

SYLLABUS¹

1. Information about the Program

1.1 Higher education institution	Politehnica University of Timișoara
1.2 Faculty ² / Department ³	Automation and Computers/ Computers
1.3 Chair	-
1.4 Domain of study	Computers and Information Technology
1.5 Study level	Bachelor
1.6 Study programme / Qualification	Computers / engineer

2. Information about the Course

2.1 Course	Object-Oriented Programming						
2.2 Lecturer	Sl.dr.ing. Marinescu Cristina						
2.3 Academic staff for seminars/labs	Sl.dr.ing. Marinescu Cristina, Sl.dr.ing. Mihancea Petru-Florin						
2.4 Study year	II	2.5 Semester	3	2.6 Assessment type	E	2.7 Course type	Mandatory

3. Total time estimated (hours/ semester of didactical activities)

3.1 Hours / week	4.5	of which:	3.2 lecture hours	2.5	3.3 seminar/lab hours	2
3.4 Total curriculum hours	63	of which:	3.5 lecture hours	35	3.6 seminar/lab hours	28
Time distribution						hours
Study using manuals, support materials, bibliography and notes						20
Supplementary documentation in library, speciality electronic platforms and on site						4
Supplementary preparation for seminars/labs, homeworks, reviews, portofolios and essays						20
Tutoring activities						4
Exams						4
Other						
3.7 Total - hours of individual study						52
3.8 Total - hours per semester						115
a. Credits						5

4. Prerequisites (if appropriate)

4.1 curriculum	<ul style="list-style-type: none"> Computer Programming, Programming Techniques
4.2 competencies	<ul style="list-style-type: none"> structural programming

5. Conditions (if appropriate)

5.1 for lectures	<ul style="list-style-type: none"> a room equipped with a video projector, laptop and whiteboard
5.2 for seminars/labs	<ul style="list-style-type: none"> 15-20 computers, java compiler and virtual machines

¹ Formularul corespunde Fișei Disciplinei promovată prin OMECTS 5703/18.12.2011 (Anexa3);
² Se înscrie numele facultății care gestionează programul de studiu căruia îi aparține disciplina;
³ Se înscrie numele departamentului căruia i-a fost încredințată susținerea disciplinei și de care aparține titularul cursului;

6. Specific competencies acquired

Professional competencies ⁴	<ul style="list-style-type: none"> • Operating with fundamentals of sciences, engineering, and computer science • Designing hardware, software and communication components • Problem solving using the instruments of computer science and engineering • Improving the performance of hardware, software and communication systems • Designing, managing the lifecycle, integration and integrity of hardware, software and communication systems • Designing intelligent systems
Transversal competencies	<ul style="list-style-type: none"> • Honorable, responsible and ethical behavior, in the spirit of the law, ensuring the reputation of the profession • Identifying, describing and executing project management processes, taking on different team roles, and describing clearly and concisely, orally and in writing, in English, results from one's activity domain • Demonstrating initiative and engaging in updating one's professional, economic and organizational knowledge

7. Objectives of the course (issued from the list of the competencies acquired)

7.1 Aim	<ul style="list-style-type: none"> • The main objective of this lecture is the introduction into the object-oriented paradigm with exemplifications in Java.
7.2 Specific objectives	<ul style="list-style-type: none"> • Student will be able to structure their code into an object-oriented manner.

8. Content

8.1 Lecture	Hours	Instruction methods
1. Introduction 1.1 Main Characteristics of Object-Oriented Programming 1.2 First Steps in Java	2.5	Presentations accompanied by slides; source code samples.
2. Classes and Objects 2.1 Defining a Class. Instantiating an Object 2.2 Attributes and Methods 2.3 Java Code Conventions 2.4 Some UML Elements	2.5	
3. Sending Messages 3.1 Method Overloading 3.2 Sending Parameters in Java 3.3 Class Object	1.5	
4. Some Predefined Java Classes 4.1 Class String 4.2 Wrapper Classes 4.3 Operations for Input/Output 4.4 Arrays	3.5	
5. Inheritance 5.1 Definition 5.2 Inherited versus Overridden Members 5.3 Composition versus Inheritance	2.5	

