7. Loops `for` and `while .. else`
8. `Break` and `continue` statements
9. Binary and hexadecimal numbers
10. Concept of an algorithm
11. The `None` value
12. Collections: `list`, `tuple`, `dict`
13. Operations on collections, iterating over collection elements
14. Programming paradigms. Concept of a function
15. File operations, reading text files
16. `os` module
17. Scope of variables
18. Exceptions. `Try except`
19. Docstrings
20. Functions with a variable number of parameters
21. `Set` type
22. Concept of class and object
23. Methods and attributes
24. Magic method `init`
25. Principles of the object-oriented paradigm
26. Inheritance
27. Method `super()`
28. List comprehensions
29. Ternary operator
30. Object identity. The `is` operator. `id` function
31. Functions as first-class objects in Python
32. Lambda expressions

33. Fundamentals of decorators
34. Variable number of function arguments *args and **kwargs
35. Context manager for files (with ... as...:)
36. numpy module
37. Regular expressions and re module
38. numpy module
39. Matplotlib, 3D plots, vector plotting
40. Animated charts
41. Iterators
42. Generators

Lecture:

- **full-time studies: 30 h**
- **part-time studies: 18 h**

Number of ECTS credits: 2

Literatura:

- 1) Eric Matthes, Python. Instrukcje dla programisty. Helion
- 2) Christian Hill, Learning Scientific Programming with Python, 2nd edition, Cambridge University Press
- 3) Podręczniki z podstaw Pythona (wersja conajmniej 3.10)

Bibliography:

- 1) Eric Matthes, Python. Python. Crash course
- 2) Christian Hill, Learning Scientific Programming with Python, 2nd edition, Cambridge University Press
- 3) Any Python textbook (version at least 3.10)

Efekty uczenia się:

Wiedza

Student zna i rozumie:

- podstawowe dylematy współczesnej cywilizacji,
- Student zna i rozumie wybrane zagadnienia z zakresu informatyki praktycznej

Umiejętności

Student potrafi:

- samodzielnie planować i realizować proces własnego uczenia się.

Kompetencje i postawa społeczna

Student jest gotów do:

- uznawania znaczenia wiedzy w rozwiązywaniu problemów poznawczych i praktycznych

Learning outcomes:

Knowledge:

The student knows and understands:

basic dilemmas of contemporary civilization,

- The student knows and understands selected topics in the field of practical computer science.

Skills:

The student is able to:

- independently plan and implement the process of their own learning.

Competencies and social attitude:

The student is prepared to:

- recognize the importance of knowledge in solving cognitive and practical problems

Metody i kryteria oceniania:

Wykład

Zaliczenie w formie sprawdzianu pisemnego. Kryterium zaliczenia: 50%

Assessment methods and assessment criteria:

Lecture

Passing the course in the form of a written test. Criterion for passing the course 50%

**Przynależność do grup przedmiotów w cyklach:
Element of course groups in various terms:**

Opis grupy przedmiotów Course group description	Cykl pocz. First term	Cykl kon. Last term
przedmioty obieralne studia stacjonarne stopień studiów – dowolny kierunek studiów – dowolny, semestr dowolny elective courses full-time degree - any field of study - any semester - any		