COURSE SYLLABUS:

Field of study: **Computer Science**	Profile: **practical**	Level of studies: **first-cycle studies**	Year of study:	Language of instruction: **English**
Academic year: **2023/2024** **Semester (winter/summer): summer**	Form (full-time / part-time): **full-time**	Status (compulsory / elective): **compulsory**	Prerequisites: **In accordance with the rules of studying at the Faculty of Design, par. 11. sec. 6-8. (Rules for qualification for the next year of study).**	
Number of hours in the study plan: **30**	Number of ECTS points: **9**	Total number of student work hours (1 ECTS=25-30h of work): **270**		

Course coordinator

Name and surname, academic degree:
dr inż. Maksymilian Bujok

Course components, form of classes, number of hours, instructor

Tyoe of classes	*Number of hours*	*Instructor*
Laboratories	*24*	*Mgr inż. Krzysztof Ropiak*
Laboratories	*24*	*Dr Ghafarian Mabhoot Toktam*

REFERENCE TO FIELD-SPECIFIC LEARNING OUTCOMES (symbols of field-specific outcomes)	LEARNING OUTCOMES FOR THE COURSE	TASKS TO COMPLETE
INF1_W01	Has basic knowledge of the work environment of a programmer, especially Linux systems. Understands the issues and benefits related to application versioning and virtualization, using containerization as an example.	laboratory exercises
INF1_U05 INF1_U15 INF1_U14	Can use basic shell commands, write and run a simple Bash script. Can create and manage a project repository. Can create a configuration file to create and run a container (with a low level of complexity).	laboratory exercises
INF1_K01 INF1_K02	The knowledge gained allows us to search for sources for further development, The knowledge and skills acquired allow for a critical approach to the content received, as well as verification of sources.	laboratory exercises activity in class

Tasks to complete - characteristics	Implementation deadline	Scope of material to be prepared by students	Number of points to be scored	
			Max	Min
laboratory exercises	During every class"	*All material from every class will be covered*	100	51

NOTE!

The student has the right to one retake of each task not passed on the first attempt. There is no possibility to improve passed tasks.

Attendance at classes is mandatory. A maximum of 2 unexcused absences are allowed. In case of an excused additional absence, the student will receive an additional task from the instructor.

CONVERTING POINTS TO GRADE:

PASSING THE COURSE requires the student to obtain **more than 50% of the points possible to obtain for each task** described in the table above. The final grade, provided that >50% of points are obtained for each task, results from the sum of points obtained. The grading scale is as follows:

GRADING SCALE:
54 – 60 – means 3 (satisfactory)
61 – 70 – means 3+ (satisfactory plus)
71 – 80 – means 4 (good)
81 – 90 – means 4+ (good plus)
91 – 100 – means 5 (very good)

COURSE CONTENT	
TYPE of classes: laboratory	
Topics covered in classes:	The subject is designed to familiarise students with the basic tools of the job. During the course, the student will become familiar with Linux and the basic Bash shell commands that will allow them to navigate freely in this type of system, install packages, read and edit files, etc. Further on, students will understand the basics of a version control system using Git as an example.

Required Literature:

Classic Shell Scripting. Hidden Commands that Unlock the Power of Unix

S. Chacon, Pro Git – professional version contro

Supplementary Literature:

Kane, S. P., Matthias, K. Docker. PDocker: Up & Running, 3rd Edition O'Reilly Media, Inc.