COMPUTER PROGRAMMING 2

Grado en Computación e Inteligencia Artificial / Bachelor in Computer Science and Artificial Intelligence BCSAI SEP-2025 CP2-CSAI.3.M.A

Area Computer Science

Number of sessions: 30

Academic year: 25-26

Degree course: THIRD

Number of credits: 6.0

Semester: 1º

Category: COMPULSORY

Language: English

Professor: **ANAS ABBOOD**

E-mail: aabbood@faculty.ie.edu

Dr. Anas Abbood holds a Ph.D. in Computer Science from Universiti Putra Malaysia (Faculty of Computer Science and Information Technology), an M.Sc. in Computer Science and Data Security from the University of Technology, Iraq, a Higher Diploma in Computer Science and Data Security from the National Computer Center in Baghdad, and a B.Sc. in Operations Research from Mansour University College, Iraq.

She is the holder of patent P12012001329 and has published numerous research papers in international peer-reviewed journals. Her research interests include human-computer interaction, computer vision and visualization, computer-integrated health systems, geometric modeling, and augmented reality.

Dr. Abbood is an experienced university lecturer and academic mentor with over 20 years of teaching, research, and academic leadership across international institutions. She currently serves as a professor at Saint Louis University – Madrid Campus

Office Hours

Office hours will be on request. Please contact at:

Fridays, 10:00 AM – 11:20 AM
(Or by appointment via email)

SUBJECT DESCRIPTION

This course deepens programming skills using Java, focusing on object-oriented design, advanced data structures, and efficient algorithm implementation. Students will develop scalable applications, explore Java-based AI programming, and gain industry-relevant coding skills.

This course offers a comprehensive introduction to Java programming, progressing from core language fundamentals to advanced object-oriented design and application development. Through a blend of theory and hands-on practice, students will explore essential programming constructs, control flow, data structures, and Java APIs. Key object-oriented principles such as encapsulation, inheritance, polymorphism, and abstraction are deeply emphasized, alongside modern software engineering practices.

Students will also engage in individual and team-based projects, leading to a final application that showcases their ability to design and implement maintainable, modular Java software. By course end, students will be prepared to build robust Java applications using professional tools and methodologies.

LEARNING OBJECTIVES

By the end of the course, students will:

1. Master Java OOP principles (Encapsulation, Inheritance, Polymorphism).
2. Implement advanced data structures (Trees, Graphs, Hash Tables).
3. Optimize algorithms for performance & scalability.
4. Apply multi-threading & parallel programming in Java.
5. Develop basic AI applications using Java libraries (e.g., Deeplearning4j).
6. Work with unit testing, debugging, and performance profiling.

By the end of this course, students will be able to:

Understand Java's Core Concepts

Explain Java’s history, ecosystem (JDK, JRE, JVM), and distinguish it from other programming languages.

Write and Run Java Programs

Develop, compile, and execute Java applications using modern IDEs and tools such as IntelliJ IDEA, Maven, and Gradle.

Apply Fundamental Programming Techniques in Java

Use variables, operators, control flow (if-else, switch), and loops to write basic Java programs.

Utilize Object-Oriented Programming (OOP)

Design and implement classes and objects using encapsulation, inheritance, polymorphism, and abstraction.

Modularize and Reuse Code

Define and invoke methods, overload functions, and understand variable scope and parameter passing.

Handle Exceptions Effectively

Implement robust error-handling techniques using try-catch blocks, custom exceptions, and best practices.

Develop Interactive Applications with JavaFX

Build GUI applications using JavaFX UI controls, multimedia components, and event-driven

programming.

Practice Software Engineering Principles

Apply design principles (e.g., SOLID), design patterns, and modular architecture to create maintainable software.

TEACHING METHODOLOGY

IE University's teaching approach emphasizes active, collaborative, and applied learning. Students engage in live sessions, hands-on activities, peer reviews, and instructor feedback..

This is achieved through a diverse range of teaching methods and learning activities, including but not limited to:

Learning Activity	Weighting	Estimated time a student should dedicate to prepare for and participate in
Lectures	40.0 %	60.0 hours
Group work	26.7 %	40.0 hours
Individual studying	33.3 %	50.0 hours
TOTAL	100.0 %	150.0 hours

AI POLICY

In this course, the use of generative artificial intelligence (GenAI) is encouraged, with the goal of developing an informed critical perspective on potential uses and generated outputs.

However, be aware of the limits of GenAI in its current state of development:

If you provide minimum effort prompts, you will get low quality results. You will need to refine your prompts to get good outcomes. This will take work

Don't take ChatGPT's or any GenAI's output at face value. Assume it is wrong unless you either know the answer or can cross-check it with another source. You are responsible for any errors or omissions. You will be able to validate the outputs of GenAI for topics you understand.

LLMs are a good productivity/search tool when you are an expert on the field and can thus be critical of its output. Be aware of this when using it for learning.

AI is a tool, but one that you need to acknowledge using. Failure to do so is in violation of academic honesty policies. Acknowledging the use of AI will not impact your grade.

