



MODULE DESCRIPTION

Module code	
Module name	Podstawy Programowania 2
Module name in English	The Fundamentals of Programming 2
Valid from academic year	2012/2013

MODULE PLACEMENT IN THE SYLLABUS

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

MODULE OVERVIEW

Type of subject/group of subjects	Basic <i>(basic / major / specialist subject / conjoint / other HES)</i>
Module status	Compulsory <i>(compulsory / non-compulsory)</i>
Language of conducting classes	Polish
Module placement in the syllabus - semester	2nd semester
Subject realisation in the academic year	Summer semester <i>(winter / summer)</i>
Initial requirements	The Fundamentals of Programming 1 <i>(module codes / module names)</i>
Examination	No <i>(yes / no)</i>
Number of ECTS credit points	5

Method of conducting classes	Lecture	Classes	Laboratory	Project	Other
Per semester	30		30	15	

TEACHING RESULTS AND THE METHODS OF ASSESSING TEACHING RESULTS



Projekt współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego

Module target	The aim of the module is to familiarise students with dynamic memory management, abstract data structures, practical aspects of utilising recurrence in programming as well as with the fundamentals of an object paradigm of creating software.
----------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Effect symbol	Teaching results	Teaching methods (l/c/l/p/other)	Reference to subject effects	Reference to effects of a field of study
W_01	A student knows the issues connected with the indicators and dynamic memory allocation.	l	KW_06, KW_07	T1A_W07
W_02	A student is acquainted with the issues connected with simple abstract data structures such as lists and binary trees.	l	KW_06, KW_07	T1A_W07, T1A_W04
W_03	A student knows the issues connected with practical aspects of applying recurrence technologies in programming.	l	KW_06, KW_07	T1A_W07, T1A_W04
W_04	A student is knowledgeable about the fundamentals of an object model of creating software.	l	KW_11, KW_07	T1A_W07, T1A_W04
U_01	A student can apply a dynamic memory allocation to build abstract data structures.	l/p	K_U12, K_U13	T1A_U09, T1A_U14, T1A_U16
U_02	A student can create programs by applying recurrence.	l/p	K_U12, K_U13, K_U19	T1A_U09, T1A_U14, T1A_U16
U_03	A student can create simple object programs.	l/p	K_U17, K_U12	T1A_U09, T1A_U15, T1A_U16
K_01	The ability of teamwork.	p	K_K03, K_U02	T1A_U02, T1A_K04

Teaching contents:

Teaching contents as regards lectures

Lecture number	Teaching contents	Reference to teaching results for a module
1.	Indicators.	W_01
2.	Stack.	W_01, W_02
3.	Queues.	W_01, W_02
4.	Singly linked linear list.	W_01, W_02
5.	Doubly linked linear list.	W_01, W_02
6.	Doubly linked circular list.	W_01, W_02
7. 8.	Binary Search Trees (BST).	W_01, W_02
9.	Recurrence and the divide and conquer technique.	W_03
10.	Backtracking.	W_03
11.	Quicksort and Heapsort.	W_03
12.	The fundamentals of object-oriented programming.	W_04
13.	Inheritance and composition.	W_04
14.	Polymorphism.	W_04



15.	Dynamic objects.	W_04
-----	------------------	------

Teaching contents as regards laboratory classes

Laboratory class number	Teaching contents	Reference to teaching results for a module
1.	Indicators.	U_01
2.	Stack.	U_01
3.	Queues.	U_01
4.	Singly linked linear list.	U_01
5.	Doubly linked linear list.	U_01
6.	Doubly linked circular list.	U_01
7-8.	Binary Search Trees (BST).	U_01, U_02
9.	Recurrence and the divide and conquer technique.	U_02
10.	Backtracking.	U_01, U_02
11.	Quicksort and Heapsort.	U_02
12.	The fundamentals of object-oriented programming.	U_03
13.	Inheritance and composition.	U_03
14.	Polymorphism.	U_03
15.	Dynamic objects.	U_03

The characteristics of project assignments

Project assignments consist in designing, creating, and providing documentation for a computer program. The subject of the problem solved by the program is unlimited; however, the task requires using programming techniques and a high-level programming language presented during the lectures. Project assignments are completed in teams of two. Students ought to demonstrate their independence in project completion; furthermore, they should suggest a method of checking the correctness of the suggested solution as well as provide documentation for a program source code.

The methods of assessing teaching results

Effect symbol	Methods of assessing teaching results (assessment method, including skills – reference to a particular project, laboratory assignments, etc.)
W_01	A written test
W_02	A written test
W_03	A written test
W_04	A written test
U_01	Laboratory class assignments, a test on laboratory classes, a project assignment
U_02	Laboratory class assignments, a test on laboratory classes, a project assignment
U_03	Laboratory class assignments, a test on laboratory classes, a project assignment
K_01	A project assignment

STUDENT'S INPUT

ECTS credit points		Student's workload
	Type of student's activity	
1	Participation in lectures	30



Projekt współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego

2	Participation in classes	
3	Participation in laboratories	30
4	Participation in tutorials (2-3 times per semester)	2
5	Participation in project classes	
6	Project tutorials	15
7	Participation in an examination	
8		
9	Number of hours requiring a lecturer's assistance	77 <i>(sum)</i>
10	Number of ECTS credit points which are allocated for assisted work <i>(1 ECTS credit point=25-30 hours)</i>	3
11	Unassisted study of lecture subjects	10
12	Unassisted preparation for classes	
13	Unassisted preparation for tests	10
14	Unassisted preparation for laboratories	10
15	Preparing reports	
16	Preparing for a final laboratory test	10
17	Preparing a project or documentation	15
18	Preparing for an examination	
19	Preparing questionnaires	
20	Number of hours of a student's unassisted work	55 <i>(sum)</i>
21	Number of ECTS credit points which a student receives for unassisted work <i>(1 ECTS credit point=25-30 hours)</i>	2
22	Total number of hours of a student's work	132
23	ECTS credit points per module <i>1 ECTS credit point=25-30 hours</i>	5
24	Work input connected with practical classes <i>Total number of hours connected with practical classes</i>	80
25	Number of ECTS credit points which a student receives for practical classes <i>(1 ECTS credit point=25-30 hours)</i>	3