

**AGH**AGH UNIVERSITY OF SCIENCE  
AND TECHNOLOGY

Code: UBPJO-235 Module name: Scripting languages

Academic year: 2017/2018 Semester: Spring, Fall ECTS credits: 4

Programme: Physics and Applied Computer Science

Course homepage: Lecture language: English

Responsible teacher: dr hab. inż. Bołd Tomasz (tomasz.bold@fis.agh.edu.pl)

Academic teachers: dr hab. inż. Bołd Tomasz (tomasz.bold@fis.agh.edu.pl)

### Module summary

During the course, 4 scripting languages will be introduced (shell, python, ROOT, JavaScript). They will illustrate advantages of scripting approach.

### Description of learning outcomes for module

MLO code	Student after module completion has the knowledge/ knows how to/is able to	Method of learning outcomes verification (form of completion)
Social competence		
M_K001	The student is capable of constructive discussion on subjects covered in the course.	Participation in a discussion
M_K002	The student is able to discuss solutions publicly.	Participation in a discussion
Skills		
M_U001	The student is able to adjust technique to the problem at hand during laboratory classes	Execution of laboratory classes
Knowledge		
M_W001	The student can write scripts in four presented languages.	Project

### FLO matrix in relation to forms of classes

MLO code	Student after module completion has the knowledge/ knows how to/is able to	Form of classes										
		Lectures	Auditorium classes	Laboratory classes	Project classes	Conversation seminar	Seminar classes	Practical classes	Fieldwork classes	Workshops	Others	E-learning
Social competence												
M_K001	The student is capable of constructive discussion on subjects covered in the course.	+	-	-	-	-	-	-	-	-	-	-
M_K002	The student is able to discuss solutions publicly.	+	-	-	-	-	-	-	-	-	-	-
Skills												
M_U001	The student is able to adjust technique to the problem at hand during laboratory classes	-	-	+	-	-	-	-	-	-	-	-
Knowledge												
M_W001	The student can write scripts in four presented languages.	+	-	+	+	-	-	-	-	-	-	-

## Module content

### Lectures

#### Introduction to scripting

Historical perspective, the proliferation of scripting languages today. Compiled program lifetime vs script lifetime. Scripts validation.

#### Shell scripting

Text processing, structuring textual data. BASH scripts. Overview of the associated utilities.

#### Python

Brief introduction: philosophy, variables and memory model, control statements. Using objects. Testing: testable code, simple tests, and testing systems. Python as integration language.

#### ROOT

Cling interpreter. Philosophy of the ROOT. Memory model in ROOT, current context. Types of ROOT scripts. Testing.

#### Java Script

Applicability and environment. Language review. Basic of jQuery. Testing.

### Laboratory classes

#### Shell scripting 1

Text processing in shell scripts: Connecting programs for filtering and transforming

textual data.

#### Tasks automation

BASH scripts. Processing experimental data.

#### Elementary python scripts

Acquiring basic language skills, simple scripts, input, and output.

#### Text processing in Python

Processing files. Data extraction and transformation. Data persistency.

#### ROOT

Basic usage. Data visualization. Data storage.

#### ROOT

Cling scripts. Compiled scripts.

#### JavaScript

Interaction with the interpreter. Loading and using jQuery. Interaction with the XML document.

#### JavaScript

External script. External data loading. Result storage. Testing scripts.

#### Projects presentation

Students present their projects (5-10 minutes).

### **Project classes**

#### Shell script

The first project is a shell script. The scope and exact requirements will be established individually for each student. Well seen are scripts proposed by students.

#### Python script

Again, the scope and exact requirements will be established for each student. A rewrite of the previous shell script is possible. Most welcome are projects proposed by students.

#### ROOT or JS script

The third project is a script written in ROOT or JS. It is a free choice of every student to pick the technology. Again, the most welcome are scripts that are meant to be useful for students.

### **Method of calculating the final grade**

The final grade is calculated from an average of the points obtained for each project. Each project is graded from 0-6 points. Additional 4 points on top of that are given for its presentation. Projects not delivered in a timely manner are graded with 0 points.

### **Prerequisites and additional requirements**

Reasonable unix/linux experience. A bit of c++ knowledge. Knowledge of basics of HTML documents.

### **Recommended literature and teaching resources**

Recommended literature and teaching resources not specified

### **Scientific publications of module course instructors related to the topic of**

**the module**

Additional scientific publications not specified

**Additional information**

None

**Student workload (ECTS credits balance)**

Student activity form	Student workload
Participation in lectures	10 h
Participation in laboratory classes	20 h
Completion of a project	40 h
Preparation for classes	30 h
Summary student workload	100 h
Module ECTS credits	4 ECTS