LearnNova

MOST ASKED - 40

Java

Interview Questions

Q 1. What is Java?

Ans: Java is a high-level, object-oriented programming language developed by Sun Microsystems that enables the development of platform-independent applications.

Q 2. What are the 4 pillars of OOP concepts in Java?

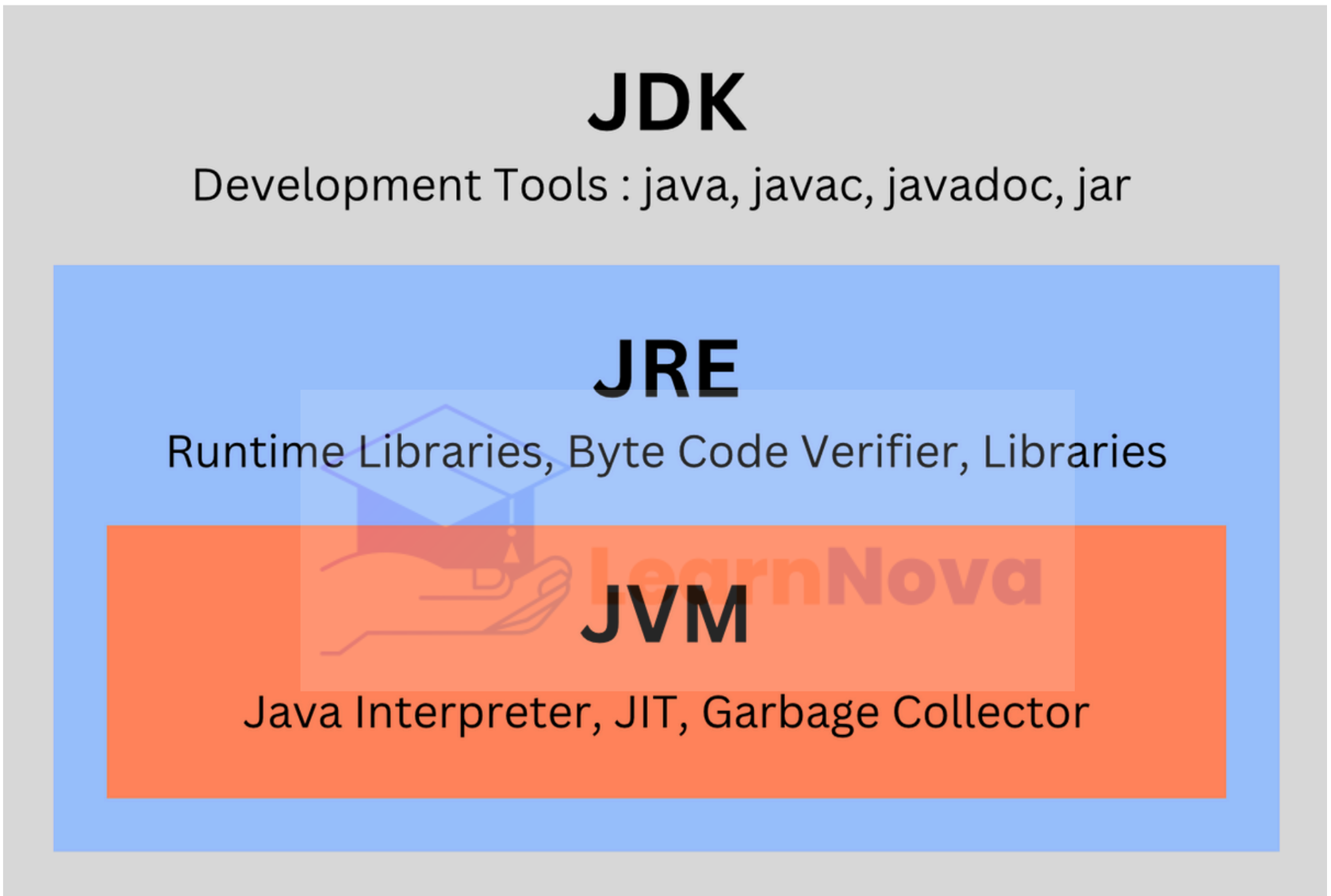
Ans: Encapsulation, Inheritance, Polymorphism, and Abstraction.

Q 3. What is the difference between JDK and JRE?

