

Paper-XIII: Programming in Java

Comprehensive Question Bank (All Units Covered Explicitly)

Unit-I: Introduction to Java, Syntax, Constructs & I/O Streams

1 Mark Short Questions

- Q1.** Who developed the Java Programming language and in which year? [1 Mark]
- Q2.** What is the full form of JVM, JRE, and JDK? [1 Mark]
- Q3.** State the major architectural feature that makes Java platform-independent. [1 Mark]
- Q4.** How does Java realize its 'Write Once, Run Anywhere' (WORA) capability? [1 Mark]
- Q5.** Name any two major semantic differences between C++ and Java. [1 Mark]
- Q6.** Why is there no pointer data type available to developers in Java? [1 Mark]
- Q7.** Which command line tool is used to compile a Java program? [1 Mark]
- Q8.** What file extension is generated after successful compilation of a Java source code? [1 Mark]
- Q9.** Define a literal or constant value in Java. [1 Mark]
- Q10.** List any four fundamental primitive data types in Java. [1 Mark]
- Q11.** What is a wrapper class in Java? Give one example. [1 Mark]
- Q12.** State the purpose of the 'final' keyword when applied to variables. [1 Mark]
- Q13.** What is the function of the static keyword when used with a variable? [1 Mark]
- Q14.** What is the difference between standard bitwise AND (&) and logical AND (&&) operators? [1 Mark]
- Q15.** Write the correct syntax for a single-line comment and a multi-line comment in Java. [1 Mark]
- Q16.** Which built-in system object is used to print output to the standard console view? [1 Mark]
- Q17.** What is the default return type of the main() method in Java? [1 Mark]
- Q18.** How are command line arguments passed into a Java application program? [1 Mark]
- Q19.** Which package must be imported to utilize the Scanner class for user input? [1 Mark]
- Q20.** Name the class from java.io package that provides buffered character-based input readings. [1 Mark]

2 Marks Concept Questions

- Q1.** Differentiate between Java's Bytecode and native Machine Code. [2 Marks]
- Q2.** Explain how memory management differs fundamentally between C++ and Java. [2 Marks]

- Q3.** What are keywords in Java? Explain the specific roles of 'this' and 'super'. [2 Marks]
- Q4.** Explain automatic type promotion and explicit type casting rules in Java expressions. [2 Marks]
- Q5.** What is a Wrapper class? Explain its necessity with respect to Java collection framework collections. [2 Marks]
- Q6.** Describe the flow of control within a nested switch-case conditional statement. [2 Marks]
- Q7.** Compare the behavior of the while loop versus the do-while loop structure. [2 Marks]
- Q8.** How does variable scope limit accessibility within nested method blocks? [2 Marks]
- Q9.** Define method signature. What components make up a valid Java method signature? [2 Marks]
- Q10.** What is the fundamental difference between passing arguments by value versus by reference? [2 Marks]
- Q11.** Explain how Type Checking is strictly enforced during Java program compilation. [2 Marks]
- Q12.** List two built-in methods of the Java Math class along with their functions. [2 Marks]
- Q13.** How do you safely extract an integer from a command line string array argument? [2 Marks]
- Q14.** Compare the execution mechanism of Scanner versus BufferedReader for reading inputs. [2 Marks]
- Q15.** What happens if an invalid input format is parsed using Scanner's nextInt() method? [2 Marks]

5 Marks Analytical Questions

- Q1.** Draw and explain the internal architecture of the Java Virtual Machine (JVM) detailing execution flows. [5 Marks]
- Q2.** Provide a detailed tabular comparison highlighting structural, memory, and syntax differences between C++ and Java. [5 Marks]
- Q3.** Explain all primitive data types in Java with their corresponding sizes in bytes, ranges, and default initial values. [5 Marks]
- Q4.** Explain bitwise operators in Java (AND, OR, XOR, Complement, Left Shift, Right Shift, Unsigned Right Shift) with code examples. [5 Marks]
- Q5.** Elucidate the decision-making constructs in Java. Provide clear code structures for if-else ladders and nested loops. [5 Marks]
- Q6.** Discuss the concept of Java Methods. Explain defining scope, returning structures, and parameter passing mechanisms. [5 Marks]
- Q7.** Describe the complete steps to compile and execute a Java program manually using standard JDK CLI commands. [5 Marks]
- Q8.** Explain Type Conversion and Type Checking. Distinguish between implicit widening and explicit narrowing conversions. [5 Marks]
- Q9.** Demonstrate how to capture varied data inputs (String, int, double) using the Scanner class with a complete program. [5 Marks]

