

Java Question Bank

Unit-1: Introduction to Java

History, Features, Buzzwords

1. What is Java?
2. Explain the history of Java OR Explain the history of Java and its evolution from Oak.
3. Explain features of Java OR Describe the major features of Java that make it platform-independent.
4. Explain Java buzzwords OR List any five buzzwords of Java.
5. Explain common misconceptions about Java.
6. Compare Java with other programming languages (like C/C++).
7. Justify why Java is considered secure and robust.

Installation, Jargons, Architecture

8. Write steps to install JDK.
9. Explain Java jargons.
10. Define JVM, JDK, and JRE OR Differentiate between JVM, JDK, and JRE.
11. Illustrate Java program execution with a neat diagram of Java architecture OR Design a flow diagram showing Java compilation and execution process.

Bytecode & OOP

12. What is bytecode in Java? OR Explain how bytecode enables portability in Java.
13. State the four main properties of OOP.
14. Describe abstraction and encapsulation with examples.
15. Argue why OOP makes Java more flexible than procedural languages.
16. Develop a Java class implementing all four OOP principles.

Java Program Structure, Keywords & Identifiers

17. Explain general structure of Java program with block diagram OR Explain Java program structure with an example.
18. What are Java keywords? Give examples.
19. State the rules for identifiers in Java.
20. Identify valid and invalid identifiers in Java: class1, \$name, int, _temp.
21. What are Java Class Libraries? Describe their role and importance.
22. Assess the importance of class libraries for Java developers.
23. Create a Java program using at least two different packages from Java Class Libraries.

First Program & Syntax

24. Write a program to display "Hello Java" and explain each line.
25. Write a simple Java program to calculate the sum of two numbers.
26. Write a program to print your name and roll number.
27. What is syntax error? OR Identify and correct the errors in the given Java program snippet.
28. Explain the role of main() method in Java.
29. Evaluate common mistakes beginners make in Java syntax.
30. Develop a simple program and deliberately include at least 3 syntax errors, then fix them.

Data Types, Literals, Variables, Constants

31. Explain Java data types in detail OR List all primitive data types in Java.
32. Define literals with examples OR Compare octal, hexadecimal, and binary literals in Java.
33. What are variables? Differentiate between local, instance, and static variables.
34. Explain scope and lifetime of variables with a diagram.
35. What are constants in Java? Write a program to demonstrate use of constants using final.
36. Assess why strong data typing in Java is important.
37. Develop a program that demonstrates use of all primitive data types.

Operators & Type Conversions

38. What is operator? Explain types of operators in Java OR Define arithmetic, relational, and logical operators.
39. Explain short-circuit logical operators with examples.
40. Explain conditional operator (ternary operator) with an example.
41. Explain unary operator with an example.
42. Explain type conversions in detail OR evaluate the difference between implicit type conversion and explicit casting.

Code Blocks & Formatting

43. What is a code block in Java?
44. Explain why indentation is important in Java.
45. Examine the effect of missing semicolon in a Java program.
46. Write a Java program to find the largest of three numbers.
47. Explain how to generate formatted output in Java OR Explain System.out.printf() in detail.

Strings & Arrays

48. Explain String API in detail.
49. Explain arrays in Java. What are types of arrays? Explain with examples.

Big Numbers

50. What is big number in Java? Explain it in detail OR Explain Big number in Java.

Control Structures & Loops

51. Explain control structures in Java.
52. Explain decision structures in Java.
53. Explain looping structures in Java OR Explain entry-controlled and exit-controlled loops with examples.
54. Compare for, while, and do-while loops.
55. Judge which loop is best suited for menu-driven programs.
56. Explain difference between break and continue OR State the use of break and continue.
57. Describe the working of switch statement.
58. Create a program to display Fibonacci series using a loop.

Input/Output

59. What is the Scanner class in Java?
60. Explain commonly used methods of Scanner with examples.
61. Write a program to input name, age, and marks using Scanner and display them.
62. Develop a Java program using Scanner that calculates the area of a rectangle and circle.
63. Differentiate between Scanner and BufferedReader.
64. Assess the limitations of Scanner for user input.

Questions for References:

1. List and explain the primitive data types in Java.
2. Define literals with examples (decimal, octal, hexadecimal, binary, string, character).
3. What are Java keywords? Give examples.
4. Define JVM, JDK, and JRE.
6. Explain the history of Java and its evolution from Oak.
7. Explain the major features of Java that make it platform-independent and secure.
8. Explain Java buzzwords with examples.
9. Describe the role of Java Class Libraries (APIs) in programming.
10. Explain entry-controlled and exit-controlled loops with suitable examples.
16. Compare Java with other programming languages like C/C++.
17. Differentiate between local, instance, and static variables with examples.
18. Compare for, while, and do-while loops with examples.

19. Differentiate between break and continue with examples.
20. Evaluate the difference between implicit type conversion and explicit type casting in Java.
21. Justify why Java is considered secure and robust.
22. Assess the importance of indentation and formatting in Java programs.
23. Assess the limitations of Scanner class for user input.
24. Develop a Java class implementing all four OOP principles (abstraction, encapsulation, inheritance, polymorphism).
25. Create a Java program that demonstrates use of operators (arithmetic, relational, logical, ternary, and shorthand assignments).

Unit-2: Objects and Classes, Inheritance, Interface

1. What is Class and Object explain difference between it
 2. What is Encapsulation? Explain Benefits of Encapsulation
 3. Explain use of Final keyword in detail.
 4. What is Overloading and Overriding Explain with example?
 5. Explain object Initialization Blocks and static block in detail.
 6. Define a class and an object in Java.
 7. What is the general form of a class in Java?
 8. List and explain Java's access modifiers (private, protected, public, default).
 9. What are constructors in Java? Differentiate between default and parameterized constructors.
- OR**
10. What is constructor? Explain Type of constructor.
 11. What are static fields and static methods in Java?
- OR**
12. Explain static field and static method in detail
 13. Explain how objects are created in Java using the new operator.
 14. Explain mutator (setter) and accessor (getter) methods with examples.
 15. Explain constructor overloading with an example.
 16. Describe the purpose of the this and super keywords in Java.
 17. Explain garbage collection in Java. Why is it needed?
 18. Compare method overloading and method overriding with suitable examples.
 19. Differentiate between instance variables and reference variables in Java.
 20. Analyse the difference between static and non-static methods.
 21. Compare this and super keyword usage in Java with examples.
 22. Differentiate between nested classes and inner classes in Java.
 23. Evaluate the advantages of using access modifiers for data security.
 24. Assess the role of constructors in object initialization compared to setter methods.
 25. Judge the importance of recursion in Java compared to iterative solutions.
 26. Design a Java class BankAccount with methods to deposit, withdraw, and display balance using access modifiers properly.
 27. Develop a Java program demonstrating the use of static fields, static blocks, and static methods.