MODULE DESCRIPTION

Module code	
Module name	**Programowanie obiektowe (Java)**
Module name in English	**Object-Oriented Programming (Java)**
Valid from academic year	**2011/12**

MODULE PLACEMENT IN THE SYLLABUS

Subject	**Computer Science**
Level of education	**1st degree** *(1st degree / 2nd degree)*
Studies profile	**General** *(general / practical)*
Form and method of conducting classes	**Full-time** *(full-time / part-time)*
Specialisation	
Unit conducting the module	**The Department of Computer Science**
Module co-ordinator	**Roman Stanisław Deniziak, PhD hab., Eng., Professor of the University**
Approved by:	

MODULE OVERVIEW

Type of subject/group of subjects	**Major** *(basic / major / specialist subject / conjoint / other HES)*
Module status	**Compulsory** *(compulsory / non-compulsory)*
Language of conducting classes	**Polish**
Module placement in the syllabus - semester	**4th semester**
Subject realisation in the academic year	**Summer semester** *(winter / summer)*
Initial requirements	**The Fundamentals of Programming, Programming in the C Language** *(module codes / module names)*
Examination	**No** *(yes / no)*
Number of ECTS credit points	**7**

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

TEACHING RESULTS AND THE METHODS OF ASSESSING TEACHING RESULTS

Module target	The aim of the module is to acquaint students with basic notions and principles concerning object-oriented programming; another aim is to master the ability to program in Java language as regards creating applets and applications; furthermore, students should acquire the ability of team designing and implementing programs using object-oriented technology.

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	Knowledge of notions determining the paradigm of object-oriented programming.	l	K_W11	T1A_W03
W_02	Knowledge of the principles concerning object-oriented programming.	l	K_W11	T1A_W07
W_03	Knowledge of basic constructions of the Java language.	l	K_W11 K_W12	T1A_W04
U_01	The ability of designing programs in object-oriented technology.	l/p	K_U17 K_U21	T1A_U13 T1A_U14 T1A_U15
U_02	The ability to program in the Java language as regards applications and applets.	l/p	K_U17 K_U18	T1A_U16
U_03	The ability of programming and implementing object-oriented programs in programming units.	p	K_U02	T1A_U02
K_01	The ability of working in programming teams.	p	K_K03	T1A_K03 T1A_K04

Teaching contents:

Teaching contents as regards lectures

Lecture number	Teaching contents	Reference to teaching results for a module
1	Introduction to object-oriented programming, the paradigms of object-oriented programming: abstraction, encapsulation, inheritance, and polymorphism.	W_01 W_02
2	Basic constructions of the Java language; class declarations, fields and static methods.	W_03
3	The principles of creating and initiating objects, constructors, and method names overloading.	W_01 W_02 W_03
4	Encapsulation in the Java language: access qualifiers, interface in relation to implementation, packages.	W_01 W_02 W_03
5	Multiple utilisation of implementation: inheritance and composition, inheritance principles in the Java language. Polymorphism.	W_01 W_02 W_03
6	Final classes, methods, and fields. Abstract classes and interfaces. The applications of polymorphism.	W_01 W_03
7	Generalised types, the principles of declaring classes and methods parametrised with types.	W_03
8	Arrays and object collections, the possibilities of lists, maps, queues, and sets.	W_03
9	Exceptions: the principles of exception specification in methods as well as constructors, and exception handling.	W_03
10	In/out streams in the Java language, the principles of handling a file system.	W_03
11	Object serialisation, the methods of serialisation control.	W_03
12	Creating a GUI, the principles of event handling, and applets.	W_03
13	Type identification during execution, reflection mechanism.	W_01 W_03
14	Enumerated type in Java languages: declaration of type and the principles of application.	W_03
15	The practice of object-oriented programming, examples.	W_02

Teaching contents as regards laboratory classes

Laboratory class number	Teaching contents	Reference to teaching results for a module
1	Introduction to object-oriented programming in Java.	U_01 U_02
2	Basic constructions of the Java language; class declarations, fields and static methods.	U_01 U_02
3	Operations, controlling instructions, loops, and conditional instructions.	U_02
4	The principles of creating and initiating objects, constructors.	U_01 U_02
5	Encapsulation in the Java language: access qualifiers, interface in relation to implementation, packages.	U_01 U_02
6	Multiple utilisation of implementation: inheritance and composition.	U_01 U_02
7	Polymorphism, method overloading and overriding. Finite classes, methods, and fields. Abstract classes and interfaces.	U_01 U_02
8	Operations and one- and multi-dimensional arrays.	U_02
9	Generalised types, the principles of declaring classes and methods parametrised with types, enumerated types.	U_01 U_02
10	Object collections, the possibilities of lists, maps, queues, and sets.	U_02
11	Exceptions and exception handling, type identification during execution, and the reflection mechanism.	U_01 U_02
12	In/out streams in the Java language, the principles of handling a file system, and serialisation.	U_02
13	Creating a GUI, the principles of event handling, and applets.	U_02
14	The elements of concurrent programming and exceptions.	U_02
15	Network programming –sockets handling.	U_02

The characteristics of project assignments

The subject matter of programming issues covers creating an application in Java programming language with a GUI and utilising network communication. The project is completed in teams of a few students. The main aspects taken into consideration while assessing the project are as follows:
- correct issue presentation in terms of object-oriented technique
- correct utilisation of object-oriented programming techniques
- effective division of assignments to be completed by team members
- functional correctness, clarity and code documentation

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 test
W_02	A test
W_03	A test
U_01	Obtaining a credit on the basis of results from particular laboratory class assignments as well as an oral answer.
U_02	Obtaining a credit on the basis of results from particular laboratory class assignments as well as an oral answer.
U_03	Obtaining a credit on the basis of a report on the project, a correct division of tasks in the team, and the progress level as regards assignment completion.
K_01	Obtaining a credit on the basis of completion results as regards the project and an oral answer reflecting the level of involvement in teamwork.

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)	**5**
5	Participation in project classes	**30**
6	Project tutorials	**5**
7	Participation in an examination	
8		
9	**Number of hours requiring a lecturer's assistance**	**100** *(sum)*
10	**Number of ECTS credit points which are allocated for assisted work** *(1 ECTS credit point=25-30 hours)*	**4**
11	Unassisted study of lecture subjects	**15**
12	Unassisted preparation for classes	
13	Unassisted preparation for tests	**20**
14	Unassisted preparation for laboratories	**15**
15	Preparing reports	
16	Preparing for a final laboratory test	
17	Preparing a project or documentation	**30**
18	Preparing for an examination	
19	Preparing questionnaires	
20	**Number of hours of a student's unassisted work**	**80** *(sum)*
21	**Number of ECTS credit points which a student receives for unassisted work** *(1 ECTS credit point=25-30 hours)*	**3**
22	**Total number of hours of a student's work**	**180**
23	**ECTS credit points per module** *1 ECTS credit point=25-30 hours*	**7**
24	**Work input connected with practical classes** *Total number of hours connected with practical classes*	**110**
25	**Number of ECTS credit points which a student receives for practical classes** *(1 ECTS credit point=25-30 hours)*	**4**