Q10. Write an executable Java program that reads text lines using `BufferedReader` and handles standard input streams cleanly. [5 Marks]

8 Marks Comprehensive Questions

Q1. Deconstruct the complete execution cycle of a Java program from writing source code to runtime execution inside JVM. [8 Marks]

Q2. Provide a complete architectural analysis of Java's core features including Security, Robustness, Portability, and Object-Orientation. [8 Marks]

Q3. Develop an end-to-end executable Java application demonstrating conditional structures, looping constructs, and nested iteration blocks. [8 Marks]

Q4. Analyze Java's Input/Output landscape via console. Compare Command Line Arguments, `Scanner` class, and `BufferedReader` class comprehensively. [8 Marks]

Q5. Design a comprehensive Java class containing multiple methods that illustrate deep usage of argument passing types and local scopes. [8 Marks]

Unit-II: Object-Oriented Programming, Inheritance, Interfaces & Packages

1 Mark Short Questions

Q1. State the four fundamental principles of Object-Oriented Programming. [1 Mark]

Q2. What is a class definition in the context of Java OOP? [1 Mark]

Q3. How do you create a live reference instance or object of a class in Java? [1 Mark]

Q4. What is an object reference variable? [1 Mark]

Q5. Can an object instance be passed directly as a parameter to a method in Java? [1 Mark]

Q6. What is the primary role of the Garbage Collector in Java? [1 Mark]

Q7. Name the keyword used to explicitly trigger or suggest garbage collection execution. [1 Mark]

Q8. What is a constructor in Java? [1 Mark]

Q9. State the default return type specification of a standard Java constructor. [1 Mark]

Q10. What is constructor overloading? [1 Mark]

Q11. Which keyword represents the current executing instance of a class block? [1 Mark]

Q12. What keyword is used to call a superclass constructor from a subclass code? [1 Mark]

Q13. Does Java support direct multiple inheritance through classes? (Yes/No) [1 Mark]

Q14. Which keyword is used by a class to inherit another class properties? [1 Mark]

- Q15.** Which keyword is used by a class to adopt a declared interface structure? [1 Mark]
- Q16.** Define Polymorphism in a single clear sentence. [1 Mark]
- Q17.** What is Up-Casting in Java class relationships? [1 Mark]
- Q18.** Define Auto-Boxing with a simple inline code or textual example. [1 Mark]
- Q19.** What is an Enumeration (enum) structure in Java? [1 Mark]
- Q20.** Which keyword is used to create and declare a custom package namespace? [1 Mark]

2 Marks Concept Questions

- Q1.** Differentiate between Class Variables (static fields) and Instance Variables. [2 Marks]
- Q2.** Explain the structural and behavioral properties of a 'final' class. [2 Marks]
- Q3.** How does a parameterized constructor differ from a default non-parameterized constructor? [2 Marks]
- Q4.** Explain the exact utility of constructor overloading with a clear conceptual scenario. [2 Marks]
- Q5.** Differentiate between Aggregation (HAS-A) and Inheritance (IS-A) relationships. [2 Marks]
- Q6.** Explain the core functional differences between using 'extends' vs 'implements'. [2 Marks]
- Q7.** List and sketch the types of inheritance supported directly by Java class hierarchies. [2 Marks]
- Q8.** Define an Interface and state its primary difference from a standard class. [2 Marks]
- Q9.** What is Down-Casting, and when is an explicit cast required to prevent runtime faults? [2 Marks]
- Q10.** Explain Unboxing and provide a small example code snippet showing it. [2 Marks]
- Q11.** Explain Method Overloading and its purpose in compile-time polymorphism. [2 Marks]
- Q12.** What is Method Overriding? State any one basic rule for overriding a method. [2 Marks]
- Q13.** What are the specific access restriction rules when overriding a superclass method? [2 Marks]
- Q14.** Explain the concept of pre-defined packages in Java with two standard library examples. [2 Marks]
- Q15.** How does importing a specific class differ from importing an entire package using an asterisk? [2 Marks]

5 Marks Analytical Questions

- Q1.** Explain the life-cycle of an object in Java from instantiation via the 'new' operator to its final Garbage Collection. [5 Marks]
- Q2.** Discuss the complete mechanics of constructors. Explain default, parameterized, and copy constructor concepts with examples. [5 Marks]
- Q3.** Explain Method Overloading versus Method Overriding. Detail how Java resolves calls for both at compile-time and runtime. [5 Marks]

