

Object Oriented Programming with JAVA – Predicted Question Paper

CONTENTS

Object Oriented Programming with JAVA

General Instructions

SECTION A

SECTION B

Marks Distribution at a Glance

After You Finish – Self Check

Object Oriented Programming with JAVA

Predicted Question Paper · Semester End Examination

Subject Code	303105205 / 03010503PC03
Programme	B.Tech — CSE, Artificial Intelligence & Machine Learning
Semester	III
Academic Year	2026–27
Total Marks	60
Time Allowed	3 Hours

General Instructions

1. This paper has two sections – Section A and Section B. Attempt all questions.
2. Every question carries an internal OR. Attempt only one alternative from each.
3. Q1 is compulsory and contains four short parts.
4. Marks for each part are shown in square brackets.
5. Draw neat labelled diagrams wherever the question asks for one.

6. Write programs in Java. Keep the syntax correct – a program that would not compile loses marks.
7. Write answers in this order: **definition** 'explanation' **example** 'program' **output**.

What each command word actually wants

Command word	What the examiner expects
Define	One or two clear sentences. No more.
Explain	Definition, then <i>how it works</i> , then an example.
Differentiate	A table with at least four rows.
Write a program	Complete code plus the expected output.
Describe with a diagram	The diagram is compulsory – a paragraph alone loses marks.

Suggested time split

Question	Marks	Spend about
Q1	10	20 minutes
Q2	10	25 minutes
Q3	10	30 minutes
Q4	10	35 minutes
Q5	10	30 minutes

Q6	10	30 minutes
Revision	—	10 minutes

If you run short on time, do Q4 first. Unit 4 alone carries 30% of the syllabus weight.

SECTION A

Q1 — Answer all four parts [$4 \times 2.5 = 10$]

- Define Object-Oriented Programming. List any four features of an object-oriented language.
- Differentiate between JDK, JRE and JVM.
- Differentiate between checked and unchecked exceptions. Give one example of each.
- What is autoboxing and unboxing? Give an example.

Q2 [$2 \times 5 = 10$]

- Explain the JVM architecture with a neat diagram. Describe the Java compilation and execution process.
- Explain the four pillars of OOP – Encapsulation, Inheritance, Polymorphism and Abstraction – with a suitable example for each.

OR

- a) Differentiate between **procedural programming** and **object-oriented programming**. Explain any **two** features of OOP.
 - b) What is **bytecode**? Why is Java called **platform-independent**? Explain the structure of a basic Java program.
-

Q3 [2 × 5 = 10]

- a) Explain **type casting**. Differentiate between **implicit (widening)** and **explicit (narrowing)** casting with examples. Explain **operator precedence** with an example.
- b) Write a Java program to find the **second-largest element** in an array **without using built-in sorting methods**. Handle duplicate values correctly.

OR

- a) Differentiate between **String**, **StringBuffer** and **StringBuilder**. Explain with examples when each one should be used.
 - b) Write a Java program to **remove duplicate elements** from a list and display the unique elements in **sorted order**.
-

SECTION B

Q4 [2 × 5 = 10]

- a) What is constructor overloading? Explain the types of constructors with examples. Explain the use of the **this** and **super** keywords.
- b) Explain method overloading and method overriding with examples. Differentiate between them on at least four points.

OR

- a) Explain inheritance in Java. Describe the types of inheritance with examples. Why does Java not support multiple inheritance through classes?
- b) What is an interface? Differentiate between an abstract class and an interface. Write a program demonstrating multiple inheritance using interfaces.
-

Q5 [2 × 5 = 10]

- a) Explain the try-catch-finally block with an example. Explain multiple catch blocks and the rule about their order.
- b) Differentiate between **throw** and **throws**. Write a Java program to create a custom exception **InvalidAgeException** that is thrown when age is less than 18.

OR

- a) Explain the exception hierarchy. Differentiate between **Error** and **Exception**. Differentiate between checked and unchecked exceptions.
- b) Explain exception-handling best practices in Java. Mention at least five.
-

Q6 [2 × 5 = 10]

a) Explain the **List** interface. Differentiate between **ArrayList** and **LinkedList**. Write a program to demonstrate any **three methods** of **ArrayList**.

b) Explain the **Map** interface. Differentiate between **HashMap** and **TreeMap**. Write a program to store and retrieve **student records** using **HashMap**.

OR

a) What is **multithreading**? Explain the **two ways to create threads** in Java with examples. Explain the **thread life cycle** with a diagram.

b) Explain **thread synchronization**. Write a program using **two Runnable threads** and a **shared counter**.

Marks Distribution at a Glance

Question	Units covered	Course Outcome	Marks
Q1	Mixed – 1, 2, 3, 5	CO1–CO5	10
Q2	Unit 1 – OOP and Java	CO1	10
Q3	Units 2 and 3 – Basics, Arrays, Strings	CO2, CO3	10
Q4	Unit 4 – Object Oriented Programming	CO3, CO4	10
Q5	Unit 5 – Exception Handling	CO4, CO5	10

Q6	Units 6 and 7 — Collections, Advanced Java	CO6	10
Total			60

After You Finish — Self Check

Run through this list before you close your answer sheet.

#	Check	Marks at risk
1	Did every explain answer include a code example ?	High
2	Did every differentiate answer have at least 4 rows ?	High
3	Did every program include its expected output ?	High
4	Did Q2(a) and Q6(a) include the diagrams ?	Medium
5	Did you skip the OR alternative instead of attempting both?	Medium
6	Are the imports written for programs that use ArrayList, HashMap or Scanner?	Medium

7	Did you number your answers to match the question numbers?	Low
8	Is there a final 10 minutes left to re-read?	Low

Remember: the most common way to lose marks on this paper is writing a definition and stopping. Every explain question is worth 5 marks – 1 for the definition, 2 for how it works, and 2 for the example and its output.