BACSE102 Problem Solving Using JAVA

Premanand S

Assistant Professor
School of Electronics Engineering
Vellore Institute of Technology
Chennai Campus

premanand.s@vit.ac.in

December 9, 2025

Course Overview: BACSE102 - Computer Programming: Java

- **Credits:** 2 (0-0-4)
- **Total Hours:** 60
- **Objective:** Build strong foundations in problem-solving through Java programming.
- **Core Areas Covered:**
  - Java fundamentals and OOP
  - Exception handling and multithreading
  - File handling and I/O streams
  - Collections, Generics, Lambda
  - Basic database connectivity (JDBC)

Why This Course Matters to Every Engineering Student

Computer Programming with Java is not just for CSE students.

- **For CSE/IT:** Core language to understand OOP, software design, algorithms, backend systems.
- **For ECE/EEE:** Required for embedded systems, IoT applications, automation scripts, data acquisition, and control logic.
- **For Mechanical:** Used in simulations, robotics control, CAD/CAM automation, intelligent systems, and Industry 4.0.
- **For Civil:** Useful for modelling, simulation, data handling, construction automation, GIS tools, and workflow automation.

Programming is today's "engineering mathematics"—a universal tool for all fields.

Similarities Between Python and Java

Both Python and Java share several core programming concepts, making it easier for Python learners to transition into Java.

- Both are high-level, general-purpose languages
- Both support Object-Oriented Programming (OOP)
- Both emphasize readability and structured programming
- Both have automatic memory management (Garbage Collection)
- Both have large standard libraries
- **Both are platform independent** Python uses interpreters; Java uses JVM and bytecode
- Both are widely used in industry - Web development, mobile apps, automation, AI, enterprise systems.

Differences Between Python and Java

Feature	**Python**	**Java**
Typing System	Dynamically typed	Statically typed (variables must have declared types)
Compilation Model	Interpreted line-by-line	Compiled to bytecode, executed by JVM
Syntax Style	Indentation-based, no braces or semicolons	Uses braces { } and semicolons ;
Speed	Slower for CPU-heavy tasks	Faster due to JIT compilation and strong typing
Use Cases	AI/ML, scripting, automation, data science	Enterprise apps, Android apps, banking, backend systems
Memory Management	Automatic, very abstracted	Automatic but with more structure (types, OOP design)
Error Handling	Mostly runtime errors	Strict compile-time + runtime exception checking

Question 1

Where is Java used in the real world?

Question 1

Where is Java used in the real world?

Java is used in:

- Large-scale enterprise software (banking, insurance, healthcare)
- Android mobile apps
- Cloud applications (AWS, Google Cloud)
- Embedded systems and smart devices
- Web applications and backend servers
- Big data platforms (Hadoop ecosystem)

Java powers billions of devices and mission-critical systems worldwide.

Question 2

Do I really need Java if I already know Python?

Question 2

Do I really need Java if I already know Python?

Yes, because:

- Companies expect engineers to know more than one language.
- Python is great for scripting and AI; Java is strong for scalable software.
- Java teaches strict structure, which improves problem-solving discipline.
- Many core subjects (OOP, data structures) align better with Java.
- Understanding both makes you adaptable and industry-ready.

Question 3

Will learning Java help in placements?

Question 3

Will learning Java help in placements?

Absolutely. Java is one of the top languages required for:

- Product-based companies (Amazon, Flipkart, PayPal)
- Service companies (TCS, Infosys, Wipro, Cognizant)
- Banking and fintech (HDFC, JPMorgan, Goldman Sachs)
- Android development companies
- Companies using Microservices, Spring Boot, and backend systems

Java developers are always in high demand.

Question 4

Is Java still in demand today?

Question 4

Is Java still in demand today?

Yes, very much. Java remains:

- One of the top 3 most used languages globally
- The backbone of enterprise applications
- A preferred language for scalable backend systems
- Essential for Android and cloud-native development

Question 5

Is Java difficult compared to Python?

Question 5

Is Java difficult compared to Python?

Not difficult—just more structured.

- Python is easier for beginners due to simple syntax.
- Java enforces clear rules (types, classes, structure).
- This discipline helps you become a stronger programmer.
- Once you understand the syntax, logic becomes easier.

The challenge is short-term; the skill is long-term.

Question 6

What jobs can I get by learning Java?

Question 6

What jobs can I get by learning Java?

