SYLLABUS OF A MODULE

Polish name of a module	**Paradygmaty programowania**
English name of a module	**Programming paradigms**
ISCED classification - Code	0613
ISCED classification - Field of study	*Software and applications development and analysis*
Languages of instruction	*English*
Level of qualification:	*1 – BSc (EQF 6)*
Number of ECTS credit points	*5*
Examination:	*EW – exam written*

Number of hours per semester:

Lecture	Exercises	Laboratory	Seminar	E-learning	Project
15	0	30	0	0	0

MODULE DESCRIPTION

MODULE OBJECTIVES

O1. Obtaining knowledge in the basic programming paradigms

O2. Obtaining knowledge in the selecting programming language to solve posed problem

PRELIMINARY REQUIREMENTS FOR KNOWLEDGE, SKILLS AND OTHER COMPETENCES

1. Knowledge of mathematics.
2. Basics of computer skills.
3. Basic knowledge in some high level programming language.
4. Ability to correctly interpret and present their own actions.

LEARNING OUTCOMES

LO 1 – Student has knowledge of the basic programming paradigms

LO 2 – Student has the skills to create basic applications in the learned programming paradigms

LO 3 – Student has competence in creating basic applications in the learned programming paradigms

MODULE CONTENT

Type of classes – lecture	**Number of hours**
Lec 1 - Introduction to Programming Paradigms	**1**
Lec 2 - Introduction to functional programming	**1**
Lec 3 - Types and pattern matching	**1**
Lec 4 - Higher order functions	**1**
Lec 5,6 - Functional data structures	**2**
Lec 7,8 - Patterns in functional programming	**2**
Lec 9-11 - Multiparadigms languages.	**3**
Lec 12,13 - Patterns in multiparadigms languages.	**2**
Lec 14,15 - Asynchronous programming in multiparadigms language.	**2**
Sum	**15**
Type of classes– laboratory.	**Number of hours**
Lab 1 - Introduction to Programming Paradigms	**2**
Lab 2 - Introduction to functional programming	**2**
Lab 3 - Types and pattern matching	**2**
Lab 4 - Higher order functions	**2**
Lab 5,6 - Functional data structures	**4**
Lab 7,8 - Patterns in functional programming	**4**
Lab 9-11 - Multiparadigms languages	**6**
Lab 12,13 - Patterns in multiparadigms languages	**4**
Lab 14,15 - Asynchronous programming in multiparadigms language	**4**
Sum	**30**

TEACHING TOOLS

1. - multimedial presentations for lectures	
2. - instructions for laboratories	
3. - wide range of programming tools	
4. - workplaces for students equipped with workstations	

WAYS OF ASSESSMENT (F – FORMATIVE, S – SUMMATIVE

F1. - assessment of preparation for laboratory exercises
F2. - assessment of the ability to apply the acquired knowledge while doing the exercises
F3. - evaluation of reports on the implementation of exercises covered by the curriculum
F4. - assessment of activity during classes
S1. - assessment of the ability to solve the problems posed and the manner of presentation obtained results - pass mark *
S2. - assessment of mastery of the teaching material being the subject of the lecture - exam

*) in order to receive a credit for the module, the student is obliged to attain a passing grade in all laboratory classes as well as in achievement tests.

STUDENT’S WORKLOAD

L.p.	Forms of activity	Average number of hours required for realization of activity
1. Contact hours with teacher		
1.1	Lectures	15
1.2	Tutorials	
1.3	Laboratory	40
1.4	Seminar	
1.5	Project	
1.6	Consulting teacher during their duty hours	3
1.7	Examination	5
	Total number of contact hours with teacher:	63
2. Student’s individual work		
2.1	Preparation for tutorials and tests	
2.2	Preparation for laboratory exercises, writing reports on laboratories	22
2.3	Preparation of project	
2.4	Preparation for final lecture assessment	20
2.5	Preparation for examination	
2.6	Individual study of literature	20
	Total number of hours of student’s individual work:	62
	Overall student’s workload:	125
Overall number of ECTS credits for the module		5 ECTS
Number of ECTS points that student receives in classes requiring teacher’s supervision:		2,5 ECTS
Number of **ECTS** credits acquired during practical classes including laboratory exercises and projects:		2,2 ECTS

BASIC AND SUPPLEMENTARY RESOURCE MATERIALS

1. C. Smith “Programming F# 3.0”, OReilly 2012
2. T. Liu “F# for C# Developers”, Microsoft Press 2013
3. E. Buonanno, “Functional Programming in C#. How to write better C# code.”, 2017
4. L. Atencio, „Functional Programming in JavaScript. How to improve your JavaScript programs using functional techniques”, Manning, 2016

MODULE COORDINATOR (NAME, SURNAME, E-MAIL ADDRESS)

Lukasz Bartczuk (lukasz.bartczuk@pcz.pl)