**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 | **Undergraduate** |
| 1.6 Study programme / Qualification | **Computer Science** |

## 2. Information regarding the discipline

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 2.1 Name of the discipline | | | | **Metode Avansate de Programare**<br>**Advanced Programming Methods** | | | |
| 2.2 Course coordinator | | | | **Assoc. Prof. Eng. Florin Craciun** | | | |
| 2.3 Seminar coordinator | | | | **Assoc. Prof. Eng. Florin Craciun** | | | |
| 2.4. Year of study | **2** | 2.5 Semester | **3** | 2.6. Type of evaluation | **E** | 2.7 Type of discipline | **Compulsory** |

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

| | | | | | |
|---|---|---|---|---|---|
| 3.1 Hours per week | 6 | Of which: 3.2 course | 2 | 3.3 seminar/laboratory | 2 sem. + 2 lab. |
| 3.4 Total hours in the curriculum | 84 | Of which: 3.5 course | 28 | 3.6 seminar/laboratory | 28 sem + 28 lab |
| 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 | | | | | 26 |
| Tutorship | | | | | 5 |
| Evaluations | | | | | 5 |
| Other activities: ..... | | | | | - |

| | |
|---|---|
| 3.7 Total individual study hours | 66 |
| 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 |

## 5. Conditions (if necessary)

| | |
|---|---|
| 5.1. for the course | projector |
| 5.2. for the seminar /lab activities | projector |

## 6. Specific competencies acquired

| | |
|---|---|
| **Professional competencies** | • Knowledge, understanding and use of basic concepts of object-oriented analysis and design.<br>• Ability to work independently and/or in a team in order to solve problems in defined professional contexts.<br>• Good programming skills in object-oriented languages especially in Java |
| **Transversal competencies** | • Ability to apply design patterns in different contexts<br>• Ability to build software projects by following the main phases in software applications development.<br>• 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 ofknowledge 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 object-oriented concepts in program analysis and design. |

**8. Content**

| 8.1 Course | Teaching methods | Remarks |
|---|---|---|
| 1. Introduction to Java platform: platform, language syntax, primitive data types, arrays, classes, interfaces, packages, enums, overriding, overloading, exceptions | Exposure,description, explanation, debate and dialogue, discussion of case studies | |
| 2. Collections and Generic Types: anonymous classes, polymorphism, casting | Exposure,description, explanation, debate and dialogue, discussion of case studies | |
| 3. IO,NIO: binary and character oriented streams, files, channels and buffers | Exposure,description, explanation, debate and dialogue, discussion of case studies | |
| 4. Functional programming: lambda expressions, streams | Exposure,description, explanation, debate and dialogue, discussion of case studies | |

| | | |
|---|---|---|
| 5. GUI: Java FX components, event handling | Exposure,description, explanation, discussion of case studies | |
| 6. Concurrency: threads, executors, futures, exception handling | Exposure,description, explanation, discussion of case studies | |
| 7. Concurrency: sync vs async methods, callback methods, cancellation | Exposure,description, explanation, debate and dialogue, discussion of case studies | |
| 8. XML: schema, documents | Exposure,description, explanation, debate and dialogue, discussion of case studies | |
| 9. GUI (cont.):FXML, CSS. Metaprogramming: reflection, serialization | Exposure,description, explanation, discussion of case studies | |
| 10. Introduction in C# and .Net | Exposure,description, explanation, discussion of case studies | |
| 11. Collections in C# | Exposure,description, explanation, discussion of case studies | |
| 12. IO operations in C# | Exposure,description, explanation, discussion of case studies | |
| 13. GUI in C# | Exposure,description, explanation, discussion of case studies | |
| 14. LINQ | 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. ***, The Java Tutorial, 2013. http://download.oracle.com/javase/tutorial/
6. Joseph Albahari and Ben Albahari, C# 4.0 in a Nutshell, Fourth Edition, O'Reilley, 2010