You can become:

- Java Developer / Backend Developer
- Android App Developer
- Full-Stack Developer (Java + JS)
- Cloud Microservices Engineer
- Big Data Engineer (Hadoop/Spark)
- Automation Engineer

Java skills open doors across multiple industries.

Question 7

How does Java help students from non-CSE branches?

Question 7

How does Java help students from non-CSE branches?

Mechanical: Robotics, automation scripts, simulation tools

Civil: GIS systems, modelling tools, construction automation

EEE/ECE: IoT devices, embedded applications, control software

All branches: Logic building, automation, and problem-solving

Question 8

How important is Java for coding interviews?

Question 8

How important is Java for coding interviews?

- Most coding platforms (HackerRank, LeetCode) fully support Java.
- Strong OOP concepts give you an advantage.
- Java collections simplify solving data structure problems.
- Many companies explicitly ask for Java in coding rounds.

Java is one of the best languages for competitive programming and interviews.

Question 9

Can learning Java improve my logic and problem-solving?

Question 9

Can learning Java improve my logic and problem-solving?

Yes! Java strengthens:

- Analytical and logical thinking
- Understanding of algorithms and OOP concepts
- Structured coding habits
- Ability to design modular solutions

Java builds a strong foundation for any future programming career.

Assessment Design Overview

- 4 Programming Assessment Tests (PATs)
- 1 Competitive Participation Component
- 1 Practice Progress Component

PAT Distribution

PAT No.	Topics Covered	Marks	Tentative Duration
PAT 1	Java Basics, Variables, Data Types, Looping Constructs and Arrays	30	75 Minutes
PAT 2	Classes and Objects, Inheritance, Polymorphism and Abstract Classes	30	75 Minutes
PAT 3	Packages and Exception Handling	30	75 Minutes
PAT 4	Serialization, Deserialization and Collection Framework	15	45 Minutes

Detailed PAT Composition (PAT 1 to 3)

Exam Time: 75 Minutes
Marks: 30
Weightage: 15 Marks Each

Component	**Questions**	**Marks**	**Approx. Tim**
Coding	1 Easy/Medium (5) + 1 Hard (10)	15	60 minutes
MCQs	6 × 1 mark	6	6 minutes
Debugging	6 × 1.5 marks	9	9 minutes
Total		**30**	**75 minutes**

Final score out of 30 will be scaled to 15 marks per PAT.

PAT 4 Composition

Exam Time: 45 Minutes
Marks: 15
Weightage: 8 Marks

Component	**Questions**	**Marks**	**Approx. Time**
Coding	1 Medium	7.5	37.5 minutes
MCQs	3 × 1 mark	3	3 minutes
Debugging	3 × 1.5 marks	4.5	4.5 minutes
Total		**15**	**45 minutes**

Final score out of 15 will be scaled to 8 marks.

Other Internal Assessments (8 Marks)

Component	**Marks**
Competitive Participation (Codeathon by IamNeo)	4
IamNeo Practice Progress (measured via platform)	3

Event eligibility and scoring will be decided by the course committee.

PAT Evaluation Ratio Summary

Component	**Approximate Weight (%)**
Coding	50%
MCQs	20%
Debugging	30%

FAT (Final Assessment Test) Structure – 50 Marks

Component	**Marks**	**Mode**	**Description**
Part A: Theory Section	10	Paper-based	Algorithm / Pseudocode (5 × 2 marks): Step-by-step logic before implementation
Part B: Practical Section	40	IamNeo (Online)	Conducted via IamNeo in secure exam mode
Coding Section	25	IamNeo	1 Medium (10 marks), 1 Hard (15 marks) — Logic + output + edge cases
Debugging Section	10	IamNeo	10 questions × 1 mark each (Syntax + logic correction)
MCQs	5	IamNeo	10 questions × 0.5 marks (Concept understanding)
Total	**50**		

Module 1: Introduction to Java Fundamentals

- OOP Paradigm and Features of Java
- JVM, Bytecode, Java Program Structure
- Data Types, Variables, Naming Conventions
- Operators, Control and Looping Constructs
- One- and Multi-dimensional Arrays
- Enhanced for-loop
- Strings, StringBuffer, StringBuilder, Math Class
- Wrapper Classes

Module 2: Object-Oriented Design

- Class Fundamentals, Access/Non-access Specifiers
- Declaring Objects, Object References, Array of Objects
- Constructors, Destructors
- *this* and *static*
- Enums and Iterations
- Inheritance Types, *super*, *final*
- Polymorphism: Overloading vs Overriding
- Abstract Classes and Interfaces

