



### MODULE DESCRIPTION

Module code	
Module name	<b>Programowanie współbieżne</b>
Module name in English	<b>Concurrent Programming</b>
Valid from academic year	<b>2012/2013</b>

### MODULE PLACEMENT IN THE SYLLABUS

Subject	<b>Computer Science</b>
Level of education	<b>1<sup>st</sup> degree</b> <i>(1st degree / 2nd degree)</i>
Studies profile	<b>General</b> <i>(general / practical)</i>
Form and method of conducting classes	<b>Full-time</b> <i>(full-time / part-time)</i>
Specialisation	
Unit conducting the module	<b>The Department of Computer Science</b>
Module co-ordinator	<b>Paweł Paduch, PhD, Eng.</b>
Approved by:	

### MODULE OVERVIEW

Type of subject/group of subjects	<b>Major</b> <i>(basic / major / specialist subject / conjoint / other HES)</i>
Module status	<b>Compulsory</b> <i>(compulsory / non-compulsory)</i>
Language of conducting classes	<b>Polish</b>
Module placement in the syllabus - semester	<b>3<sup>rd</sup> semester</b>
Subject realisation in the academic year	<b>Winter semester</b> <i>(winter / summer)</i>
Initial requirements	<b>Programming in the C Language 2; Operating Systems</b> <i>(module codes / module names)</i>
Examination	<b>No</b> <i>(yes / no)</i>
Number of ECTS credit points	<b>5</b>

Method of conducting classes	Lecture	Classes	Laboratory	Project	Other
Per semester	<b>30</b>		<b>30</b>		

### TEACHING RESULTS AND THE METHODS OF ASSESSING TEACHING RESULTS



Projekt współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego

<b>Module target</b>	The aim of the module is to familiarise students with: basic concurrent programming issues, the methods of building processes and themes, the methods of communication and synchronisation as regards multi-theme applications, the selected concurrent algorithms, concurrent programming on the .NET platform, and concurrent programming in Java.
----------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Effect symbol	Teaching results	Teaching methods (l/c/v/p/other)	Reference to subject effects	Reference to effects of a field of study
W_01	Knowledge of the issues concerning concurrent programming.	l	KW_09	T1A_W02 T1A_W03 T1A_W04
W_02	Knowledge of synchronisation methods and communication between the processes and themes.	l	KW_09	T1A_W02 T1A_W03 T1A_W04
W_03	Learning the principles of analysing concurrent programmes in terms of effectiveness and safety.	l	KW_09	T1A_W02 T1A_W03 T1A_W04
W_04	Knowledge of the selected concurrent algorithms.	l	KW_09	T1A_W02 T1A_W03 T1A_W04
U_01	The ability of concurrent programming.	l	KU_15	T1A_U05 T1A_U07 T1A_U08 T1A_U09 T1A_U10 T1A_U14 T1A_U15 T1A_U16

**Teaching contents:**

**Teaching contents as regards lectures**

Lecture number	Teaching contents	Reference to teaching results for a module
1	Introduction to concurrent programming.	W_01,W_03
2	Semaphores and buffers.	W_02
3	Processes, signals, and links.	W_02
4	Monitors and conditional variables.	W_02
5	IPC message queues.	W_02
6	IPC semaphores and shared memory.	W_02
7	Themes and mutexes.	W_02
8	Posix message queues.	W_02
9	Posix signals, semaphores, and shared memory.	W_02
10	Concurrent algorithms, concurrent programming in the .NET programming.	W_02, W_04
11	Processes and themes in Java.	W_02
12	Synchronised blocks and methods, immutable objects.	W_02
13	Utilising collections by numerous themes.	W_02
14	Synchronizers.	W_02
15	Task cancelling.	W_02



### Teaching contents as regards laboratory classes

Laboratory class number	Teaching contents	Reference to teaching results for a module
1	The fundamentals of the Linux system.	U_01
2	Process.	U_01
3	PIPE links.	U_01
4	FIFO links.	U_01
5	IPC_MSG message queues.	U_01
6	Semaphores.	U_01
7	Shared memory.	U_01
8	Themes and mutexes.	U_01
9	Posix message queues.	U_01
10	Concurrent programming in C#.	U_01
11	Processes and themes in Java.	U_01
12	Synchronised blocks and methods, immutable objects.	U_01
13	Utilising collections by many themes.	U_01
14	Synchronizers.	U_01
15	Task cancelling.	U_01

### The methods of assessing teaching results

Effect symbol	Methods of assessing teaching results <i>(assessment method, including skills – reference to a particular project, laboratory assignments, etc.)</i>
W_01	Obtaining a credit for the lectures.
W_02	Obtaining a credit for the lectures.
W_03	Obtaining a credit for the lectures.
W_04	Obtaining a credit for the lectures.
U_01	Obtaining a credit for laboratory classes on the basis of a student's involvement during laboratory classes and written tests.

### STUDENT'S INPUT

ECTS credit points		
	Type of student's activity	Student's workload
1	Participation in lectures	30
2	Participation in classes	
3	Participation in laboratories	30
4	Participation in tutorials (2-3 times per semester)	3
5	Participation in project classes	
6	Project tutorials	
7	Participation in an examination	
8		
9	<b>Number of hours requiring a lecturer's assistance</b>	<b>63</b> <i>(sum)</i>
10	<b>Number of ECTS credit points which are allocated for assisted work</b> <i>(1 ECTS credit point=25-30 hours)</i>	<b>2</b>
11	Unassisted study of lecture subjects	<b>15</b>
12	Unassisted preparation for classes	



Projekt współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego

13	Unassisted preparation for tests	<b>15</b>
14	Unassisted preparation for laboratories	<b>30</b>
15	Preparing reports	<b>15</b>
16	Preparing for a final laboratory test	
17	Preparing a project or documentation	
18	Preparing for an examination	
19	Preparing questionnaires	
20	<b>Number of hours of a student's unassisted work</b>	<b>75</b> <i>(sum)</i>
21	<b>Number of ECTS credit points which a student receives for unassisted work</b> <i>(1 ECTS credit point=25-30 hours)</i>	<b>3</b>
22	<b>Total number of hours of a student's work</b>	<b>138</b>
23	<b>ECTS credit points per module</b> <i>1 ECTS credit point=25-30 hours</i>	<b>5</b>
24	<b>Work input connected with practical classes</b> <i>Total number of hours connected with practical classes</i>	<b>75</b>
25	<b>Number of ECTS credit points which a student receives for practical classes</b> <i>(1 ECTS credit point=25-30 hours)</i>	<b>3</b>