UNIT-I: Java Basics & OOP Concepts

1. What is Object-Oriented Programming?

OOP is a programming paradigm based on the concept of "objects", which contain data and methods.

2. What are the basic principles of OOP?

Encapsulation, Inheritance, Polymorphism, and Abstraction.

3. Difference between Procedure-Oriented and Object-Oriented Programming?

POP focuses on functions, OOP on objects. OOP offers better modularity and reuse.

4. What are Java buzzwords?

Simple, Secure, Portable, Object-Oriented, Robust, Multithreaded, Architecture-neutral, High performance, Interpreted, Distributed, Dynamic.

5. What is bytecode in Java?

Bytecode is an intermediate code generated by the Java compiler, executed by the JVM.

6. What is a Java variable?

A memory location that holds a value during program execution.

7. What are the data types in Java?

Primitive (int, float, boolean, char, etc.) and Reference (arrays, objects, etc.)

8. What are control statements?

They control the flow of execution: if, switch, for, while, do-while.

9. What is type casting?

Converting one data type to another. Explicit (manual) and Implicit (automatic).

10. What are arrays?

A container object that holds a fixed number of elements of the same type.

11. How do you pass command-line arguments in Java?

They are passed as an array of String to the main(String[] args) method.

UNIT-II: Classes, Objects, Inheritance

12. What is a class?

A blueprint to create objects, defining fields and methods.

13. What is an object?

An instance of a class.

14. What is a constructor?

A special method used to initialize objects.

15. What is the difference between default and parameterized constructors?

Default has no arguments. Parameterized accepts arguments for initialization.

16. What is the this keyword?

Refers to the current object instance.

17. What is garbage collection?

Automatic memory management that removes unused objects.

18. What is method overloading?

Multiple methods with the same name but different parameter lists.

19. What is constructor overloading?

Defining multiple constructors with different parameters.

20. Can you return an object from a method?

Yes, using the return keyword.

21. What is the difference between final, static, and this?

final: Constant

static: Belongs to class

• this: Refers to current object

22. Difference between String and StringBuffer?

• String: Immutable

• StringBuffer: Mutable and thread-safe

23. What is inheritance?

Mechanism where one class acquires properties of another class.

24. Types of inheritance in Java?

Single, Multilevel, Hierarchical (Java doesn't support multiple with classes, only with interfaces).

25. Use of super keyword?

To call parent class methods, variables, and constructors.

26. What is method overriding?

Redefining a method in the child class with the same signature.

27. What is dynamic method dispatch?

Runtime polymorphism using method overriding and parent class reference.

UNIT-III: Packages & Interfaces

28. What is a package in Java?

A namespace for classes to avoid name conflicts and control access.

29. How to import a package?

Using import package_name.*;

30. What is access protection in Java?

Controls visibility using access modifiers: private, default, protected, public.

31. What is an abstract class?

A class with abstract (incomplete) and non-abstract methods.

32. What is an interface?

A collection of abstract methods and constants.

33. Can a class implement multiple interfaces?

Yes, Java supports multiple interface implementation.

34. Difference between abstract class and interface?

- Abstract class: can have method definitions
- Interface: only method declarations (till Java 7), Java 8+ allows default methods.

UNIT-IV: GUI Programming & Event Handling

35. What is Swing?

A GUI toolkit in Java for creating windows-based applications.

36. Difference between AWT and Swing?

Swing is lightweight, platform-independent, and has more powerful components.

37. What is a layout manager?

Manages the layout of components (FlowLayout, BorderLayout, GridLayout, etc.)

38. What is MVC in Swing?

Model-View-Controller separates data (model), UI (view), and logic (controller).

39. What is a container in Swing?

A component that can hold other components like JPanel, JFrame.

40. What is an event in Java?

An action like a button click that is handled using event listeners.

41. What is a listener in Java?

Interface that receives and handles events.

42. Name any 3 event listener interfaces.

ActionListener, MouseListener, KeyListener.

✓ UNIT-V: java.lang, java.util, java.io, java.net

43. What are wrapper classes?

Classes that convert primitive types into objects (e.g., Integer, Float).

44. What is the use of Math class?

Provides math functions like sqrt(), abs(), pow().

45. What does the Runtime class do?

Interacts with the Java runtime environment (e.g., execute system commands).

46. What is ArrayList?

A resizable array implementation of List.

47. Difference between HashSet and HashMap?

- HashSet stores only unique elements.
- HashMap stores key-value pairs.

48. What is StringTokenizer?

Splits a string into tokens based on delimiters.

49. What is a File class used for?

To create, delete, and manipulate files and directories.

50. How to create a client-server program in Java?

Using Socket for client and ServerSocket for server communication.