- Q4.** Elucidate the concepts of Up-Casting and Down-Casting in an inheritance tree. Include code examples and explicit safety rules. [5 Marks]
- Q5.** What is an Interface? Discuss how interfaces enable multiple inheritance designs and loose coupling frameworks in Java. [5 Marks]
- Q6.** Explain the 'super' keyword. Detail its distinct usages in calling parent variables, methods, and parent constructor structures. [5 Marks]
- Q7.** Explain Auto-Boxing and Auto-Unboxing. Detail how they function behind the scenes with primitive types and wrapper classes. [5 Marks]
- Q8.** Discuss Enumerations (enums) in Java. Provide a code example demonstrating how enums work with fields, constructors, and loops. [5 Marks]
- Q9.** Explain Java Packages. Provide the exact directory structure and code syntax for creating and compiling custom packages. [5 Marks]
- Q10.** Create a fully working Java program illustrating access modifiers (private, default, protected, public) across different packages. [5 Marks]

8 Marks Comprehensive Questions

- Q1.** Design a comprehensive OOP architecture illustrating Polymorphism, dynamic method dispatch, overriding restrictions, and runtime binding. [8 Marks]
- Q2.** Analyze the structural differences between Abstract Classes and Interfaces in Java. Write a detailed program reflecting real-world use. [8 Marks]
- Q3.** Develop an entire modular Java application involving custom package design, inheritance trees, constructor chains, and aggregation models. [8 Marks]
- Q4.** Deconstruct the complete concept of Inheritance in Java. Detail all types, structural restrictions, 'final' class blocks, and type conversions. [8 Marks]
- Q5.** Write a fully functional inventory simulation program demonstrating object parameters, method overloading, custom enums, and static blocks. [8 Marks]

Unit-III: Arrays, Strings, Advanced I/O Package & File Compression

1 Mark Short Questions

- Q1.** How is a 1D array allocated memory dynamically using the 'new' keyword in Java? [1 Mark]
- Q2.** What is a Jagged Array in Java? [1 Mark]
- Q3.** State the default index value of the first element in any Java array structure. [1 Mark]
- Q4.** How can you determine the total length of an array dynamically at runtime? [1 Mark]

- Q5.** Can you create an array that holds object reference instances instead of primitives? [1 Mark]
- Q6.** Which class represents immutable text sequences in the standard java.lang package? [1 Mark]
- Q7.** What does 'String Immutability' mean in Java? [1 Mark]
- Q8.** Which method is used to check for structural character equality between two String objects? [1 Mark]
- Q9.** Name the class used for creating mutable string sequences in a thread-safe environment. [1 Mark]
- Q10.** What is the core structural difference between StringBuffer and StringBuilder? [1 Mark]
- Q11.** Which package contains the fundamental Stream classes for executing system I/O operations? [1 Mark]
- Q12.** Name the specific class used to query metadata properties of a file or directory paths. [1 Mark]
- Q13.** What is the difference between a Byte Stream and a Character Stream base class? [1 Mark]
- Q14.** Name the base abstract class for all character-oriented input streams. [1 Mark]
- Q15.** Which class is used to write raw bytes directly into an external file target? [1 Mark]
- Q16.** What is the purpose of the InputStreamReader class layer? [1 Mark]
- Q17.** Which class provides formatted printing capabilities like print() and println() for text writing? [1 Mark]
- Q18.** Name the stream class often wrapped around a FileWriter to output structured text cleanly. [1 Mark]
- Q19.** Which Java package contains classes like ZipOutputStream for file compression tasks? [1 Mark]
- Q20.** What is the main advantage of compressing files programmatically within enterprise applications? [1 Mark]

2 Marks Concept Questions

- Q1.** Explain how 2D arrays are structurally organized in Java's memory allocation scheme. [2 Marks]
- Q2.** What is an Array of Objects? Show the declaration syntax for an array of a custom class 'Student'. [2 Marks]
- Q3.** Explain how the String Constant Pool works to save heap memory spaces. [2 Marks]
- Q4.** Compare the functional outcomes of using '==' versus the '.equals()' method on two string references. [2 Marks]
- Q5.** Describe the process of passing a String object into a method block and returning a new modified string. [2 Marks]
- Q6.** When should a developer prefer StringBuilder over standard String concatenation inside loops? [2 Marks]
- Q7.** What is a Stream in Java? Differentiate between Input Streams and Output Streams. [2 Marks]
- Q8.** List four useful metadata inspection methods provided inside the java.io.File class structure. [2 Marks]
- Q9.** Explain how FileInputStream reads raw data byte-by-byte from an external disk source. [2 Marks]