Module 3: Packages, Exception Handling, and Multithreading

- Creating and Accessing Packages, Sub-packages
- Exception Types and Control Flow
- try, catch, finally, throw, throws
- User-defined Exceptions
- Basics of Threads
- Runnable Interface
- Multithreading and Inter-thread Communication
- Synchronization

Module 4: IO Streams and Files

- File Object
- I/O Streams
    - FileInputStream, FileOutputStream
    - FileReader, FileWriter
    - DataInputStream, DataOutputStream
    - BufferedInputStream, BufferedOutputStream
    - PrintOutputStream
- Serialization and Deserialization

Module 5: Generic Programming and Java Collections

- Generic Classes and Methods
- List and Map Interfaces
- ArrayList, LinkedList, TreeMap, HashMap
- Lambda Expressions
- JDBC Architecture
- CRUD operations (Create, Read, Update, Delete)

Module 6: Contemporary Issues

- Latest trends in Java ecosystem
- Industry use-cases and applications

Popular Platforms to Code and Run Java Programs

- **IntelliJ IDEA (Most Popular)** Industry-standard IDE used by professionals; excellent code suggestions and debugging.

Popular Platforms to Code and Run Java Programs

- **IntelliJ IDEA (Most Popular)** Industry-standard IDE used by professionals; excellent code suggestions and debugging.
- **Eclipse IDE** Widely used in academia and industries for Java and enterprise applications.

Popular Platforms to Code and Run Java Programs

- **IntelliJ IDEA (Most Popular)** Industry-standard IDE used by professionals; excellent code suggestions and debugging.
- **Eclipse IDE** Widely used in academia and industries for Java and enterprise applications.
- **NetBeans IDE** Simple, beginner-friendly IDE maintained by Apache.

Popular Platforms to Code and Run Java Programs

- **IntelliJ IDEA (Most Popular)** Industry-standard IDE used by professionals; excellent code suggestions and debugging.
- **Eclipse IDE** Widely used in academia and industries for Java and enterprise applications.
- **NetBeans IDE** Simple, beginner-friendly IDE maintained by Apache.
- **VS Code + Java Extensions** Lightweight, fast, and supports Java through Microsoft Red Hat extensions.

Popular Platforms to Code and Run Java Programs

- **IntelliJ IDEA (Most Popular)** Industry-standard IDE used by professionals; excellent code suggestions and debugging.
- **Eclipse IDE** Widely used in academia and industries for Java and enterprise applications.
- **NetBeans IDE** Simple, beginner-friendly IDE maintained by Apache.
- **VS Code + Java Extensions** Lightweight, fast, and supports Java through Microsoft Red Hat extensions.
- **BlueJ** Designed for beginners to visualize classes and objects; great for OOP learning.

Popular Platforms to Code and Run Java Programs

- **IntelliJ IDEA (Most Popular)** Industry-standard IDE used by professionals; excellent code suggestions and debugging.
- **Eclipse IDE** Widely used in academia and industries for Java and enterprise applications.
- **NetBeans IDE** Simple, beginner-friendly IDE maintained by Apache.
- **VS Code + Java Extensions** Lightweight, fast, and supports Java through Microsoft Red Hat extensions.
- **BlueJ** Designed for beginners to visualize classes and objects; great for OOP learning.
- **JGrasp** Easy interface; automatically generates control structure diagrams.

Popular Platforms to Code and Run Java Programs

- **IntelliJ IDEA (Most Popular)** Industry-standard IDE used by professionals; excellent code suggestions and debugging.
- **Eclipse IDE** Widely used in academia and industries for Java and enterprise applications.
- **NetBeans IDE** Simple, beginner-friendly IDE maintained by Apache.
- **VS Code + Java Extensions** Lightweight, fast, and supports Java through Microsoft Red Hat extensions.
- **BlueJ** Designed for beginners to visualize classes and objects; great for OOP learning.
- **JGrasp** Easy interface; automatically generates control structure diagrams.
- **Online Platforms**
    - Replit.com
    - JDoodle.com
    - OnlineGDB.com
    - HackerRank, LeetCode, CodeChef, Codeforces

Thank You!

Stay Connected

Premanand S

Email:	premanand.s@vit.ac.in
Phone:	+91-7358679961
LinkedIn:	Premanand S
Instagram:	premsanand
WhatsApp Channel:	anandsDataX
Google Scholar:	Premanand S
GitHub:	anandprems