1. What are OOPs concepts in Java? How are they used in automation frameworks?

OOPs has 4 pillars: Encapsulation, Inheritance, Polymorphism, and Abstraction. - Encapsulation: Used in POM (locators are private, actions are public). - Inheritance: Base Test classes inherited by test classes. - Polymorphism: Method overloading/overriding (cross-browser). - Abstraction: WebDriver interface hides implementation.

2. Difference between class and object?

Class is a blueprint; Object is an instance of a class created in memory.

3. What are access modifiers in Java?

public: everywhere, protected: package + subclass, default: package only, private: within class.

4. Difference between abstract class and interface?

Abstract class: can have abstract + concrete methods, supports single inheritance. Interface: 100% abstraction (till Java 7), supports multiple inheritance.

5. What is method overloading and method overriding?

Overloading: Same method name, different parameters (compile-time polymorphism). Overriding: Subclass changes parent’s method (runtime polymorphism).

6. Difference between String, StringBuilder, and StringBuffer?

String: immutable, StringBuilder: mutable, fast, not thread-safe, StringBuffer: mutable, thread-safe, slower.

7. Why is String immutable in Java?

For security, caching, synchronization, and performance reasons.

8. Difference between == and .equals()?

== compares reference, .equals() compares content.

9. What are Wrapper classes? What is autoboxing/unboxing?

Wrapper converts primitive to object (int → Integer). Autoboxing: primitive → wrapper, Unboxing: wrapper → primitive.

10. Difference between Array and ArrayList?

Array: fixed size, can store primitives/objects. ArrayList: dynamic, stores objects, widely used in automation.

11. Difference between List, Set, and Map?

List: ordered, allows duplicates. Set: unordered, no duplicates. Map: key-value pairs.

12. Difference between HashMap and Hashtable?

HashMap: not synchronized, allows one null key. Hashtable: synchronized, no null keys/values.

13. Difference between ArrayList and LinkedList?

ArrayList: better for search, LinkedList: better for insertion/deletion.

14. What are checked and unchecked exceptions?

Checked: compile-time (IOException). Unchecked: runtime (NullPointerException).

15. Difference between throw and throws?

throw: explicitly throws exception, throws: declares exception in method signature.

16. Difference between final, finally, and finalize()?

final: constant, finally: always executes block, finalize(): called before GC destroys object.

17. Difference between sleep() and wait()?

sleep(): pauses thread, doesn’t release lock. wait(): releases lock, waits until notify.

18. Difference between static and non-static methods?

static: belongs to class, called without object. non-static: belongs to object.

19. What is super and this keyword?

this: current class reference. super: parent class reference.

20. What is polymorphism?

Ability to perform same action in different ways. Example: method overloading & overriding.

21. What is encapsulation? How is it used in Selenium?

Binding data & methods together. In POM, locators are private, public methods expose actions.

22. What is inheritance?

One class inherits another. Used in automation base classes for setup/teardown reuse.

23. What is abstraction?

Hiding implementation details, showing only functionality. Example: WebDriver interface.

24. What is synchronization in Java?

Ensures one thread accesses resource at a time. Used in parallel test execution.

25. Difference between process and thread?

Process: independent, own memory. Thread: lightweight, shares memory.

26. What are constructors in Java? Can we overload them?

Special method to initialize object. Yes, constructors can be overloaded.

27. What is a static block in Java?

Block that runs once when class is loaded. Used for static initializations.

28. What is a singleton class? Where used in automation?

Class with only one object. Used for DriverManager or ConfigReader in frameworks.

29. What is a functional interface?

Interface with exactly one abstract method. Example: Runnable, Callable.

30. Difference between Comparable and Comparator?

Comparable: natural order, one compareTo(). Comparator: custom order, multiple compare() methods.