

SYLLABUS

1. Information regarding the programme

1.1 Higher education institution	Babeş Bolyai University
1.2 Faculty	Faculty of Mathematics and Computer Science
1.3 Department	Department of Computer Science
1.4 Field of study	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. Ing. Florin Craciun						
2.3 Seminar coordinator	Assoc. Prof. PhD. Ing. Florin Craciun						
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	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	<ul style="list-style-type: none"> Object oriented programming, Algorithmics, Data structures
4.2. competencies	<ul style="list-style-type: none"> Basic notions and programming skills

5. Conditions (if necessary)

5.1. for the course	<ul style="list-style-type: none"> projector
5.2. for the seminar /lab activities	<ul style="list-style-type: none"> Laboratory with computers; high level programming language environment (any Java environment, any C# environment)

6. Specific competencies acquired

Professional competencies	<ul style="list-style-type: none"> • 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 object-oriented languages especially in Java and C#
Transversal competencies	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • The students should have the ability to use Java language, C# language, design patterns, and to create GUI for their applications. Also they have to be able to use object-oriented concepts in program analysis and design.

8. Content

8.1 Course	Teaching methods	Remarks
1. - Introduction to Java platform and .Net platform. - Basic concepts of the object-oriented languages. - Basic Java and C# languages elements: expressions, statements, primitive data types	Exposure,description, explanation, debate and dialogue, discussion of case studies	
2. Object-oriented fundamentals of the Java and C# programming: - Classes and Objects - Arrays - Enum Types - Inheritance - Interfaces - Abstract Classes - Nested classes	Exposure,description, explanation, debate and dialogue, discussion of case studies	
3. Object-oriented fundamentals of the Java and C# programming: - Reference Types - Value Types - Polymorphism, - Overriding	Exposure,description, explanation, debate and dialogue, discussion of case studies	

<ul style="list-style-type: none"> - Overloading - Casting 		
<p>4. Object-oriented fundamentals of the Java and C# programming:</p> <ul style="list-style-type: none"> - Encapsulation - Access Modifiers - Java Packages - C# Namespaces - C# Properties - C# Indexes - C# Delegates and Events 	Exposure,description, explanation, debate and dialogue, discussion of case studies	
<p>5. Exceptions Handling in Java and C#</p>	Exposure,description, explanation, discussion of case studies	
<p>6. Reusable code using Generics Types in Java and C#</p> <ul style="list-style-type: none"> - Java Generics: Generics Types, Raw Types, Generic Methods, Bounded Type Parameters, Wildcards: Upper Bounded, Unbounded, Lower Bounded, Subtyping - C# Generics: Generic Types, Generic Constraints, Subtyping, Generic Methods 	Exposure,description, explanation, discussion of case studies	
<p>7. I/O Libraries in Java and C#</p>	Exposure,description, explanation, debate and dialogue, discussion of case studies	
<p>8. Reflection and its applications in Java and C#</p> <ul style="list-style-type: none"> - serialization 	Exposure,description, explanation, debate and dialogue, discussion of case studies	
<p>9. Collections in Java and C#</p> <ul style="list-style-type: none"> - classes, interfaces, algorithms 	Exposure,description, explanation, discussion of case studies	
<p>10. Concurrent Programming in Java and C#</p> <ul style="list-style-type: none"> - basic concepts of multithreading - Threads - Synchronization 	Exposure,description, explanation, discussion of case studies	
<p>11. Concurrent Programming in Java and C#</p> <ul style="list-style-type: none"> - Thread interference - Memory consistency errors - Synchronization: locks, synchronized methods, synchronized statements - Atomic access - Immutable objects - Liveness: Deadlock, Starvation, Livelock 	Exposure,description, explanation, discussion of case studies	
<p>12. Concurrent Programming in Java and C#</p> <ul style="list-style-type: none"> - High-Level Concurrency - Mutex - Semaphore - Signaling - Synchronization Contexts 	Exposure,description, explanation, discussion of case studies	

13. GUI programming in Java and C# - Event-driven programming - Java Swing	Exposure,description, explanation, discussion of case studies	
14. GUI programming in Java and C# -Java Swing - C# Windows Forms	Exposure,description, explanation, discussion of case studies	

