COURSE DESCRIPTION CARD - SYLLABUS

Course name
Programming Languages [S2MiBP1>JP]

Course

Field of study
Mechanical and Automotive Engineering

Year/Semester
1/1

Area of study (specialization)
Motor Vehicles

Profile of study
general academic

Level of study
second-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

2,00

Coordinators

dr inż. Przemysław Grzymisławski
przemyslaw.grzymislawski@put.poznan.pl

Lecturers

Prerequisites

KNOWLEDGE: The student has elementary knowledge of the basics of computer science, ie computer architecture, types of variables, general knowledge of high-level languages used in computer programming. SKILLS: The student is able to use the terms in the description of programming languages and is able to solve specific problems that arise when writing programs. SOCIAL COMPETENCES: The student is able to work in a group, assuming different roles in it, is able to define priorities important in solving the tasks set before him and shows independence in solving problems, acquiring and improving acquired knowledge and skills.

Course objective

The aim of the course is to provide students with information on selected programming languages (Python, C++), definitions and concepts. Students acquire knowledge and skills in creating computer programs.

Course-related learning outcomes

Knowledge:
Has extended knowledge in the field of computer science, concerning computer programming and engineering calculation programs in the field of computer simulation of physical systems.

He knows the modern engineering methods of computer graphics and the theoretical basis of engineering calculations using the finite element method.
Has a general knowledge of the types of research and methods of testing working machines with the use of modern measurement techniques and data acquisition.

Skills:
Can use a popular numerical system to program a simple system simulation task with a small number of degrees of freedom.
Can write a simple computer program with the use of modern RAD environments in a language known to him for the optimization calculations of structures using learned elementary numerical methods.
Can conduct a debate.

Social competences:
He is ready to critically assess his knowledge and received content.
Is willing to think and act in an entrepreneurial manner.
Is ready to fulfill professional roles responsibly, taking into account changing social needs, including:
- developing the professional achievements,
- maintaining the ethos of the profession,
- observing and developing the rules of professional ethics and acting towards the observance of these rules.

Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

Learning outcomes presented above are verified as follows:
The knowledge acquired during the lecture is verified by 45-minute tests carried out during the last lecture. The test consists of 6-10 questions with different scores. Passing threshold: 50% of points. Final issues, on the basis of which the questions are developed, will be sent to students by e-mail using the university"s e-mail system.
The skills acquired during the laboratory classes are verified on the basis of 4 short programs written by the student in the Python language. The pass mark is 3 correctly functioning
programs.

Programme content

Building computer programs. Comparison of C++ and Python language structures. Overview of declarations of constants, variables and variable types. Arithmetic operators. Functions - function value and parameters, passing arguments by value and reference. Expressions - assignment, data comparison, priority and assignment. Branches and loops. Arrays and structures. Standard Python libraries.
Basic concepts related to numerical calculations: iteration, interpolation, approximation, extrapolation, numerical integration, solving ordinary differential equations. Algorithms: calculating the square root, finding zeros of functions - Newton"s method, secant and bisection, numerical integration with Richardson extrapolation, solving ordinary differential equations using the Euler method and midpoint. Procedures for these algorithms in Python.

Course topics

none

Teaching methods

1. Lecture: multimedia presentation, illustrated with examples given on the board.
2. Laboratory: examples given on the blackboard and carrying out the tasks given by the teacher.

Bibliography

Basic
1. Michael Dawson, Python dla każdego - podstawy programowania, Wydawnictwo Helion, Wydanie III.
2. https://docs.python.org/3/

Breakdown of average student's workload

	Hours	ECTS
Total workload	50	2,00
Classes requiring direct contact with the teacher	30	1,00
Student's own work (literature studies, preparation for laboratory classes/ tutorials, preparation for tests/exam, project preparation)	20	1,00