COURSE DESCRIPTION CARD - SYLLABUS

Course name
Scripting languages [S1Cybez1>PS]

Course

Field of study
Cybersecurity

Year/Semester
1/2

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
24

Laboratory classes
24

Other
0

Tutorials
0

Projects/seminars
0

Number of credit points

3,00

Coordinators

dr inż. Paweł Sroka
pawel.sroka@put.poznan.pl

Lecturers

Prerequisites

A student starting this course should have extended knowledge in mathematics in physics, including the basics of linear algebra and probability theory. Moreover, he/she should have basic skills in programming and on using the PC computer. It is suggested also to know the selected telecommunication protocols, including HTTP. A studetd should understand the need to extend his/her knowledge and skills according to the programme of the studies.

Course objective

The main aim of this course is to teach students the basic knowledge and skills in programming using selected script languages. Moreover, methods and tools for processing and presenting the collected data using script languages are taught, as well as the implementation of solution to the selected mathematical problems.

Course-related learning outcomes

Knowledge:
1. Has a solid knowledge of construction and implementation of computer programmes, algorithms and data structures using the learned scripting languages.

Skills:
1. Is able to prepare simple applications using learned scripting languages with the purpose of analysis and solving of problems related to the field of study.
2. Is able to store, present and process the collected data in numerical or graphical form using learned scripting languages.
3. Is able to implement and use selected mathematical models and algorithms to solve problems with the aid of learned scripting languages.

Social competences:
1. Understands the need to further extend the knowlegde about scripting languages; is aware that the knowledge and skills in this area evolve quickly.

Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

The knowledge acquired in the lectures is verified in form of a written credit test. The written credit test comprises 6-10 test or open-ended questions that are graded (with points) differently.
The abilities acquired during the laboratories are verified with 4-8 practical exercises, relying on implementation of specific programs or algorithms using selected script languages, according to the provided instructions. Each exercise will be graded with a specific number points depending on the completeness of the provided report and the complexity of the task. The final grade will be also influenced by the overal engagement and dedication to work by the student, and, eventually, realization of additional homeworks.
In each form of the course assessment, the grade depends on the number of points the student earns relative to the maximum number of required points. Earning at least 50% of the possible points is a prerequisite for passing. The relationship between the grade and the number of points is defined by the Study Regulations. Additionally, the course completion rules and the exact passing thresholds will be communicated to students at the beginning of the semester through the university’s electronic systems and during the first class meeting (in each form of classes).

Programme content

This course presents and teaches the theoretical and practical aspects of using of selected scripting languages to solve selected problems in mathematics, computer science and telecommunications. The course topics comprise the basics of implementation of computer programs using scripting languages, including the details of their syntax, as well as methods of generating, storing and processing of data.

Course topics

Lectures will cover the following topics:
1. Introduction to scripting languages - basic syntax of selected scripting languages, data types, operators and precedence, loops, branching, functions, array operations.
2. Reading, storing and processing of data using selected scripting languages. Graphical presentation of results.
3. Implementation of selected mathematical problems using scripting languages - elements of probability theory, statistics and linear algebra.
4. Use of scripting languages for the purpose of basic Internet services provisioning.
5. Basics of use of script languages to work with databases.
Laboratory exercises will consist of the following topics:
1. Basic elements and syntax of selected scripting languages: Podstawowe elementy i składnia wybranych języków skryptowych (np. Python, JavaScript): data types, operators and precedence, loops, branching, functions, array operations.
2. Data manipulation using selected scripting languages - reading and storing data in different formats, graphical presentation of results, selected libraries in scripting languages.
3. Implementation of selected aspects of probability theory - generating pseudorandom numbers, calculating moments, elements of statistics.
4. Implementation of selected solutions of linear algebra and matrix operations.
5. Use of scripting languages for the purpose of basic Internet services provisioning and database manipulation.

Teaching methods

1. Lectures: multimedia presentation illustrated with examples shown using a computer or given on the board; eventually tutorial-style classes.
2. Laboratories: realization of exercises using computers according to instructions provided by the teacher - practical exercises, eventually supplemented with multimedia presentation.

Bibliography

Basic:
Mark Lutz, Python. Wprowadzenie. Wydanie V, Helion 2020
Boschetti A., Massaron L., Python: podstawy nauki o danych, Helion, 2017
Suehring S., JavaScript: krok po kroku, RM, 2009

Additional:
McKinney W., Python for data analysis, O'Reilly, 2013
Robert Johansson, Matematyczny Python. Obliczenia naukowe i analiza danych z użyciem NumPy, SciPy i Matplotlib, Helion 2021
Krajka A., Python: podstawy języka i aplikacje internetowe, Wyd. Uniwersytetu Marii Curie-Skłodowskiej, 2011
www.w3schools.com

Breakdown of average student's workload

	Hours	ECTS
Total workload	88	3,00
Classes requiring direct contact with the teacher	48	1,50
Student's own work (literature studies, preparation for laboratory classes/ tutorials, preparation for tests/exam, project preparation)	40	1,50