Coding LLMs (Github copilot) are really good at writing boiler-plate code (which C++ has lots of). This is an incredible time-saver when one understands the produced code. Be aware and critical of the impact generating this kind of code has for your learning process

PROGRAM

The following description of the material covered is tentative. An attempt will be made to cover all listed topics and to include other advanced topics that will help the student throughout their career in computer science. However, the pace of the classes will depend on individual and group performance, which may introduce some variations in the syllabus

SESSION 1 (LIVE IN-PERSON)

Course Overview and Introduction to Java

Course objectives, structure, and expectations
Introduction to Java
History and evolution of Java
Java compared to other programming languages
First Java Program

SESSION 2 (LIVE IN-PERSON)

Java tools and environments

Java Language Specification, API, JDK, JRE, and IDE
Creating, Compiling, and Executing a Java Program
Developing Java Programs Using IntelliJ IDEA

SESSION 3 (LIVE IN-PERSON)

Elementary Programming in Java

Reading input from the console
Variables and Constants
Assignment Statements and Assignment Expressions
Naming Conventions
Data types
Exercises in class

SESSION 4 (LIVE IN-PERSON)

Elementary Programming in Java

Numeric data types and operators
Numeric Literals
Evaluating Expressions and Operator Precedence
Augmented Assignment Operators
Increment and Decrement Operators
Numeric type conversions
Exercises in class

SESSION 5 (LIVE IN-PERSON)

Elementary Programming in Java

boolean Data Types, Values, and Expressions
If statements
Two-Way if-else Statements
Nested if and Multi-Way if-else Statements
Logical Operators

Conditional Operators

Switch statements

Common Errors and Pitfalls

Exercises in class

SESSION 6 (LIVE IN-PERSON)

Loops in Java

The while loop

The do while loop

The for loop

Nested loops

Loop design strategy

Which loop to use

Exercises in class

SESSION 7 (LIVE IN-PERSON)

Methods in Java

Method declaration and invocation

Parameters and return values

Void methods and value returning methods

Modularizing Code

Overloading methods

The scope of variables

Exercises in class

SESSION 8 (LIVE IN-PERSON)

Arrays in Java

Array Basics

Processing arrays

Foreach Loops

Copying arrays

Passing arrays to methods

Variable-Length Argument Lists

Searching and sorting arrays

Multidimensional arrays

Exercises on arrays

SESSION 9 (LIVE IN-PERSON)

Quiz - 1

Quiz on fundamental Java topics

Summarizing topics covered for fundamentals of Java

SESSION 10 (LIVE IN-PERSON)

Objects and classes - 1

Defining classes and objects

Constructors

Accessing Objects via Reference Variables

Object’s data and methods

Passing objects to methods

this keyword

Static Variables, Constants, and Methods

Exercises in class

SESSION 11 (LIVE IN-PERSON)

Objects and classes - 2

Visibility modifiers

Immutable objects and classes

The scope of variables

this keyword

Differences between Variables of Primitive Types and Reference Types

Exercises in class

SESSION 12 (LIVE IN-PERSON)

Object-Oriented Thinking

Thinking in objects

Class relationships

The String class

Case study

SESSION 13 (LIVE IN-PERSON)

Encapsulation in Java

Getters and setters

Encapsulation principles

Immutable classes

Exercises in class

SESSION 14 (LIVE IN-PERSON)

Inheritance in Java

- Superclasses and Subclasses
- Using the **super** keyword
- Constructor chaining
- Overriding methods
- Overriding vs. Overloading
- The Object Class and Its **toString()** Method
- Exercise in class

SESSION 15 (LIVE IN-PERSON)

Polymorphism in java

- Introduction to Polymorphism
- Dynamic binding
- Casting objects
- The Object’s equals Method
- The ArrayList Class
- The instanceof operator
- Exercises in class

SESSION 16 (LIVE IN-PERSON)

Abstract classes and Interfaces

- Abstract classes
- Why abstract methods are used
- Case study
- Defining and implementing interfaces
- Interfaces vs. Abstract Classes
- Multiple interfaces and ambiguity resolution
- Exercises in class

SESSION 17 (LIVE IN-PERSON)

Object-Oriented Design Principles

- SOLID principles overview
- Composition vs inheritance
- Common OO design pitfalls
- Shallow vs deep modules
- Exercises in class

SESSION 18 (LIVE IN-PERSON)

Quiz on Object Oriented Programming

A quiz covering all OOP related topics covered

A summary of fundamental OOP concepts covered

SESSION 19 (LIVE IN-PERSON)

Individual Project Workshop

Introduction to individual project

Expectation and guideline

Timeline and deliverable

SESSION 20 (LIVE IN-PERSON)

Compilation and Packaging

Building with Maven/Gradle

Creating **.jar** files

Package management with Maven Central

Delivering Java projects

SESSION 21 (LIVE IN-PERSON)

Error Handling in Java

Exception-Handling Overview

Exception Types

Declaring, Throwing, and Catching Exceptions

Creating custom exceptions

Best practices for exception design

Exercises in class

SESSION 22 (LIVE IN-PERSON)