- Q10.** Differentiate between the fundamental purposes of `FileReader` and `FileInputStream`. [2 Marks]
- Q11.** Why is it a critical programming practice to invoke the `close()` method on I/O stream streams? [2 Marks]
- Q12.** Explain the role of `PrintStream`. How does it interact with the standard `System.out` channel? [2 Marks]
- Q13.** What is the main benefit of utilizing `BufferedWriter` over standard raw `FileWriter` streams? [2 Marks]
- Q14.** Name two core classes from `java.util.zip` used for compressing and uncompressing file inputs. [2 Marks]
- Q15.** What exception class must be handled when performing operations using the `java.io` package? [2 Marks]

5 Marks Analytical Questions

- Q1.** Explain Jagged Arrays in Java. Provide a clear code example showing declaration, initialization, and nested loop traversal. [5 Marks]
- Q2.** Discuss the internal memory allocation mechanics of String objects. Contrast Heap allocations with String Constant Pool storage. [5 Marks]
- Q3.** Compare String, `StringBuffer`, and `StringBuilder` classes based on performance, mutability, storage mechanisms, and thread safety. [5 Marks]
- Q4.** Explain the `java.io` Stream hierarchy. Differentiate between Byte Streams and Character Streams with their main abstract classes. [5 Marks]
- Q5.** Write an executable Java program that creates an Array of Objects, populates data via loops, and calls object methods dynamically. [5 Marks]
- Q6.** Demonstrate how to read data from one text file and write it to another file using `FileReader` and `FileWriter` character blocks. [5 Marks]
- Q7.** Explain the `File` class in Java. Provide an operational script showing file creation, deletion, checking paths, and listing contents. [5 Marks]
- Q8.** Develop an operational Java snippet utilizing `FileInputStream` and `FileOutputStream` to create a precise binary copy of an image file. [5 Marks]
- Q9.** Elucidate the concepts of `InputStreamReader` and `OutputStreamWriter`. Explain how they bridge byte channels into character streams. [5 Marks]
- Q10.** Write a complete, compiled Java program that compresses a single target text file into a standard compressed ZIP file format. [5 Marks]

8 Marks Comprehensive Questions

- Q1.** Develop a robust file utility class in Java that handles file generation, reading, formatted printing, and robust compression routines. [8 Marks]

- Q2.** Analyze the complete architecture of multi-dimensional and jagged arrays in Java. Code an algorithm to process complex tabular records. [8 Marks]
- Q3.** Provide a comprehensive analysis of String handling in Java. Cover pool mechanics, mutability issues, and stream performance optimizations. [8 Marks]
- Q4.** Design and implement a complete object serialization simulation that reads and writes complex user records using deep I/O channels. [8 Marks]
- Q5.** Construct an enterprise-ready program that monitors a directory, processes text using character buffers, and handles file archives. [8 Marks]

Unit-IV: Exception Handling, Multithreading, Networking & JDBC

1 Mark Short Questions

- Q1.** What is an Exception in Java programming? [1 Mark]
- Q2.** Name the root superclass for all exception and error types in the java.lang package. [1 Mark]
- Q3.** What is an Uncaught Exception? [1 Mark]
- Q4.** Which keyword is used to explicitly throw a specific exception instance from code blocks? [1 Mark]
- Q5.** What is the function of the 'throws' keyword in a method signature definition? [1 Mark]
- Q6.** Can you create a custom exception class? (Yes/No) [1 Mark]
- Q7.** Define Multithreading in a brief, concise sentence. [1 Mark]
- Q8.** Name the two native ways to build or create a thread structure in Java. [1 Mark]
- Q9.** Which method must be overridden when implementing the Runnable interface framework? [1 Mark]
- Q10.** What method is called to transition a newly created Thread into the runnable execution state? [1 Mark]
- Q11.** What is thread prioritization? [1 Mark]
- Q12.** Which keyword enforces mutual exclusion across concurrent threads accessing a block or method? [1 Mark]
- Q13.** Name a method used for inter-thread communication to make a thread yield its execution slot. [1 Mark]
- Q14.** Which core package provides standard classes for executing networking and socket tasks? [1 Mark]
- Q15.** What does TCP/IP stand for in networking paradigms? [1 Mark]
- Q16.** What is Datagram programming in connectionless networks? [1 Mark]
- Q17.** What is the full form of JDBC in database system connection architecture? [1 Mark]
- Q18.** Which method of the DriverManager class is invoked to establish a live connection to a database? [1 Mark]

