COURSE DESCRIPTION CARD - SYLLABUS

Course name
Computer science [S1MiTPM1>INF]

Course

Field of study
Materials and technologies for automotive industry

Year/Semester
1/1

Area of study (specialization)
–

Profile of study
general academic

Level of study
first-cycle

Course offered in
Polish

Form of study
full-time

Requirements
compulsory

Number of hours

Lecture
15

Laboratory classes
15

Other
0

Tutorials
0

Projects/seminars
0

Number of credit points

3,00

Coordinators

dr inż. Szymon Maćkowiak
szymon.mackowiak@put.poznan.pl

Lecturers

Prerequisites

Wiedza z matematyki i informatyki na poziomie osiągniętym po szkole średniej o profilu ogólnym. Podstawowa umiejętność obsługi komputera z systemem Windows. Aktywna postawa podczas rozwiązywania problemów, zrozumienie konieczności poszerzania swoich kompetencji.

Course objective

1. Introducing students to the basics of programming in Python, enabling the creation of programs for data processing and analysis, as well as solving selected problems in the field of materials engineering. 2. Demonstrating the usefulness of the computer as a tool for assisting in the analysis and resolution of simple technical problems. 3. Developing students' ability to independently use a computer to analyze simple problems in the field of engineering.

Course-related learning outcomes

Knowledge:
1. A student who has completed the course has basic knowledge of computer science, enabling them to describe computer systems, use selected graphic and office software, as well as programming languages and computer-aided engineering systems in the automotive industry. They know how to formulate and

analyze problems.

Skills:
1. A student who has completed the course is able to communicate using various techniques in a professional environment and other contexts. To this end, they can select and apply IT systems and quality management systems. They are able to use information and communication technologies appropriate for carrying out tasks typical of engineering activities, including organizing teamwork.
2. A student who has completed the course has the ability to self-educate.
3. A student who has completed the course is able to plan and conduct measurements, experiments, and simulations, interpret the results obtained, and draw conclusions. They are capable of using computer-aided tools to solve technical tasks and can apply material testing methods and operate specialized measuring equipment.
4. A student who has completed the course can apply analytical, simulation, and experimental methods to formulate and solve engineering tasks. They are able to define problems and use mathematical methods to analyze technical issues. In formulating and solving engineering tasks, they can recognize their systemic and non-technical aspects.

Social competences:
1. A student who has completed the course understands the need for lifelong learning and is able to inspire and organize the learning process of others.
2. A student who has completed the course is able to cooperate, think and act in an entrepreneurial manner, and work in a group, assuming different roles within it.
3. A student who has completed the course is able to appropriately set priorities to accomplish a task defined by themselves or others.

Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

1. A written exam covering the content delivered during the lectures (passing threshold: 50.1%).
2. A practical exam (on the computer) or the completion of a project task based on the content covered in the laboratory classes (passing threshold: 50.1%).

Programme content

1. Python interpreters and programming environments.
2. Basic data types and structures in Python.
3. Control flow statements in Python.
4. Selected Python libraries.
5. Data analysis using Python.

Course topics

1. Introduction to the Python distribution and selected code editors (Idle, Spyder, Jupyter, PyCharm, VS Code).
2. Handling standard input and output in Python.
3. The concept of variables and basic data types.
4. Overview of complex data structures (lists, tuples, dictionaries, strings, sets).
5. Conditional statements and exception handling.
6. "For" and "while" loops in Python.
7. Recursion: comparison of factorial(n) implementation using iterative and recursive approaches.
8. Overview of selected classic algorithms - sorting algorithms.
9. The concept of computational complexity and Big O notation.
10. Creating and handling functions.
11. Using Python libraries (numpy, pandas, matplotlib).
12. Reading data from files and writing data to files.
13. Basics of scientific data analysis.
14. Creating plots using the matplotlib and seaborn libraries.
15. Basic image processing operations using the scikit-image library.

Teaching methods

1. Multimedia presentation - introduction to the topic of the laboratory (computer-based) exercises.
2. Laboratory exercises (computer-based) - development of specialized programs, individual work, discussion.
3. Individual student work, discussion.

Bibliography

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

Additional:
1. Al Sweigart, "Automatyzacja nudnych zadań z Pythonem", Helion, Gwilice 2021.
2. Maciej M. Sysło, "Algorytmy", Helion, Gliwice 2016.
3. Brett Slatkin, "Efektywny Python", Helion, Gliwice 2016.

Breakdown of average student's workload

	Hours	ECTS
Total workload	75	3,00
Classes requiring direct contact with the teacher	30	1,50
Student's own work (literature studies, preparation for laboratory classes/ tutorials, preparation for tests/exam, project preparation)	45	1,50