JavaFX Basics

Introduction to JavaFX

JavaFX vs. Swing and AWT

Panes, Groups, UI Controls, and Shapes

Property binding

Color and Font class

Image and ImageView Classes

Exercises in class

SESSION 23 (LIVE IN-PERSON)

JavaFX UI Controls and Multimedia

Labeled and Label

Button, CheckBox, Radio Button

TextField, ComboBox

List View, Scroll Bar, Slider

Video and Audio

Exercises in class

SESSION 24 (LIVE IN-PERSON)

Final Project Workshop - 1

Introduction to final project idea

Expectation and guideline

Timeline and deliverable

SESSION 25 (LIVE IN-PERSON)

Generics in Java

Motivations and Benefits

Generic methods and classes

Type parameters and wildcards (?, extends, super)

Bounded types

Raw types and type erasure

Exercises in class

SESSION 26 (LIVE IN-PERSON)

Sets and Maps in Java

Sets and Hashsets

Sets vs Lists

Maps in Java

Singleton and Unmodifiable Collections and Maps

Exercises in class

SESSION 27 (LIVE IN-PERSON)

Final Project Workshop - 2

Asses progress of final project

Work on final project

SESSION 28 (LIVE IN-PERSON)

Design Patterns

Introduction to design patterns

Creational design patterns

Structural design patterns

Behavioral design patterns

SESSION 29 (LIVE IN-PERSON)

Final Project Presentation

Students present their final projects

Feedback and evaluation

Course review

SESSION 30 (LIVE IN-PERSON)

Final Exam

Final exam

EVALUATION CRITERIA

criteria	percentage	Learning Objectives	Comments
Final Exam	25 %		
Individual Assignments/Projects	15 %		
Group Project	25 %		
Class Participation and Attendance	10 %		
Intermediate Exam	25 %		

RE-SIT / RE-TAKE POLICY

Each student has four chances to pass any given course distributed over two consecutive academic years: ordinary call exams and extraordinary call exams (re-sits) in June/July.

Students who do not comply with the 80% attendance rule during the semester will fail both calls for this Academic Year (ordinary and extraordinary) and have to re-take the course (i.e., re-enroll) in the next Academic Year.

Evaluation criteria:

Students failing the course in the ordinary call (during the semester) will have to re-sit the exam in June / July (except those not complying with the attendance rule, who will not have that opportunity and must directly re-enroll in the course on the next Academic Year).

The extraordinary call exams in June / July (re-sits) require your physical presence at the campus you are enrolled in (Segovia or Madrid). There is no possibility to change the date, location or format of any exam, under any circumstances. Dates and location of the June / July re-sit exams will be posted in advance. Please consider this when planning your summer.

The June / July re-sit exam will consist of a comprehensive exam. Your final grade for the course will depend on your performance in this exam only; continuous evaluation over the semester will not be taken into consideration. Students will have to achieve the minimum passing grade of 5 and can obtain a maximum grade of 8.0 (out of 10.0) – i.e., “notable” in the re-sit exam.

Retakers: Students who failed the subject on a previous Academic Year and are now re-enrolled as re-takers in a course will be needed to check the syllabus of the assigned professor, as well as contact the professor individually, regarding the specific evaluation criteria for them as retakers in the course during that semester (ordinary call of that Academic Year). The maximum grade that may be obtained in the retake exam (3rd call) is 10.0.

After the professor grades ordinary and extraordinary call exams, you will have the possibility to attend a review session for that exam and course grade. Please be available to attend the session in order to clarify any concerns you might have regarding your exam. Your professor will inform you about the time and place of the review session. Any grade appeals require that the student attended the review session prior to appealing.

Students failing more than 18 ECTS credits after the June / July re-sits will be asked to leave the Program. Please, make sure to prepare yourself well for the exams in order to pass your failed subjects.

In case you decide to skip the opportunity to re-sit for an exam during the June/July extraordinary call, you will need to enroll in that course again for the next Academic Year as a retaker and pay the corresponding extra cost. As you know, students have a total of four allowed calls to pass a given subject or course, in order to remain in the program.

BIBLIOGRAPHY

Recommended

- Marc Loy, Patrick Niemeyer, Daniel Leuck. (2023). *O'REILLY Learning Java.* 6th. O'Reilly Media, Inc.. ISBN 9781098145538 (Digital)

An Introduction to Real-World Programming with Java

- Y. Daniel Liang. (2024). *Introduction to Java Programming and Data Structures.* 13. Pearson. ISBN 0138123536 (Digital)

Seamlessly integrates programming, data structures and algorithms into 1 text. he 13th Edition is aligned to the latest Java 18 technology and completely revised in every detail to enhance the clarity, presentation, content, examples, and exercises.

- Josh Bloch. (2017). *Effective Java 3rd Edition.* 3. Addison-Wesley Professional. ISBN 0134686042 (Digital)

The Definitive Guide to Java Platform Best Practices–Updated for Java 7, 8, and 9

BEHAVIOR RULES

Please, check the University's Code of Conduct here. The Program Director may provide further indications.

ATTENDANCE POLICY

Please, check the University's Attendance Policy here. The Program Director may provide further indications.

ETHICAL POLICY

Please, check the University's Ethics Code here. The Program Director may provide further indications.