

## MODULE/ COURSE FORM

### A. general information

To be completed by Course Team	Module name : <b>INTRODUCTION TO SCRIPT LANGUAGES</b>					Module code:		
	Course name: <b>Introduction to script languages</b>					Course code:		
	Faculty: <b>INSTITUTE OF APPLIED INFORMATICS</b>							
	Field of study: <b>INFORMATICS</b>				Level of education: <b>first</b>			
	Mode of study : <b>Full-time</b>		Learning profile: <b>Practical</b>		Speciality:.			
	Year/ semester: <b>2/4</b>		Module/ course status:. obligatory			Module/ course language: <b>Polish/English</b>		
	Type of classes		lecture	lessons	lab	project	Tutorial	other (please specify)
	Course load		<b>15</b>		<b>15</b>	<b>15</b>		
	Module/ course objectives		<p>The aim of the course is to familiarize the student with the basics of programming in scripting languages such as JavaScript and Python. Teaching the creation of simple object-oriented applications using selected standard and additional libraries. Transfer of knowledge in the use of basic types and constructs of language and functions (JavaScript / Python) or objects (Python). Teaching how to create your own functions or classes. Teaching how to create objects (Python) and call a programmer function (JavaScript).</p>					
Entry requirements		<p>Basics of programming in C. Knowledge of the basics of object-oriented programming: the principle of class creation, inheritance, encapsulation, etc.</p>						
<b>LEARNING OUTCOME</b>								
Nr	LEARNING OUTCOME DESCRIPTION					Learning outcome reference		
	Knowledge:							
1.	He knows the basic functions of the JavaScript language					K_W07, K_W11		
2.	He knows the basic classes of objects from Python standard libraries					K_W07, K_W11		
3.	He knows the basic rules of using the JavaScript framework					K_W07, K_W11		
4.	He knows the basic principles of using an object library in Python					K_W07, K_W11		
	Skills:							
5.	He writes and runs simple applications using standard JavaScript functions					K_U01,K_U16,K_U17, K_U19		
6.	He writes simple user interface applications in JavaScript					K_U01,K_U16,K_U17, K_U19		
7.	He can create a simple class in Python					K_U01,K_U16,K_U17, K_U19		

8.	He can call selected Python objects	K_U01,K_U16,K_U17, K_U19
9.	Works independently, looking for solutions to the problems encountered in the documentation and on internet forums	K_K01
<b>Assessment method</b>		Learning outcome number
Reports		5,6,7,8,9
Short written test on laboratory		5,6,7,8
Colloquium (lecture)		1,2,3,4
<b>STUDENT WORKLOAD</b>		
	Number of hours	
	In all	including practical
Participation in lectures	15	12
Independent study of lecture topics	10	8
Participation in tutorials, labs, projects and seminars	30	30
Independent preparation for tutorials*	10	10
Preparation of projects/essays/etc. *	20	20
Preparation/ independent study for exams	10	10
Participation during consultation hours	5	
Other	2	
<b>TOTAL student workload in hours</b>	102	90
<b>Number of ECTS credit per course unit</b>	<b>4 ECTS</b>	
Number of ECTS credit associated with practical classes	<b>3,5 ECTS</b>	
Number of ECTS for classes that require direct participation of professors	<b>2 ECTS</b>	

## B. details information

To be completed by Course Team	Module name : <b>INTRODUCTION TO SCRIPTING LANGUAGES</b>					Module code:	
	Course name: <b>Introduction to scripting languages</b>					Course code:	
	Faculty: <b>Institute of Applied Informatics</b>						
	Field of study: <b>INFORMATICS</b>			Level of education: <b>first</b>			
	Mode of study : <b>Full-time</b>		Learning profile: <b>Practical</b>		Speciality::		
	Year/ semester: <b>2/4</b>		Module/ course status:: <b>obligatory</b>			Module/ course language: <b>Polish/English</b>	
	Type of classes	lecture	lessons	lab	project	Tuto rial	other (please specify)
	Course load	<b>15</b>		<b>15</b>	<b>15</b>		
	Module/ course coordinator		<b>dr inż. Robert Smyk</b>				
Lecturer		<b>dr inż. Robert Smyk</b>					
<b>CURRICULUM CONTENTS</b>							
<b>Lecture</b>							
<ul style="list-style-type: none"> <li>- basics of functional programming in javascript (using standard language functions),</li> <li>- elementary basics of creating applications in the node. js,</li> <li>- the basics of creating interfaces using jQuery, and an elementary discussion of the selected framework of application skeleton on the user's side (eg angular.js),</li> <li>- general basics of object-oriented programming in Python v2 / v3: basic data types, class creation rules, object instance, overview of standard libraries,</li> <li>- basics of programming web applications in Python using the chosen framework (eg Django).</li> </ul> <p>Including content related to practical professional preparation: [80%]</p>							
<b>Lessons</b>							
<ul style="list-style-type: none"> <li>- creating simple scripts containing programming functions in JavaScript to support the user interface,</li> <li>- creating simple user interface scripts using the jQuery JavaScript library,</li> <li>- creating elementary scripts in the node.js environment,</li> <li>- creating a simple application based on the application framework JavaScript framework,</li> <li>- creating elementary scripts in Python using basic data types and selected objects of standard libraries</li> <li>- creating and using Python classes in Python,</li> <li>- creating a simple web application based on a Python framework.</li> </ul>							
<b>Project (other)</b>							
- development of a frontend project using JavaScript for a sample web application (50%)							

- implementation of a simple Python application / service on a given topic, having a user interface or other type of interface containing programmers 'classes and using external programmers' libraries (50%)

Basic literature	<ol style="list-style-type: none"><li>1. Documentation of JavaScript (<a href="https://developer.mozilla.org/pl/docs/Web/JavaScript">https://developer.mozilla.org/pl/docs/Web/JavaScript</a>)</li><li>2. Documentation of Python (<a href="https://www.python.org/doc/">https://www.python.org/doc/</a>)</li><li>3. Jon Duckett, JavaScript i jQuery. Interaktywne strony WWW dla każdego, <a href="http://helion.pl">helion.pl</a></li><li>4. Mark Lutz, Python wprowadzenie. Wydanie IV, helion.pl</li></ol>
Additional literature	Libraries manuals
Teaching methods	Lecture with presentation, introducing basic concepts and problems (slides), and presenting examples of solutions ("live" programming) Laboratory exercises, where students perform basic lecture examples, then additional tasks with increasing complexity; throughout the semester, they develop the project, supplementing it gradually with new elements
Form and terms of an exam	Laboratory reports, laboratory test, final lecture (written test)