SYLLABUS

1. Information regarding the programme

1.1 Higher education institution	**"Babes_Bolyai" University**
1.2 Faculty	**Faculty of Mathematics and Computer science**
1.3 Department	**Department of Computer Science**
1.4 Field of study	**Informatics(Computer Science)**
1.5 Study cycle	**Bachelor**
1.6 Study programme / Qualification	**Computer Science**

2. Information regarding the discipline

2.1 Name of the discipline	**Advanced Programming Methods**						
2.2 Course coordinator	**Assoc.Prof.PhD. Niculescu Virginia**						
2.3 Seminar coordinator	**Assoc.Prof.PhD. Niculescu Virginia**						
2.4. Year of study	**2**	2.5 Semester	**1**	2.6. Type of evaluation	**E.**	2.7 Type of discipline	**Mandatory**

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

3.1 Hours per week	5	Of which: 3.2 course	2	3.3 seminar/laboratory	1 sem. + 2 lab.
3.4 Total hours in the curriculum	70	Of which: 3.5 course	28	3.6 seminar/laboratory	14/28= 42

Time allotment:	hours
Learning using manual, course support, bibliography, course notes	20
Additional documentation (in libraries, on electronic platforms, field documentation)	10
Preparation for seminars/labs, homework, papers, portfolios and essays	23
Tutorship	7
Evaluations	20
Other activities:	-

3.7 Total individual study hours	80
3.8 Total hours per semester	150
3.9 Number of ECTS credits	6

4. Prerequisites (if necessary)

4.1. curriculum	• Object oriented programming, Algorithmics, Data structures
4.2. competencies	• Basic notions and programming skills of/in object oriented programming.

5. Conditions (if necessary)

5.1. for the course	• projector
5.2. for the seminar /lab activities	• Laboratory with computers; high level programming language environment (any Java environment)

6. Specific competencies acquired

Professional competencies	• Knowledge, understanding and use of basic concepts of object-oriented analysis and design. • Ability to work independently and/or in a team in order to solve problems in defined professional contexts. • Good programming skills in Java and other object-oriented languages
Transversal competencies	• Ability to apply design patterns in different contexts • Ability to build software projects by following the main phases in software applications development. • Ability to create projects with clear separations on architectural layers, based on different architectural patterns.

7. Objectives of the discipline (outcome of the acquired competencies)

7.1 General objective of the discipline	• Each student has to prove that (s)he acquired an acceptable level of knowledge and understanding of the subject, that (s)he is capable of stating these knowledge in a coherent form, that (s)he has correct habits of analysis, design, and implementation based on design patterns and general object oriented paradigms.
7.2 Specific objective of the discipline	• The students should have the ability to use Java language, design patterns, and to create GUI for their applications. Also they have to be able to use UML diagrams in program analysis and design.

8. Content

8.1 **Course**	Teaching methods	Remarks
1. Java Language and Platform and UML basic review - general features of Java language - Java Virtual Machine - Differences between C++ and Java - Primitive types, Reference types, Arrays - Statements UML notations for classes, associations, dependencies, generalization (review)	Exposure: description, explanation, examples, discussion of case studies	
2. Fundamentals of the Java programming language Java Language	Exposure: description, explanation,	

- Classes and Objects - Subclasses and Inheritance - Garbage Collector - Packages	examples, discussion of case studies	
3. Interfaces and Exceptions Handling - Abstract Classes - Interfaces - Template Method Design & Adaptor Patterns Exceptions handling in Java	Exposure: description, explanation, examples, discussion of case studies	
4. Java parameterized classes - generics - Generic collections Generic algorithms: package <java.util> -	Exposure: description, explanation, examples, discussion of case studies	
5. The Java I/O System - Input/ Output Streams - Decorator Design Pattern Serialization/ Persistent Objects	Exposure: description, explanation, examples, discussion of case studies	
6. GUI - Windows and graphical components - Containers and elements: window, menu, label, text field, button, etc. - Lists and Menus - Event Handling Decouplig the sender and the receiver of an event	Exposure: description, explanation, examples, discussion of case studies	
7. Model-View-Controller Pattern - Observer Design Pattern - Strategy Design Pattern Architectural Layers Decoupling presentation and logic layers	Exposure: description, explanation, examples, discussion of case studies	
8. Project development phases and iterations Project development: phase I - Plan and elaborate phase - Requirements analysis, - Use cases Project development: phase II Analyze phase Project development – different approaches	Exposure: description, explanation, examples, discussion of case studies	

Test Driven Development(TDD)		
9. Project development: Phase III: Design phase/ GRASP patterns/ UML interaction diagrams Phase IV: Construct Phase Phase V: Testing and Validation	Exposure: description, explanation, examples, discussion of case studies	
10. Multithreading programming	Exposure: description, explanation, examples, discussion of case studies	
11. Persistency - Building a persistent frameworks - Broker and Bridge Design Patterns - Virtual Proxy Design Pattern Reference and Value Object Patterns	Exposure: description, explanation, examples, discussion of case studies	
12. Event Delegation Command Pattern Repository Pattern Creational design patterns: Factory Method, Abstract Factory, Prototype, Builder	Exposure: description, explanation, examples, discussion of case studies	
13. Inversion of Control;Dependency injection; Swing Framework	Exposure: description, explanation, examples, discussion of case studies	
14. - Spring Framework, Spring JDBC	Exposure: description, explanation, examples, discussion of case studies	

