COURSE DESCRIPTION CARD - SYLLABUS

Course name
Basics of programming - Scripting languages [N1MiBM2>JSPP]

Course

Field of study
Mechanical Engineering

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
part-time

Requirements
elective

Number of hours

Lecture
8

Laboratory classes
16

Other
0

Tutorials
0

Projects/seminars
0

Number of credit points

4,00

Coordinators

Lecturers

Prerequisites

Basic computer skills

Course objective

The aim of the course is to introduce students to the basics of programming in high-level languages (using a selected language). During the classes, fundamental concepts and structures encountered in programming are presented (variables, data types, statements, user communication handling, subroutines), as well as the construction of programs utilizing these structures (combined with analysis of problems in an algorithmic context) and verification of their correctness

Course-related learning outcomes

Knowledge:
The student is familiar with the typology of programming languages and has knowledge of their basic applications. They are aware of the tools that assist programmers in their work. They know the sources of information covering issues related to application creation, programming, and usage. Additionally, they are capable of formulating algorithms and programming them using at least one popular tool.

Skills:
The student can create simple applications and design a good user interface for them. They are capable

of assessing, at a basic level, the usefulness of routine methods and IT tools and can select and apply appropriate methods and tools for typical IT tasks.

Social competences:
The student understands the need for lifelong learning and can inspire and organize the learning process for others. They are aware of the importance and understanding of non-technical aspects and consequences of engineering activities.

Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

Lecture: Passing is based on a written exam consisting of 30 multiple-choice questions on a 0/1 scale. Passing requires a minimum score of 51%.
Laboratory: Passing is based on a practical exam during lab sessions worth 60 points and tasks completed during labs worth 40 points. The final grade is determined according to the following scale: 100 points based on the adopted grading system - 3.0 from 41 points, 3.5 from 56 points, 4.0 from 71 points, 4.5 from 81 points, 5.0 from 91 points.

Programme content

In the "Scripting Languages" lecture, one programming language chosen by the instructor (either C or Python) will be presented. The lecture is organized to be accessible for individuals with very basic programming skills (especially in the initial phase of the lecture), as well as useful for advanced individuals. In the first part, the semantics and syntax of the language, basic instructions and operators, and detailed principles of loop operation will be presented. The later part will discuss methods of building simple and advanced data structures along with examples.
Laboratory sessions - practical application of knowledge acquired during the lecture, implementation, and testing - will cover the following topics:
1. Compilation model.
2. Operators and their precedence.
3. Preprocessor directives, macros.
4. Arrays.
5. Pointers and their relationship with arrays.
6. Passing values to functions.
7. Function pointers.
8. Structures.
9. Dynamic creation of data structures using pointers.
10. File operations.
11. Discussion of the standard C language library (functions related to basic string operations, mathematical functions, conversions).
12. Discussion of safe and unsafe functions (buffer overflow issues).
13. Writing large programs and the use of GNU make, autoconf, and configure.
14. Error search strategies and the use of debugger-type tools, valgrind.

Course topics

none

Teaching methods

Lecture: The lecture will consist of a multimedia presentation, illustrated with examples and videos, analyzing problems and visually representing them on the board, followed by discussion and problem analysis.
Laboratory: The laboratory sessions will be conducted through computer work, during which programs illustrating the discussed topics will be written together with the instructor. In later stages of the laboratory sessions, students will individually write programs to accomplish specific tasks that have been previously discussed and demonstrated by the instructor.

Bibliography

Basic:

1. C. Albing, J.P. Vossen, C. Newham : Bash. Receptury. Helion, Gliwice.
2. H. Schwichtenberg : Windows PowerShell. Podstawy. Helion, Gliwice.
3. P. Norton, A. Samuel : Python. Od podstaw. Helion, Gliwice.
4. M. Lutz : Python. Wprowadzenie. Helion, Gliwice.

Additional:
Źródła internetowe: np. https://docs.python.org/3/library/index.html

Breakdown of average student's workload

	Hours	ECTS
Total workload	100	4,00
Classes requiring direct contact with the teacher	24	1,00
Student's own work (literature studies, preparation for laboratory classes/ tutorials, preparation for tests/exam, project preparation)	76	3,00