

1.3. Module/ course form

To be completed by Course Team	Module name : Programming				Module code: M8		
	Course name: Object-Oriented Programming II				Course code:		
	Faculty: Institute of Applied Informatics						
	Field of study: Informatics						
	Mode of study : Full-time		Learning profile: Practical		Mode of study : Full-time		
	Year/ semester: 2/4		Module/ course status: Mandatory		Year/ semester: 3/5		
	Type of classes	lecture	lessons	lab	project	tutorial	other (please specify)
	Course load	30		30			

Module/ course coordinator	dr Joanna Jólkowska
Lecturer	dr Joanna Jólkowska, mgr Rafał Jólkowski
Module/ course objectives	Teaching how to create desktop application using an IDE (Integrated Development Environment) with GUI (Graphical User Interface) designers; Introduction to the problems associated with multiple code threads in the program and mechanisms to prevent them; Teaching how to develop a client program and a server program establishing a connection to one another to communicate over TCP, and to develop Java applications to access to databases.
Entry requirements	A student knows how to code in general Java and to analyze Java code including classes libraries, arrays and collections, exception handling mechanism by a text user interface

LEARNING OUTCOME		
Nr	LEARNING OUTCOME DESCRIPTION	Learning outcome reference
1	knows the basic techniques for creating desktop application, is familiar with the most commonly used controls (components) Swing library	K_W07 K_W11
2	is familiar with the concept of thread and the consequences (benefits and risks) resulting from the use of multiple threads in the program	K_W07 K_W11
3	is familiar with Java JDBC library to connect to databases	K_W07 K_W11
4	knows the socket interface to communicate with other applications	K_W07 K_W11

5	writes simple desktop applications with the event handlers, using the visual code editor to create a graphical user interface	K_U01 K_U16 K_U17 K_U19
6	uses the event dispatch thread (EDT) in Swing applications	K_U01 K_U16 K_U17 K_U19
7	performs basic SQL queries from a Java application	K_U01 K_U16 K_U17 K_U19
8	write applications that communicate with each other in a client-server model	K_U01 K_U16 K_U17 K_U19
9	works creatively, finding solutions in the documentation and online forums	K_K01

CURRICULUM CONTENTS

Lecture

1. Creating multi-threaded programs
 - a. Thread from creation to launch
 - b. Simultaneous access to data - threads synchronization
 - c. The problem with thread locks and thread locks management
 - d. Thread-safe collections
2. Building GUI (Graphical User Interface)
 - a. Basic components of the Swing library; event handling
 - b. The NetBeans Designer in the implementation of a graphical user interface
 - c. Threads in the Swing library
 - d. Alternative graphics libraries
3. Developing database applications with the JDBC (Java Database Connectivity) drivers
4. Establishing a network connection in Java desktop programs
 - a. TCP sockets
 - b. UCP sockets
 - c. URL connections

Tutorial

1. Basic Java elements – repetition
2. Graphical user interface
 - a. Layouts
 - b. Events
 - c. NetBeans designer
 - d. Developing desktop application “e-teacher”
3. Developing database applications
 - a. CRUD operations
 - b. Application “e-teacher”: adding a model layer with database
4. Threads
 - a. Simple threads, synchronization
 - b. Basic collections and thread-safe collections – comparison
 - c. Event dispatching thread
 - d. Background tasks in Swing applications
5. Network connections

- a. Client-server model
- b. TCP and UDP sockets – comparison
- c. URL class
- d. Developing network application

Basic literature	<ol style="list-style-type: none"> 1. Documentation http://docs.oracle.com/javase/7/docs/api/ 2. Cay S. Horstmann, Gary Cornell – Core Java. Volume I – Fundamentals. Eighth edition 3. Cay S. Horstmann, Gary Cornell – Core Java. Volume II – Advanced Features. Eighth edition
Additional literature	

Teaching methods	<ol style="list-style-type: none"> 1) lecture / lecture with multimedia presentation 2) exercises in auditorium with implementation of the project method for practical tasks (live programming) 3) alone realizes the simple tasks and in second part of the course realizes more sophisticated tasks 4) during whole course develops a individual project . The project is developed gradually by application of new learning outcome.. 5) blended-learning 	
Assessment method		Learning outcome number
Short tests at the beginning of lessons or homework		05,06,07,08
Project task		05,06,07,08,09
Course exam		01,02,03,04
Form and terms of an exam	Parts of course evaluation: 50% course exam, 50% laboratory score	

STUDENT WORKLOAD	
	Number of hours
Participation in lectures	30
Independent study of lecture topics	10
Participation in tutorials, labs, projects and seminars	30
Independent preparation for tutorials*	20
Preparation of projects/essays/etc. *	20
Preparation/ independent study for exams	10
Participation during consultation hours	5
Other	2
TOTAL student workload in hours	127
Number of ECTS credit per course unit	5
Number of ECTS credit associated with practical classes	70 2,8 ECTS
Number of ECTS for classes that require direct participation of professors	67 2,7 ECTS