SYLLABUS OF A MODULE

Polish name of a module	**Programowanie współbieżne i rozproszone**
English name of a module	**Concurrent and distributed programming**
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	*6*
Examination	*A - assignment*
Available in semester	*A – Autumn only*

Number of hours per semester:

Lecture	Tutorial	Laboratory	Seminar	Project	Others
30	0	30	0	0	0

MODULE DESCRIPTION

Module objectives

O1. Knowledge on parallel architectures and models, standards and techniques for concurrent, distributed, and parallel programming.

O2. Practical skills in concurrent / distributed / parallel programming, running and debugging the application for different types of concurrent and distributed architectures.

PRELIMINARY REQUIREMENTS FOR KNOWLEDGE, SKILLS AND OTHER COMPETENCES

1. Basic knowledge of computer architecture and operating systems.
2. Basic knowledge of the theory of algorithms and data structures.
3. Ability of C++ and Java programming.
4. Ability to use different sources of information and technical documentation.
5. Ability to work independently and in a group.
6. Ability to correctly interpret and present their own activities.

LEARNING OUTCOMES

LO 1 – has knowledge about the elements of concurrent programming, parallel and distributed computing.

LO 2 – able to implement concurrent, parallel and distributed applications.

LO 3 – competence to design distributed applications.

MODULE CONTENT

Lectures	**Number of hours**
Lect. 1 - Introduction to parallel processing and parallel system architecture	**2**
Lect. 2 - Distributed processing / parallel clusters and grid systems	**2**
Lect. 3 - Examples of application parallel and distributed computing	**2**
Lect. 4 - Constructing parallel and distributed algorithms – part 1	**2**
Lect. 5 - Constructing parallel and distributed algorithms – part 2	**2**
Lect. 6 - Parallel and distributed programming models	**2**
Lect. 7 - Introduction to languages environments of parallel and distributed programming	**2**
Lect. 8 - Parallel programming using MPI standard – part 1	**2**
Lect. 9 - Parallel programming using MPI standard – part 2 2	**2**
Lect. 10 - Basic concepts of concurrent programming 2	**2**
Lect. 11 - Representative examples of concurrent programming problems and their solutions using MPI	**2**
Lect. 12 - Multithreaded Programming – part 1	**2**
Lect. 13 - Multithreaded Programming – part 2	**2**
Lect. 14 - Representative examples of concurrent programming problems and their solutions	**2**
Lect. 15 - The use of RMI environment for building distributed applications	**2**
Sum	**30**
Laboratories	**Number of hours**
Lab. 1 - Principles of creating and running parallel programs in C/C++ for MPI environment, running simple programs.	**2**
Lab. 2 - Point-to-point communication.	**2**
Lab. 3 - Creating programs using the master-worker model.	**2**
Lab. 4 - Examples of more advanced parallel programs in MPI environment.	**2**
Lab. 5 - Analysis and optimizing the performance of parallel programs in MPI environment – part 1	**2**
Lab. 6 - Analysis and optimizing the performance of parallel programs in MPI environment – part 2	**2**
Lab. 7 - Written test/project.	**2**
Lab. 8 - Introduction to multithreaded programming.	**2**
Lab. 9 - Synchronization of threads to access shared resources.	**2**
Lab. 10 - Examples of concurrent and parallel programs – part 1	**2**
Lab. 11 - Examples of concurrent and parallel programs – part 2	**2**
Lab. 12 - Examples of concurrent and parallel programs – part 3	**2**
Lab. 13 - Profilining of concurrent applications	**2**
Lab. 14 - Analysis and optimizing the performance of parallel programs	**2**
Lab. 15 - Written test/project.	**2**
Sum	**30**

TEACHING TOOLS

1. Multimedial presentations for lectures
2. Instructions for laboratories
3. Workplaces for students equipped with workstations

WAYS OF ASSESSMENT (F – FORMATIVE, S – SUMMATIVE

F1. Assessment of preparation for laboratory
S1. Assessment of the ability to solve the problems posed and the manner of presentation obtained results - pass mark *

*) 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	30
1.2	Tutorials	
1.3	Laboratory	30
1.4	Seminar	
1.5	Project	
1.6	Examination	
	Total number of contact hours with teacher:	60
2.	**Student's individual work**	
2.1	Preparation for tutorials and tests	
2.2	Preparation for laboratory exercises, writing reports on laboratories	24
2.3	Preparation of project	
2.4	Preparation for final lecture assessment	12
2.5	Preparation for examination	
2.6	Individual study of literature	54
	Total numer of hours of student's individual work:	90
	Overall student's workload:	150
Overall number of ECTS credits for the module		6
Number of ECTS points that student receives in classes requiring teacher's supervision:		2.52
Number of **ECTS** credits acquired during practical classes including laboratory exercises and projects:		2.16

BASIC AND SUPPLEMENTARY RESOURCE MATERIALS

1. Andrews, G.R.: „Foundations of Multithreaded, Paralel and Distributed Programming". Addison Wesley, 2002.

2. Grama, A., Gupta, A., Kumar, V., Karypis, G.: „Introduction to parallel computing (second edition)". Addison-Wesley, 2003.

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

Lukasz Szustak, lszustak@icis.pcz.pl

Krzysztof Rojek, krojek@icis.pcz.pl