Q19. What interface is used to run standard static SQL queries against an active database system? *[1 Mark]*

Q20. Which object type holds the tabular data results returned by an executed SQL SELECT statement? *[1 Mark]*

2 Marks Concept Questions

Q1. Differentiate between Checked Exceptions and Unchecked (Runtime) Exceptions. *[2 Marks]*

Q2. Explain the control flow within a try-catch-finally structural layout when an exception triggers. *[2 Marks]*

Q3. Why is it vital to arrange multiple catch blocks from the most specific exception to the most general? *[2 Marks]*

Q4. How do you build a custom exception class that retains checked characteristics? *[2 Marks]*

Q5. Compare the benefits of extending the Thread class versus implementing the Runnable interface. *[2 Marks]*

Q6. Describe the main states present within the standard lifecycle of a Java Thread. *[2 Marks]*

Q7. Explain how thread synchronization prevents data race conditions in shared memory environments. *[2 Marks]*

Q8. Explain the specific functional difference between invoking the wait() method versus the sleep() method. *[2 Marks]*

Q9. What are the roles of notify() and notifyAll() in resolving Producer-Consumer thread contentions? *[2 Marks]*

Q10. Explain the basic structural differences between TCP Socket Programming and UDP Datagram Programming. *[2 Marks]*

Q11. What is the purpose of the java.net.URL class? List any one of its primary inspection methods. *[2 Marks]*

Q12. Explain the architectural role of a JDBC Driver in connecting Java apps to data stores. *[2 Marks]*

Q13. List the core steps required to query a relational database utilizing standard JDBC interfaces. *[2 Marks]*

Q14. What is the difference between Statement and PreparedStatement interfaces regarding security? *[2 Marks]*

Q15. How does the SQLException class help diagnose runtime issues during database transactions? *[2 Marks]*

5 Marks Analytical Questions

Q1. Explain Exception Handling in Java. Detail the roles of try, catch, finally, throw, and throws keywords with clean code. *[5 Marks]*

- Q2.** Discuss the process of creating custom exceptions. Write a full program showing a `UserDefinedException` triggered by logic rules. [5 Marks]
- Q3.** Explain the Thread Lifecycle in Java. Draw or describe all states (New, Runnable, Blocked, Waiting, `Timed_Waiting`, Terminated). [5 Marks]
- Q4.** Demonstrate how to implement a thread pool paradigm by creating multiple threads via the `Runnable` interface with loop logs. [5 Marks]
- Q5.** Explain Thread Synchronization. Provide a code example showing a shared bank account class using synchronized methods. [5 Marks]
- Q6.** Discuss Inter-Thread Communication. Explain `wait()`, `notify()`, and `notifyAll()` by detailing a clear Producer-Consumer structure. [5 Marks]
- Q7.** Write a fully functional Java Client-Server application using `ServerSocket` and `Socket` classes to exchange simple text messages. [5 Marks]
- Q8.** Explain Datagram programming in Java. Detail how `DatagramPacket` and `DatagramSocket` are configured for connectionless systems. [5 Marks]
- Q9.** Provide a complete architectural analysis of JDBC components: `DriverManager`, `Connection`, `Statement`, `PreparedStatement`, and `ResultSet`. [5 Marks]
- Q10.** Write a complete JDBC program that connects to a local database, inserts a record using `PreparedStatement`, and iterates a `ResultSet`. [5 Marks]

8 Marks Comprehensive Questions

- Q1.** Design and implement a highly robust, multi-threaded server application that logs network connections directly into a secure JDBC database. [8 Marks]
- Q2.** Analyze Java's Exception Handling architecture. Code a comprehensive enterprise validation engine with custom nested structures. [8 Marks]
- Q3.** Construct an advanced multi-threaded coordination program that models a high-throughput transaction environment using synchronization blocks. [8 Marks]
- Q4.** Deconstruct the entire `java.net` package capability. Write a full-scale network client framework that fetches web streams using URLs. [8 Marks]
- Q5.** Develop an end-to-end database management system in Java containing a menu-driven program for full CRUD operations via JDBC. [8 Marks]