**SYLLABUS**

## 1. Information regarding the programme

| | |
|---|---|
| 1.1 Higher education institution | **Babeş-Bolyai University of Cluj-Napoca** |
| 1.2 Faculty | **Faculty of Mathematics and Computer Science** |
| 1.3 Departament | **Departament of Computer Science** |
| 1.4 Field of study | **Mathematics** |
| 1.5 Ciclul de studii | **Bachelor** |
| 1.6 Study cycle / Qualification | **Mathematics Computer Science** |

## 2. Information regarding the discipline

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 2.1 Name of the discipline | **Advanced Programming Methods** | | | | | | |
| 2.2 Course coordinator | **Assoc. Prof. PhD Bocicor Maria Iuliana** | | | | | | |
| 2.3 Seminar coordinator | **Assoc. Prof. PhD Bocicor Maria Iuliana** | | | | | | |
| 2.4 Year of study | **1** | 2.5 Semester | **2** | 2.6. Type of evaluation | **E** | 2.7. Type of discipline | **Compulsory** |
| 2.8. Code | **MLE5008** | | | | | | |

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

| | | | | | |
|---|---|---|---|---|---|
| 3.1 Hours per week | 4 | Of which: 3.2 course | 2 | 3.3 seminar/laboratory | 1sem 1 lab |
| 3.4 Total hours in the curriculum | 56 | Of which: 3.5 course | 28 | 3.6 seminar/laboratory | 14+ 14 |
| 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 | | | | | 28 |
| Tutorship | | | | | 4 |
| Evaluations | | | | | 7 |
| Other activities: ..... | | | | | |

| | |
|---|---|
| 3.7 Total individual study hours | 69 |
| 3.8 Total hours per semester | 125 |
| 3.9 Number of ECTS credits | 5 |

## 4. Prerequisites (if necessary)

| | |
|---|---|
| 4.1 curriculum | Fundamentals of Programming, Object Oriented Programming, Data Structures and Algorithms |
| 4.2 competencies | Average programming skills in a high level programming language |

## 5. Conditions (if necessary)

| | |
|---|---|
| 5.1 For the course | • Classroom with projector |

| | |
|---|---|
| 5.2 For the seminar/lab activities | • Laboratory with computers; Java, C# and programming languages, IntelliJ IDEA/Eclipse, Visual Studio IDE<br>• Classroom with projector |

**6. Specific competencies acquired**

| | |
|---|---|
| **Professional competencies** | • C1.1 Knowledge, understanding and use of basic concepts of object oriented analysis and design.<br>• C1.2 Ability to work independently and/or in a team in order to solve small and medium scale problems.<br>• C1.3 Good programming skills in object-oriented languages especially in Java.<br>• C1.4 Application of design patterns in different contexts.<br>• C1.5 Creation of projects with clear separations on architectural layers, based on different architectural patterns. |
| **Transversal competencies** | • CT1 Application of efficient and rigorous working rules, manifest responsible attitudes towards the scientific and didactic fields, respecting the professional and ethical principles.<br>• CT2 Use of efficient methods and techniques for learning, information, research and development of abilities for knowledge exploitation, for adapting to the needs of a dynamic society and for communication in Romanian as well as in a widely used foreign language. |

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

| | |
|---|---|
| 7.1 General objective of the discipline | • To prepare an object-oriented design of small/medium scale problems and to learn the Java programming language, as well as to create graphical user interfaces. |
| 7.2 Specific objectives of the discipline | • To use object-oriented concepts in program analysis and design.<br>• To use and implement solutions in the Java programming language.<br>• To create GUI for the given requirements.<br>• To apply design patterns in various contexts.<br>• To use classes written by other programmers when constructing their systems. |

**8. Content**

| **8.1 Course** | Teaching methods | Remarks |
|---|---|---|
| **1. Introduction in Java**<br>• Platform<br>• Language syntax | • Interactive exposure<br>• Explanation<br>• Conversation | |

| | | |
|---|---|---|
| • Data types. Arrays<br>• Examples | • Examples<br>• Didactical demonstration | |
| **2. Classes, inheritance**<br>• Classes<br>• Object construction<br>• Methods<br>• Inheritance, polymorphism<br>• Abstract classes, interfaces | • Interactive exposure<br>• Explanation<br>• Conversation<br>• Examples<br>• Didactical demonstration | |
| **3. Generic types, collections in Java**<br>• Generic methods<br>• Type erasure<br>• Generic classes and subtyping<br>• Wildcards<br>• Java Collections Framework | • Interactive exposure<br>• Explanation<br>• Conversation<br>• Examples<br>• Didactical demonstration | |
| **4. Exceptions, Java I/O, JUnit**<br>• Exceptions<br>• Java I/O, streams, serialization<br>• JUnit | • Interactive exposure<br>• Explanation<br>• Conversation<br>• Examples<br>• Didactical demonstration | |
| **5. JDBC, Functional programming**<br>• JDBC API<br>• Java 8 features: Lambda expressions, Java 8 Streams | • Interactive exposure<br>• Explanation<br>• Conversation<br>• Examples<br>• Didactical demonstration | |
| **6. Graphical User Interfaces**<br>• JavaFX applications, scenes, layouts, UI controls<br>• Events | • Interactive exposure<br>• Explanation<br>• Conversation<br>• Examples<br>• Didactical demonstration | |
| **7. Graphical User Interfaces**<br>• Processing events<br>• Model-View-Controller<br>• FXML | • Interactive exposure<br>• Explanation<br>• Conversation<br>• Examples<br>• Didactical demonstration | |
| **8. Java Reflection, Concurrency**<br>• Java Reflection API<br>• Concurrency: processes, threads, multithreaded programming in Java | • Interactive exposure<br>• Explanation<br>• Conversation<br>• Examples<br>• Didactical demonstration | |
| **9. Concurrency**<br>• Threads in Java<br>• Thread synchronization<br>• Concurrent applications in Java | • Interactive exposure<br>• Explanation<br>• Conversation<br>• Examples<br>• Didactical demonstration | |
| **10. Design Patterns** | • Interactive exposure | |

