University of Miskolc
Department of Information Technology

Full-time programme
2025/26. I. semester

GEIAL31A-B2a
Java programming

The goal of this course is to provide a comprehensive introduction to the Java programming language and its core concepts. Students will learn the fundamentals of object-oriented programming, including classes, objects, inheritance, and polymorphism, as well as essential language features such as exception handling, collections, and input/output operations. Throughout the course, emphasis will be placed on writing clean, efficient, and maintainable code following best practices. Students will gain hands-on experience through practical exercises and projects that reinforce theoretical knowledge. By the end of the course, students will be confident in developing Java applications and will have a solid foundation for further studies in advanced Java frameworks and enterprise-level development.

Course lecturer: Tamás Tompa PhD, assistant professor
Preliminary requirements: GEIALXXX-XX (object oriented programming)
Course completion: signature and exam, 5 credits
Contact hours per week: 2 hours lecture, 2 hours laboratory
Lecture time and place: Tuesday 8-10 In/101
Laboratory time, place, instructor: Tuesday 10-12, In/101, Tamás Tompa, PhD

SCHEDULE

Week	Lecture	Laboratory
1.	Introduction to Java: History of Java, versions, platform independence, role of JVM, JRE, JDK, Basic philosophy of the Java language, Basic development environments, compilation and execution	Hello World, Compile & Run, Primitives and Simple Calculations
2.	Features and basic concepts of Java: High-level language features, overview of basic OOP concepts, comparison of C++ and Java, first simple Java programs	if, switch, loops, Simple algorithms (summation, maximum search)
3.	Classes and objects: Class definition, data members, methods, Constructors, initialization, instantiation, Visibility levels, encapsulation, Getter / setter patterns	Writing your own class, Constructors, getter / setter Instance management
4.	OOP I. Unit containment, inheritance, polymorphism: Abstraction and information hiding, How inheritance works, superclass calls, The role of polymorphism, Inheritance of constructors and its rules	Parent–child class pair, Method override, toString / equals override
5.	OOP II. Basics of abstract classes, interfaces, final, generics: Final fields, methods, Abstract classes and their design role, Using interfaces, multiple interface implementations, Introducing generic types	Interface + implementation, Abstract class practice, Simple generic container
6.	java.lang package and the Object ancestor: Object:equals, hashCode , toString, clone, Exception hierarchy basics, Thread management basics, String and character handling	String operations, equals – hashCode task, Simple thread management
7.	Data Structures I. Arrays, Matrices, Cloning: One-dimensional and multidimensional arrays, Initialization, array operations (length, clone, copy), Sorting, search algorithms on arrays	Filling an array with random numbers, Sorting, searching, Matrix operations
8.	Data Structures II. ArrayList and Collections Basics: Dynamic arrays	List filling, filtering, Sorting

	(ArrayList) operations, Adding, deleting, searching, sorting, Motivation of collections	with comparator, List ↔ array conversion
9.	java.util package: collections, sets, lists: Collection API, HashSet, TreeSet, LinkedHashSet, SortedSet, NavigableSet, Iterator, ListIterator usage	Set types, Iterator, ListIterator , Comparator with multiple aspects
10.	Graphical interfaces: AWT, Swing, JavaFX, Overview of GUI frameworks, Swing components (button, label, panel, layouts), Event handling, Creating a simple GUI application	Simple window + button, Event handling, Mini, project: calculator / form
11.	File Management I.: Streams, Reader / Writer, binary and text files, Stream concepts, types (byte/char, input/output), File reading/writing, Buffered types, character encoding	Text file reading/writing, BufferedReader / Writer, Object serialization basics
12.	File Management II.: Object Serialization, NIO, path operations, Writing/reading objects to files, RandomAccessFile, Java NIO (Path, Files, FileSystem)	Path & Files operations, Copy, delete, attributes, RandomAccessFile
13.	Database programming with JDBC: JDBC architecture, driver types, Connecting to SQLite, Executing SQL from Java, Statement, PreparedStatement, ResultSet	SQLite database, INSERT/SELECT from Java, using PreparedStatement
14.	Extras, XML, JSON, CSV, ZIP/JAR, PDF: XML: DOM basics, reading/writing, JSON handling (org.json, Gson), CSV reading/writing, ZIP and JAR file handling, PDF generation	XML read/write, CSV-JSON processing, ZIP, packaging, PDF generation

Course schedule and materials (slides):
www.iit.uni-miskolc.hu → Staff: Tamás Tompa → Courses → Java programming /
https://users.iit.uni-miskolc.hu/~tompa

Requirements for obtaining the signature and grade:

1. Signature: mid-term test assignment, attendance at least 70% of laboratory classes (10/14) and 60% of lectures (9/14)
2. Exam: theoretical and practical assignment

Recommended literature:

1. Lecture and laboratory materials (https://users.iit.uni-miskolc.hu/~tompa)
2. Cay S. Horstmann - Core Java Volume I – Fundamentals (11th Ed .)
3. Herbert Schildt - Java: A Beginner's Guide (8th Ed .)
4. Kathy Sierra & Bert Bates - Head First Java (2nd or 3rd Ed.)
5. Joshua Bloch - Effective Java (3rd Ed.)
6. Herbert Schildt - Java: The Complete Reference (12th Ed.)
7. Allen B. Downey & Chris Mayfield - Think Java

Miskolc, February 2, 2026

Tamas Tompa, PhD
Course coordinator