Here are another 50 viva questions covering:

- Java history, versions, JDK/JRE/JVM
- Evolution of Java
- Deep topics from all your lab syllabus units

🔽 1–10: Java History, Versions, JVM, JDK, JRE

1. Who developed Java and when?

Java was developed by James Gosling at Sun Microsystems in 1995.

2. What was Java initially called?

Java was initially called Oak.

3. Why was Java renamed from Oak?

Because Oak was already a trademarked name.

4. What are the major Java versions?

Java 1.0, 1.2, 5 (Tiger), 6 (Mustang), 7 (Dolphin), 8, 11 (LTS), 17 (LTS), 21 (LTS)

5. What is JVM?

Java Virtual Machine runs Java bytecode and provides platform independence.

6. What is JRE?

Java Runtime Environment contains JVM and libraries needed to run Java programs.

7. What is JDK?

Java Development Kit = JRE + Development tools (compiler, debugger, etc.)

8. Difference between JDK, JRE, and JVM?

o JVM: Runs bytecode

o JRE: JVM + libraries

o **JDK**: JRE + development tools

9. What is the role of javac?

javac is the Java compiler that converts .java files into .class bytecode.

10. What makes Java platform-independent?

Bytecode executed by JVM on any platform.

✓ 11–20: Java Syntax, Data Types, and Arrays

11. What are the lexical issues in Java?

Tokens: keywords, identifiers, literals, operators, separators; case-sensitive.

12. What are Java keywords?

Reserved words like class, if, new, try, etc.

13. Is Java statically or dynamically typed?

Java is **statically typed** — variable types are known at compile time.

14. What is a literal in Java?

Constant value (like 10, 'A', true, "Hello")

15. How is a 2D array declared in Java?

int[][] arr = new int[3][4];

16. Can arrays store different data types?

No, they store only one type. But Object[] can hold any reference type.

17. What is type promotion?

Automatic conversion of smaller type to larger (e.g., byte to int)

18. What is the default value of an uninitialized int array?

0

19. What is a jagged array?

An array of arrays with different row sizes: int[][] arr = new int[2][];

20. What is an enhanced for loop (for-each)?

Simplified loop to iterate arrays/collections:

for(int x : array) { ... }

21–30: Classes, Objects, Overloading, Garbage Collection

21. Can a constructor be static or final?

No. Constructors are not inherited or shared.

22. What is constructor chaining?

Calling one constructor from another using this() or super().

23. What happens if you don't define a constructor?

Java provides a default constructor automatically.

24. Can we overload main method in Java?

Yes, but only public static void main(String[] args) is used for execution.

25. What is object slicing?

Not applicable in Java; occurs in C++ when child object is assigned to parent.

26. How is memory managed in Java?

Through automatic garbage collection and heap memory allocation.

27. How to explicitly request garbage collection?

System.gc();

28. Can you create an object without a new keyword?

Yes, using reflection or cloning.

29. What is shallow and deep copy?

- Shallow: references same objects
- Deep: creates copies of referenced objects too

30. What are instance and class variables?

- o Instance: Non-static, per object
- o Class: Static, shared across all instances

31–40: Inheritance, Method Overriding, Polymorphism

31. What is multilevel inheritance?

Class $A \rightarrow B \rightarrow C$ (C inherits from B which inherits from A)

32. What is hierarchical inheritance?

One parent class, multiple child classes.

33. Can constructors be overridden?

No. Constructors are not inherited.

34. Is multiple inheritance possible in Java?

Not with classes, only via interfaces.

35. What is the use of final method?

Prevents overriding.

36. Can you override static methods?

No, static methods belong to the class, not object.

37. What is runtime polymorphism?

Method overriding resolved at runtime.

38. What is compile-time polymorphism?

Method overloading resolved at compile time.

39. Difference between method overloading and overriding?

- o Overloading: Same class, different params
- o Overriding: Subclass redefines superclass method

40. What is upcasting in Java?

Assigning subclass object to superclass reference.

41–50: Advanced APIs – Collections, IO, Networking

41. What is the difference between ArrayList and LinkedList?

- ArrayList: Fast access, slow insert/delete
- LinkedList: Fast insert/delete, slow access

42. Difference between HashMap and TreeMap?

- HashMap: Unordered
- TreeMap: Sorted by keys

43. What is the default capacity of an ArrayList?

10

44. What is a stream in Java IO?

Flow of data: input or output (byte or character-based)

45. Difference between FileReader and FileInputStream?

o FileReader: Character-based

FileInputStream: Byte-based

46. What is the use of Scanner class?

Used to take user input easily.

47. How does socket communication work in Java?

- o Server uses ServerSocket
- Client uses Socket

48. What is the use of InetAddress class?

Used to get IP and host information.

49. What is URL class used for?

Connect to web resources using HTTP, FTP, etc.

50. What is the difference between TCP and UDP in Java networking?

- TCP: Connection-oriented, reliable (Socket/ServerSocket)
- UDP: Connectionless, faster (DatagramSocket)