SYLLABUS

1. Data about the program of study

1.1 Institution	The Technical University of Cluj-Napoca
1.2 Faculty	Faculty of Automation and Computer Science
1.3 Department	Computer Science
1.4 Field of study	Computer Science and Information Technology
1.5 Cycle of study	Bachelor of Science
1.6 Program of study/Qualification	Computer science/ Engineer
1.7 Form of education	Full time
1.8 Subject code	18.

2. Data about the subject

2.1 Subject name	*Object Oriented Programming*					
2.2 Course responsible/lecturer	S.l. dr. eng. Marius Joldoş – Marius.Joldos@cs.utcluj.ro					
2.3 Teachers in charge of seminars/ laboratory/ project	Eng. Bondor Alexandru Viorel – alexandru.viorel.bondor@gmail.com Eng. Solovăstru Ioan – nelusolovastru@gmail.com Eng. Baka Aron – aronbaka97@yahoo.com Eng. Bindea Bogdan – bindeabogdan98@gmail.com					
2.4 Year of study	II	2.5 Semester	1	2.6 Type of assessment (E - exam, C - colloquium, V - verification)		E
2.7 Subject category	*DF – fundamentală, DD – în domeniu, DS – de specialitate, DC – complementară*					DD
	DI – Impusă, DOp – opţională, DFac – facultativă					DI

3. Estimated total time

3.1 Number of hours per week	4	of which:	Course	2	Seminars	Laboratory	2	Project
3.2 Number of hours per semester	56	of which:	Course	28	Seminars	Laboratory	28	Project

3.3 Individual study:	
(a) Manual, lecture material and notes, bibliography	25
(b) Supplementary study in the library, online and in the field	17
(c) Preparation for seminars/laboratory works, homework, reports, portfolios, essays	16
(d) Tutoring	6
(e) Exams and tests	5
(f) Other activities:	0

3.4 Total hours of individual study (suma (3.3(a)…3.3(f)))	69
3.5 Total hours per semester (3.2+3.4)	125
3.6 Number of credit points	5

4. Pre-requisites (where appropriate)

4.1 Curriculum	Computer Programming course
4.2 Competence	Use of a procedural programming language such as C

5. Requirements (where appropriate)

5.1. For the course	
5.2. For the applications	

6. Specific competence

6.1 Professional competences	**C2** – Designing hardware, software and communication components (5 credits) **C2.1** – Describing the structure and functioning of computational, communication and software components and systems **C2.2** – Explaining the role, interaction and functioning of hardware, software and communication components **C2.3** – Building the hardware and software components of some computing

	systems using algorithms, design methods, protocols, languages, data structures, and technologies **C2.4** - Metric based evaluation of functional and non-functional characteristics of computing systems **C2.5** - Implementation of hardware, software and communication components
6.2 Cross competences	N/A

7. Discipline objective (as results from the *key competences gained*)

7.1 General objective	To learn a rigorous treatment of object-oriented concepts using Java as an example language
7.2 Specific objectives	- to prepare object-oriented design for small/medium scale problems - to demonstrate the differences between traditional imperative design and object-oriented design - to explain class structures as fundamental, modular building blocks - to understand the role of inheritance, polymorphism, dynamic binding and generic structures in building reusable code - to write small/medium scale Java programs with simple graphical user interface - to use classes written by other programmers when constructing their systems - to be able to design and build simple Graphical User Interfaces (GUI)s.

8. Contents

8.1 Lectures	Hours	Teaching methods	Notes
Concepts and paradigms in OOP. On to Java	2	Lectures, demos and discussions	Uses a video-projector
Control structures in Java.	2		
Classes and Objects. Arrays	2		
Packages. Inheritance and polymorphism.	2		
Java Interfaces. OO Application Development	2		
UML Object and Class Diagrams. Assertions.	2		
Testing. Debugging. Java Errors and Exceptions	2		
Java Collections. Generic Programming.	2		
Introduction to Java I/O	2		
Event handling in Java. Introduction to Java Graphics	2		
Graphical User Interfaces (I)	2		
Introduction to Threads	2		
Graphical User Interfaces (II)	2		
Review	2		

Bibliography

1. Bruce Eckel, Thinking in Java, Third Edition, Prentice Hall PTR, 2002 (downloadable for free from the Web).
2. Paul & Harvey Deitel, Java. How to Program (Early Objects), Tenth Edition, Prentice Hall, 2015
3. David J. Barnes & Michael Kölling, Objects First with Java. A Practical Introduction using BlueJ, Fifth Edition, Prentice Hall / Pearson Education, 2012
4. Oracle Java Tutorials (freely downloadable from the Web)
5. Schmuller Joseph, SAMS teach yourself UML in 24 hours, 2004

8.2 Applications – Seminars/Laboratory/Project	Hours	Teaching methods	Notes
Using BlueJ IDE	2	Tutoring, discussions, and assisted program development	PCs equipped with Java SDK and IDEs (BlueJ, Eclipse, Netbeans)
Primitive Types and Simple IO in Java	2		
Variables and Expressions in Java	2		
Flow Control and Simple Classes in Java	2		
Classes, Objects and Arrays	2		
Java Inheritance	2		
Java Interfaces	2		
Laboratory test 1	2		
Java Exception Handling.	2		

Collections	2
Testing OOP programs	2
GUIs. Event Handling	2
GUIs. Keyboard and Mouse Handling	2
Laboratory test 2	2

Bibliography
1. Course Moodle site available at: https://moodle.cs.utcluj.ro/

Se vor preciza, după caz: tematica seminariilor, lucrările de laborator, tematica și etapele proiectului.

9. Bridging course contents with the expectations of the representatives of the community, professional associations and employers in the field

The contents of the course is in accordance with the ACM Computer Science Curricula recommendations Java programming language is the most widely used language.

10. Evaluation

Activity type	Assessment criteria	Assessment methods	Weight in the final grade
Course	Ability to solve problems using the object orientated paradigm	For **on-site**: Three in-class tests (T) + Final Written exam (W) For **on-line**: exam using Moodle	**On-site:** 60% = 50% W + 10% T **On-line** 60%
Seminar			
Laboratory	Quality of laboratory applications and evaluation of the laboratory tests	Analysis and evaluation of the solved assignments (for both on-site and online)	40%
Project			

Minimum standard of performance: Grade calculus: 40% laboratory + 60% exams and tests Conditions for participating in the final exam: Laboratory ≥ 5 Conditions for promotion: grade ≥ 5

Date of filling in:	**Titulari**	**Titlu Prenume NUME**	**Semnătura**
	Course	S.l. dr. eng. Marius Joldoş	
	Applications	Eng. Alexandru Viorel Bondor	
		Eng. Ioan Solovăstru	
		Eng. Baka Aron	
		Eng. Bindea Bogdan	

Date of approval in the department	Head of department Prof.dr.ing. Rodica Potolea
Date of approval in the Faculty Council	Dean Prof.dr.ing. Liviu Miclea