| | | |
|---|---|---|
| • Creational patterns<br>• Structural patterns<br>• Behavioural patterns | • Explanation<br>• Conversation<br>• Examples<br>• Didactical demonstration | |
| **11. Design Patterns (cont.), Introduction in C# and .NET** | • Interactive exposure<br>• Explanation<br>• Conversation<br>• Examples<br>• Didactical demonstration | |
| **12. C# and .NET**<br>• The .NET Architecture<br>• The C# programming language<br>• Classes in C#<br>• Generics<br>• Delegates<br>• Events<br>• Lambda expressions<br>• LINQ | • Interactive exposure<br>• Explanation<br>• Conversation<br>• Examples<br>• Didactical demonstration | |
| **13. Revision**<br>• Revision of the most important topics covered by the course<br>Examination guide | • Interactive exposure<br>• Explanation<br>• Conversation<br>• Examples<br>• Didactical demonstration | |
| **14. Written examination** | | |
| | | |

**Bibliography**


1. James Gosling, Bill Joy, Guy Steele, Gilad Bracha, Alex Buckley.
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, Addison-Wesley Longman Publishing, 1995.
5. The Java Tutorials: https://docs.oracle.com/javase/tutorial/
6. Joseph Albahari and Ben Albahari, C# 4.0 in a Nutshell, Fourth Edition, O'Reilley, 2010.


| **8.2 Seminar** | Teaching Methods | Remarks |
|---|---|---|
| 1. Simple problems in Java. Classes. Layered architecture. | • Interactive exposure<br>• Explanation<br>• Conversation<br>• Examples<br>• Didactical demonstration | The seminar is structured as a 2 hour class, every 2 weeks. |
| 2. Inheritance, interfaces, packages, iterators. | | |
| 3. Generics, collections, exceptions. | | |
| 4. Serialization, files, JDBC. | | |
| 5. Graphical User Interfaces with JavaFX. | | |
| 6. Concurrency, threads. | | |
| 7. Design patterns. | | |

**Bibliography**


1. James Gosling, Bill Joy, Guy Steele, Gilad Bracha, Alex Buckley.
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, Addison-Wesley Longman Publishing, 1995.
5. The Java Tutorials: https://docs.oracle.com/javase/tutorial/
Joseph Albahari and Ben Albahari, C# 4.0 in a Nutshell, Fourth Edition, O'Reilley, 2010.

| **8.3 Laboratory** | Teaching Methods | Remarks |
|---|---|---|
| 1. Setting up JDK, JRE and JVM, as well as an IDE of choice. Simple problems in Java. | • Explanation<br>• Conversation | • The laboratory is structured as a 2 hour class, every 2 weeks.<br>• Laboratory assignments are due 2 weeks after assignment. |
| 2. Layered architecture, generics, exceptions. | | |
| 3. Files, serialization, JUnit. | | |
| 4. JDBC, functional programming (Java 8 streams). | | |
| 5. Laboratory test. | | |
| 6. Graphical User Interfaces. | | |
| 7. Practical examination. | | |

**Bibliography**

1. James Gosling, Bill Joy, Guy Steele, Gilad Bracha, Alex Buckley.
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, Addison-Wesley Longman Publishing, 1995.
5. The Java Tutorials: https://docs.oracle.com/javase/tutorial/
6. Joseph Albahari and Ben Albahari, C# 4.0 in a Nutshell, Fourth Edition, O'Reilley, 2010.

**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 follows the 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 Lecture | The correctness and completeness of the accumulated knowledge and the capacity to design and implement correct C++ programs. | Written examination (Colloquium) | **30%** |
| 10.5 Seminar/ Laboratory | Be able to use course concepts in solving real problems. | Practical examination (Colloquium) | **30%** |
| | Correctness of delivered laboratory assignments and laboratory tests. | Laboratory assignments.<br>Laboratory test.<br>Observation during the semester. | **40%** |
| 10.6 Minimum performance standards | | | |
| • Each student has to prove that they acquired an acceptable level of knowledge and understanding of the core concepts taught in the class, that they are capable of using knowledge in a coherent form, that | | | |

they have the ability to establish certain connections and to use the knowledge in solving different problems in Java.

- For participating at the examination attendance is compulsory for seminar and for laboratory activities, as follows: minimum 5 attendances for seminar and minimum 6 attendances for laboratory activities.
- Successfully passing of the examination is conditioned by a minimum grade of 5 for each of the following: practical examination, written examination and final grade.

| Date | Signature of course coordinator | Signature of seminar coordinator |
|---|---|---|
| 06.07.2023 | Assoc. Prof. PhD. Bocicor Maria Iuliana | Assoc. Prof. PhD. Bocicor Maria Iuliana |

Date of approval

Signature of the head of department
Prof. PhD. Laura Diosan