Bibliography

1. *James Gosling, Bill Joy, Guy Steele, Gilad Bracha, Alex Buckley. The Java™ Language Specification Java SE 7 Edition.Copyright © 2011 Oracle America, Inc.*
2. *ECKEL, B.: Thinking in Java (3rd ed.). New York: Prentice Hall, 2002. [www.mindview.net/Books/TIJ/]*
3. *ECKEL, B.: Thinking in Patterns with Java, 2004. MindView, Inc. [http://www.mindview.net/Books/TIPatterns/]*
4. *Evans Eric. Domain-Driven Design: Tackling Complexity in the Heart of Software, Addison Wesley 2003.*
5. *GAMMA, E. - HELM, R. - JOHNSON R. - and VLISSIDES, J.: Design Patterns - Elements of*

Reusable Object-Oriented Software. Massachusets: Addison-Wesley, 1994.
6. *Larman, C.: Applying UML and Design Patterns: An Introduction to OO Analysis and Design, Berlin: Prentice Hall, 2000.*
7. *Fowler, M., Patterns of Enterprise Application Architecture, Addison-Wesley, 2002.*
8. *Fowler, Martin, UML Distilled - Third Edition, Addison Wesley, 2003.*
9. Walls, C., *Spring in Action*, Third Edition, Ed. Manning, 2011

8.2 Seminar	Teaching methods	Remarks
1. Simple Java programs.(week 1)	Explanation, dialogue, case studies	The seminar is structured as 2 hours classes every second week
2. Interfaces and Template Method Patterrn (week 3)	Dialogue, debate, case studies, examples, proofs	
3. Exception Handling (week 5)	Dialogue, debate, case studies, examples, proofs	
4. Generics – Java Framework Collections (week 7)	Dialogue, debate, case studies, examples	
5. GUI – example (week 9) Model-View-Controller example	Dialogue, debate, case studies, examples	
6. Threads based programming (week 11)	Dialogue, debate, case studies, examples	
7. Phases in project development – case study (week 13)	Dialogue, debate, case studies, examples	

Bibliography

1. *GAMMA, E. - HELM, R. - JOHNSON R. - and VLISSIDES, J.: Design Patterns - Elements of Reusable Object-Oriented Software. Massachusets: Addison-Wesley, 1994.*
2. Larman, C.: Applying UML and Design Patterns: An Introduction to OO Analysis and Design, Berlin: Prentice Hall, 2000.
3. OMG Unified Modeling Language Specifications, UML Resource Page,[www.uml.org]

8.3 Laboratory		Remarks

1. Eclipse environment Simple Java program (1week)	Explanation, dialogue, case studies	Any task should be delivered in time. For each week delay one point is lost, and the maximum delay is 2 weeks.
2. ADT – Java Implementation (2 weeks)	Testing, Explanation, Dialogue.	The grade for any task will count in the final grade in a proportion equal to the no. of weeks which were necessary for solving that task.
3. Test-Driven Development (1week)	Testing, Explanation, Dialogue.	The first 6 tasks will refer the same problem.
4. Exception Handling (1week)	Testing, Explanation, Dialogue.	
5. Input/Output operations (1week)	Testing, Explanation, Dialogue.	
6. GUI (2 week)	Testing, Explanation, Dialogue.	
7. Threads (1week)	Testing, Explanation, Dialogue.	
8. Project of medium complexity(3 weeks)	Testing, Explanation, Dialogue.	
9. Spring Framework – example (1 week)	Testing, Explanation, Dialogue.	
10. Verification test (1 week)	Testing	Final grade = (L1+L2*2+L3+L4+L5+L6*2)/8+ +(L7*4+L8)/5

Bibliography

1. Java Docs, [http://docs.oracle.com/javase/6/docs/]
2. Java Tutorials, [http://www.oracle.com/technetwork/java/index.html]
3. Spring Framework [http://www.springsource.org]

9. Corroborating the content of the discipline with the expectations of the epistemic community, professional associations and representative employers within the field of the program

- The course respects the IEEE and ACM Curricula Recommendations for Computer Science studies;
- The course exists in the studying program of all major universities in Romania and abroad;
- The content of the course is considered the software companies as important for average programming and design skills.

10. Evaluation

Type of activity	10.1 Evaluation criteria	10.2 Evaluation methods	10.3 Share in the grade (%)
10.4 Course	- know the basic principles of the domain; - apply the course concepts - problem solving	Written exam(partial ex.+final ex) Practical exam	50% 20%
10.5 Seminar/lab activities	- be able to apply the OO principles and patterns	-documentation -portfolio -continuous observations	5%
	- be able to solve using OO design and programming problems of medium complexity	-Practical laboratories	25%
10.6 Minimum performance standards			
➢ At least grade 5 (from a scale of 1 to 10) at both written exam and practical exam.			

Date Signature of course coordinator Signature of seminar coordinator

....Niculescu Virginia.....Niculescu Virginia.....

Date of approval Signature of the head of department

....….....