| | | |
|---|---|---|
| 7. ***, Microsoft Developer Network, Microsoft Inc., http://msdn.microsoft.com/ | | |
| **8.2 Seminar and 8.3 Laboratories** | **Teaching methods** | **Remarks** |
| 1. Java basic project | Conversation, debate, case studies, examples | |
| 2. Java project: Collections, Generics | Conversation, debate, case studies, examples | |
| 3. Java project: Generics | Conversation, debate, case studies, examples | |
| 4. Java project: IO | | |
| 5. Java project: Functional programming | Conversation, debate, case studies, examples | |
| 6. Java project: GUI | Conversation, debate, case studies, examples | |
| 7. Java project: concurrency | Conversation, debate, case studies, examples | |
| 8. Java project:xml | | |
| 9. Java project: GUI | | |
| 10. C# project basics | | |
| 11. C# project collections | | |
| 12. C# project io | | |
| 13. C# project GUI | | |
| 14. C# project Linq | | |

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'Reilley, 2010
5. ***, Microsoft Developer Network, Microsoft Inc., http://msdn.microsoft.com/
6. ***, 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**

- The course respects the IEEE and ACM Curriculla 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;<br>- apply the course concepts<br>- problem solving | Written final exam<br><br>Practical final exam | 25%<br><br>35% |

| | | | |
|---|---|---|---|
| 10.5 Seminar/lab activities | - be able to use course concepts in solving the real problems | Laboratories Assignments<br><br>Seminar Activity | 35%<br><br>5% |

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.
- Rules:
- **You can change your subgroup for the lab only once at the first lab. You have to announce the lab teacher about this. After the first lab you cannot change your subgroup lab time.**
- **you have to present each lab assignment at its deadline**
- **- for each lab assignment you will get a grade between 1 to 10**
- **- the deadline for each lab assignment is clearly specified in the assignment text file**
- **- if you delay an assignment 1 week you can get maximum 7 on that assignment**
- **- if you delay an assignment more than 1 week you will automatically get the grade 0 for it and you cannot submit it anymore**
- **- the final grade for the lab activity is the arithmetic average of the lab assignments grades**
- **you have to implement all the assignments since the problems of the final practical exam are extensions of the lab assignments**
- - the lab assignments mainly consist of a big project to implement an interpreter (virtual machine) of an imperative concurrent toy language
- - at each lab assignment (almost each week) you will add the rules and the data structures required to execute one or more new instructions of the toy language
- - the toy language interpreter will be implemented in Java
- 
- - a schedule of the lab assignments (periodically updated) can be found at LabAssignmentsSchedule.pdf
- **The first condition to get into the final exam is to attend minimum 90% of the labs and minimum 70% of the seminars. That means you must attend minimum 10 seminars and minimum 12 laboratories. Please read the following document:**
- **http://www.cs.ubbcluj.ro/wp-content/uploads/Hotarare-CDI-15.03.2017.pdf**
- **- Holydays and first week are considered by default attended**
- **The second condition to get into the final exam is to get minimum grade 5 at the lab activity.**
- **Rules for the Students from previous years ("Restantieri"): the students must attend the labs and the seminars, must do the lab assignments, and must pass the final exam**
- **- in order to pass the final exam you must have:**
- **-- at least 5 at the final theoretical exam and**
- **-- at least 5 at the final practical exam and**
- **-- the final grade must be at least 5**
- - you can pass either both the final theoretical exam and the final practical exam or nothing
- **Rules for the second exam ("restanta"): The first condition to get into the final exam is to attend minimum 90% of the labs and minimum 70% of the seminars. That means you must attend minimum 10 seminars and minimum 12 laboratories. Please read the following document:**

**http://www.cs.ubbcluj.ro/wp-content/uploads/Hotarare-CDI-15.03.2017.pdf**

**Holydays and 2 October are considered by default attended**

**The second condition to get into the final exam is to get minimum grade 5 at the lab activity.**

**in order to pass the final second exam you must have:**

**--- at least 5 at the final theoretical exam and**

**--- at least 5 at the final practical exam and**

**--- the final grade is 5**

- you can pass either both the second final theoretical exam and the second final practical exam or nothing

| 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 |
| --- | --- |
| ..... | .....…..... |