⁴ Aspectul competențelor profesionale va fi tratat cf. Metodologiei OMECTS 5703/18.12.2011. Se vor prelua competențele care sunt precizate în Registrul Național al Calificărilor din Învățământul Superior RNCIS (http://www.rncis.ro/portal/page?_pageid=117,70218&_dad=portal&_schema=PORTAL) pentru domeniul de studiu de la pct. 1.4 și programul de studii de la pct. 1.6 din această fișă.

6. Polymorphism 6.1 Definition 6.2 Open-Closed Principle 6.3 Method Overloading versus Overriding	2.5		
7. Interfaces. Inner Classes 7.1 Defining an Interface 7.2 Implementing an Interface 7.3 Creating Inner Classes	2.5		
8. Exceptions 8.1 Concepts 8.2 Generating Exceptions 8.3 try/catch/finally Clauses 8.4 Exception-Handling Antipatterns	2.5		
9. Collections 9.1 Introducing Lists, Iterators, Sets and Maps	2.5		
10. Advanced Mechanisms 10.1 Generics 10.2 Reflection	2.5		
11. Packages and more Java Classes 11.1 Defining and Importing a Package 11.2 Visibility inside and outside a Package 11.3 Concurrency in Java	3.5		
12. Graphical User Interfaces 12.1 The Swing Event Model 12.2 Some Swing Components	3.5		
13. Towards writing Clean Code 13.1 Frequent Desing Problems in Source Code 13.2 Characteristics of Clean Code	3		
References 1. Bruce Eckel. Thinking in Java, 4th edition. Prentice Hall, 2006. 2. Robert C Martin. Clean Code. A Handbook of Agile Software Craftsmanship. Prentice Hall, 2009.			
8.2 Seminar/lab	Hours		Instruction methods Every mentioned topic in the left column will be the subject of a problem. The students will have to solve each problem and they will receive three grades during the semester (Subject 4, 7 and 11).
1. Introduction	2		
2. Classes and Objects	2		
3. Sending Messages	2		
4. Some Predefined Java Classes	4		
5. Inheritance	2		
6. Polymorphism	2		
7. Interfaces. Inner Classes	4		
8. Exceptions	2		
9. Collections	2		
10. Advanced Mechanisms	2		
11. Packages. Graphical User Interfaces	4		
References 1. Cristina Marinescu, Petru Florin Mihancea. Programare orientata pe obiecte in limbajul Java, Editura Politehnica, 2006. (The laboratory will strongly depend on the problems from each chapter of this book. The problems will be available online in English)			

9. Correlation between the course content and the requirements of the specialists in the field and the expectations of the main employers

- Currently there are many companies that code their applications into an object-oriented manner. Consequently, knowing object-oriented programming is a must.

10. Assessment

Activity type	10.1 Assessment criteria	10.2 Assessment methods	10.3 Weight in final mark
10.4 Lecture	Solving different exercises/problems into an object-oriented manner.	Written exam	50%
10.5 Seminar /labs		Arithmetic mean between two degrees got during the special evaluation weeks.	50%
10.6 Minimal performance standards (minimal specific knowledge required for passing the exam, the means to assess mastering the specific knowledge)			
<ul style="list-style-type: none"> • The difference between an object and a class • Basic object-oriented terms (abstraction, encapsulation) • Advanced notions: inheritance, polymorphism, exceptions • The use of some specific Java libraries 			

11. International compatibility

Brown University (USA) - <http://www.cs.brown.edu/courses/cs015/>

University College London (UK) - http://www.cs.ucl.ac.uk/students/syllabus/undergrad/1008_object_oriented_programming/

Stanford University (USA) - <http://www.stanford.edu/class/cs108/>

Date 15.10.2013

Signature of the course instructor

Signatures of the academic staff for seminars/labs

.....

.....

Date of approval in the Department

Signature of the Department Director

.....