

# Scripting languages - course description

General information	
Course name	Scripting languages
Course ID	11.3-WE-INFN-ScriptLang-Er
Faculty	<a href="#">Faculty of Computer Science, Electrical Engineering and Automatics</a>
Field of study	Computer Science
Education profile	academic
Level of studies	Second-cycle Erasmus programme
Beginning semester	winter term 2022/2023

Course information	
Semester	2
ECTS credits to win	5
Course type	optional
Teaching language	english
Author of syllabus	<ul style="list-style-type: none"><li>• dr inż. Grzegorz Bazydło</li><li>• dr hab. inż. Remigiusz Wiśniewski, prof. UZ</li><li>• dr inż. Iwona Grobelna</li></ul>

Classes forms					
The class form	Hours per semester (full-time)	Hours per week (full-time)	Hours per semester (part-time)	Hours per week (part-time)	Form of assignment
Lecture	15	1	-	-	Credit with grade
Laboratory	30	2	-	-	Credit with grade
Project	15	1	-	-	Credit with grade

## Aim of the course

- Familiarize students with scripting languages for their practical use.
- Shaping basic skills for improving daily computer tasks (at home, work) using scripting languages.

## Prerequisites

Basic knowledge of operating systems (Windows, Linux), computer networks and websites design.

## Scope

- Virtualization, virtual machines, run the system environment from another operating system.
- Command line and scripting languages of Linux and/or Windows operating systems (e.g., *Bash*, *CMD*, *PowerShell*).
- Tips and tricks of process management and automation tasks in Linux and Windows systems, practical use of the command line, and scripting languages (e.g., *Bash*, *CMD*) to improve the efficiency of the usage of an operating system.
- Introduction to advanced scripting languages (e.g., *Perl*, *Python*, *VBA*).
- Practical application of scripts in daily computer tasks (e.g., data backup, fast file conversion between different formats, etc.), *lifehacking*.

## Teaching methods

**Lecture:** conventional lecture, discussion.

**Laboratory:** laboratory exercises, work in groups.

**Project:** project method, discussion.

## Learning outcomes and methods of theirs verification

Outcome description	Outcome symbols	Methods of verification	The class form
Is able to select the right tools and scripting solutions depending on the design requirements.		<ul style="list-style-type: none"><li>• a project</li></ul>	<ul style="list-style-type: none"><li>• Project</li></ul>
Is able to apply scripting languages to solve scientific and engineering problems.		<ul style="list-style-type: none"><li>• a project</li><li>• a quiz</li><li>• an ongoing monitoring during classes</li></ul>	<ul style="list-style-type: none"><li>• Laboratory</li><li>• Project</li></ul>
Knows current trends and the need to streamline daily computer tasks through using scripting languages.		<ul style="list-style-type: none"><li>• a discussion</li><li>• an evaluation test</li></ul>	<ul style="list-style-type: none"><li>• Lecture</li></ul>
Understands the need of using scripting languages.		<ul style="list-style-type: none"><li>• a quiz</li><li>• an ongoing monitoring during classes</li></ul>	<ul style="list-style-type: none"><li>• Laboratory</li></ul>

Outcome description	Outcome symbols	Methods of verification	The class form
Is able to apply scripting techniques to perform tasks that are a part of a larger project or IT system.		<ul style="list-style-type: none"> <li>• a project</li> <li>• an ongoing monitoring during classes</li> </ul>	<ul style="list-style-type: none"> <li>• Laboratory</li> <li>• Project</li> </ul>

## Assignment conditions

**Lecture** – the passing condition is to obtain a positive mark from the final test.

**Laboratory** – the passing condition is to obtain positive marks from all laboratory exercises to be planned during the semester.

**Project** – the passing condition is to obtain a positive mark from all projects conducted during the semester.

**Final mark components:** lecture 30% + laboratory 40% + project 30%.

## Recommended reading

1. C. Albing, JP Vossen, C. Newham, bash Cookbook: Solutions and Examples for bash Users, O'Reilly Media, 2007.
2. Chromatic, D. Conway, C. Poe, Perl Hacks: Tips & Tools for Programming, Debugging, and Surviving, O'Reilly Media, 2006.
3. P. Barry, Head First Python: A Brain-Friendly Guide, 2nd Edition, O'Reilly Media, 2016.
4. A. Pash, G. Trapani, Lifehacker: The Guide to Working Smarter, Faster, and Better, Third Edition, Wiley, 2011.
5. A. Pash, Gina Trapani, Lifehacker: The Guide to Working Smarter, Faster, and Better, part 2, Wiley, 2011.

## Further reading

1. M. Lutz, Learning Python, 5th Edition, O'Reilly Media, 2013.
2. J. Forcier, P. Bissex, W. Chun, Python Web Development with Django (Developer's Library), Addison-Wesley Professional, 2008.
3. M. Dawson, Python Programming for the Absolute Beginner, 3rd Edition, Course Technology, 2010.
4. E. T. Freeman, E. Robson, Head First HTML5 Programming: Building Web Apps with JavaScript, O'Reilly Media, 2011

## Notes

Modified by dr inż. Grzegorz Bazydło (last modification: 21-04-2022 09:57)

Generated automatically from SyllabUZ computer system