Bibliography

1. James Gosling, Bill Joy, Guy Steele, Gilad Bracha, Alex Buckley. The Java™ Language Specification Java SE 7 Edition.
2. Eckel, B., Thinking in Java, 4th edition, Prentice Hall, 2006
3. Eckel, B.: Thinking in Patterns with Java, 2004. MindView, Inc
4. E. Gamma, R. Helm, R. Johnson, J. Vlissides, Design Patterns – Elements of Reusable Object Oriented Software, Ed. Addison Wesley, 1994
5. Joseph Albahari and Ben Albahari, C# 4.0 in a Nutshell, Fourth Edition, O’Reilly, 2010
6. ***, Microsoft Developer Network, Microsoft Inc., <http://msdn.microsoft.com/>
7. ***, The Java Tutorial, 2013. <http://download.oracle.com/javase/tutorial/>

8.2 Seminar	Teaching methods	Remarks
1. Model-View-Controller pattern, ADT implemented with Arrays and Linked Lists in Java	Conversation, debate, case studies, examples	The seminar is structured as 2 hours classes every second week
2. Model-View-Controller pattern, ADT implemented with Arrays and Linked Lists in C#	Conversation, debate, case studies, examples	
3. Polymorphism, Exception Handling in Java and C#	Conversation, debate, case studies, examples	
4. Generic Types in Java and C#	Conversation, debate, case studies, examples	
5. Reflection in Java and C#, Proxy pattern and Factory pattern in C# and Java	Conversation, debate, case studies, examples	
6. Observer pattern in Java and C#, Event-driven programming	Conversation, debate, case studies, examples	
7. GUI programming in Java and C#	Conversation, debate, case studies, examples	

Bibliography

1. James Gosling, Bill Joy, Guy Steele, Gilad Bracha, Alex Buckley. The Java™ Language Specification Java SE 7 Edition.
2. Eckel, B., Thinking in Java, 4th edition, Prentice Hall, 2006
3. E. Gamma, R. Helm, R. Johnson, J. Vlissides, Design Patterns – Elements of Reusable Object Oriented Software, Ed. Addison Wesley, 1994
4. Joseph Albahari and Ben Albahari, C# 4.0 in a Nutshell, Fourth Edition, O’Reilly, 2010
5. ***, Microsoft Developer Network, Microsoft Inc., <http://msdn.microsoft.com/>
6. ***, The Java Tutorial, 2013. <http://download.oracle.com/javase/tutorial/>

8.3. Laboratory	Teaching methods	Remarks
-----------------	------------------	---------

1. The project allocation. Simple Java and C# programs. Discovering Java Eclipse and Visual Studio for C#(2 weeks)	Explanation, Discussion, Evaluation	The students work on the same project (implemented in Java and also in C#) from week1 to week14. Each laboratory assignment refers to the same project. Each assignment is graded separately. There are 0.5 points penalties for each delayed week
2. Use Model-View-Controller pattern to organize the project code. ADT implementation in Java and C# (2 weeks)	Explanation, Discussion, Evaluation	
3. Treat the errors using exceptions in Java and C# (1 week)	Explanation, Discussion, Evaluation	
4. Generalize the code using generics types in Java and C# (2 weeks)	Explanation, Discussion, Evaluation	
5. Add I/O operations and serialization in Java and C# (2 weeks)	Explanation, Discussion, Evaluation	
6. Use Java and C# Collections (1 week)	Explanation, Discussion, Evaluation	
7. Observer pattern in Java and C# (2 weeks)	Explanation, Discussion, Evaluation	
8. GUI in Java and C# (2 weeks)	Explanation, Discussion, Evaluation	
9. Verification TEST		
Bibliography		
1. ***, Microsoft Developer Network, Microsoft Inc., http://msdn.microsoft.com/		
2. ***, The Java Tutorial, 2013. http://download.oracle.com/javase/tutorial/		

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

- | |
|--|
| <ul style="list-style-type: none"> The course respects the IEEE and ACM Curricula Recommendations for Computer Science studies; The content of the course is considered by the software companies as important for average software development 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 principle of the domain; - apply the course concepts - problem solving	Written final exam	20%
		Practical final exam	30%
10.5 Seminar/lab activities	- be able to use course concepts in solving the real problems	Laboratories Assignments Practical Test	35%

			15%
10.6 Minimum performance standards			
➤ At least grade 5 (from a scale of 1 to 10) at written final exam and practical final exam. At least grade 5 for the final grade.			

Date

Signature of course coordinator

Signature of seminar coordinator

.....

Assoc. Prof. PhD. Florin CRACIUN

Assoc. Prof. PhD. Florin CRACIUN

Date of approval

Signature of the head of department

.....

.....