Ans: The JDK (Java Development Kit) is used by developers for creating Java applications and includes the necessary tools, libraries, and compilers. The JRE (Java Runtime Environment) is used by end-users to run Java applications and provides the runtime environment and essential class libraries, but does not include development tools.

Q 4. What are the different components of the Java platform?

Ans: It is a field (or collection of fields) in one table, that refers to the PRIMARY KEY in another table



Q 5. Reverse a String without using built-in methods

Ans:

```java
public class ReverseString {
    public static String reverse(String str) {
        String result = "";
        for (int i = str.length() - 1; i >= 0; i--)
            result += str.charAt(i);
        return result;
    }
}
```

Q 6. Check if a number is Prime

Ans:

```java
public class PrimeCheck {
    public static boolean isPrime(int n) {
        if (n <= 1) return false;
        for (int i = 2; i <= Math.sqrt(n); i++) {
            if (n % i == 0) return false;
        }
        return true;
    }
}
```


Q 7. Fibonacci Series (Iterative)?

Ans:

```java
public class Fibonacci {
    public static void printFibonacci(int n) {
        int a = 0, b = 1;
        System.out.print(a + " " + b + " ");
        for (int i = 2; i < n; i++) {
            int c = a + b;
            System.out.print(c + " ");
            a = b;
            b = c;
        }
    }
}
```

Q 8. Why java is platform independent?

Ans: Java's most unique feature is its platform independence. Unlike other languages that compile source code into platform-specific executables, Java compiles code into .class files containing bytecode. This bytecode is interpreted by the Java Virtual Machine (JVM), which is available on all major platforms. As a result, Java code compiled on Windows can run on Linux or any other platform with a JVM.

Q 9. What is difference between c++ and Java ?

Ans:

Java	C++
Java is platform independent	C++ is platform dependent.
There are no pointers in java	There are pointers in C++.
There is no operator overloading in java	C ++ has operator overloading.
There is garbage collection in java	There is no garbage collection
Supports multithreading	Does’nt support multithreading

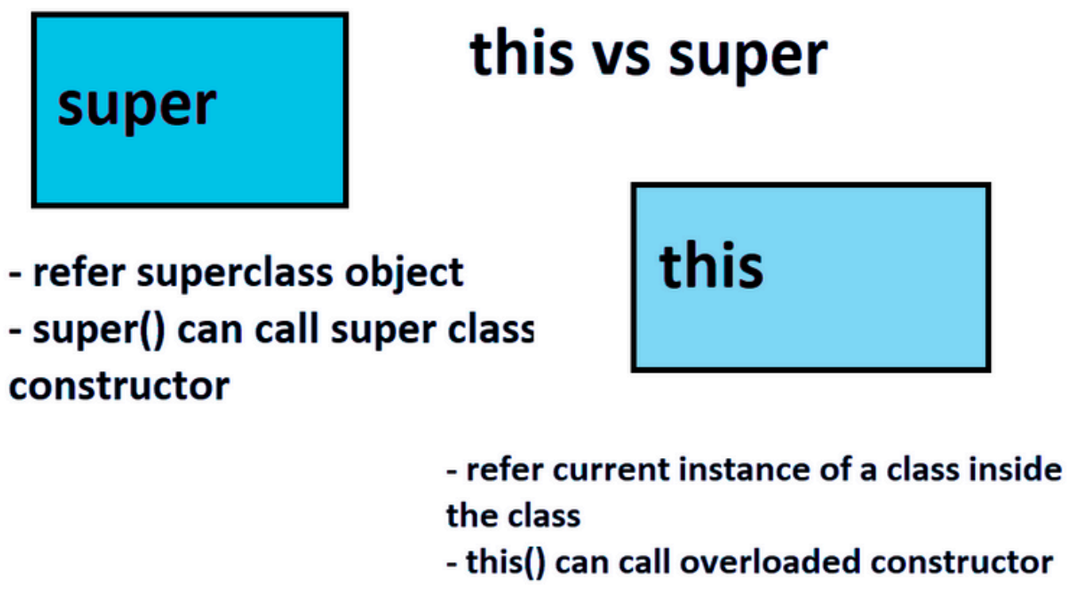
Q 10. What is bytecode in java ?

Ans: When a javac compiler compiler compiles a class it generates .class file. This .class file contains set of instructions called byte code. Byte code is a machine independent language and contains set of instructions which are to be executed only by JVM. JVM can understand this byte codes.

Q 11.Difference between this() and super() in java ?

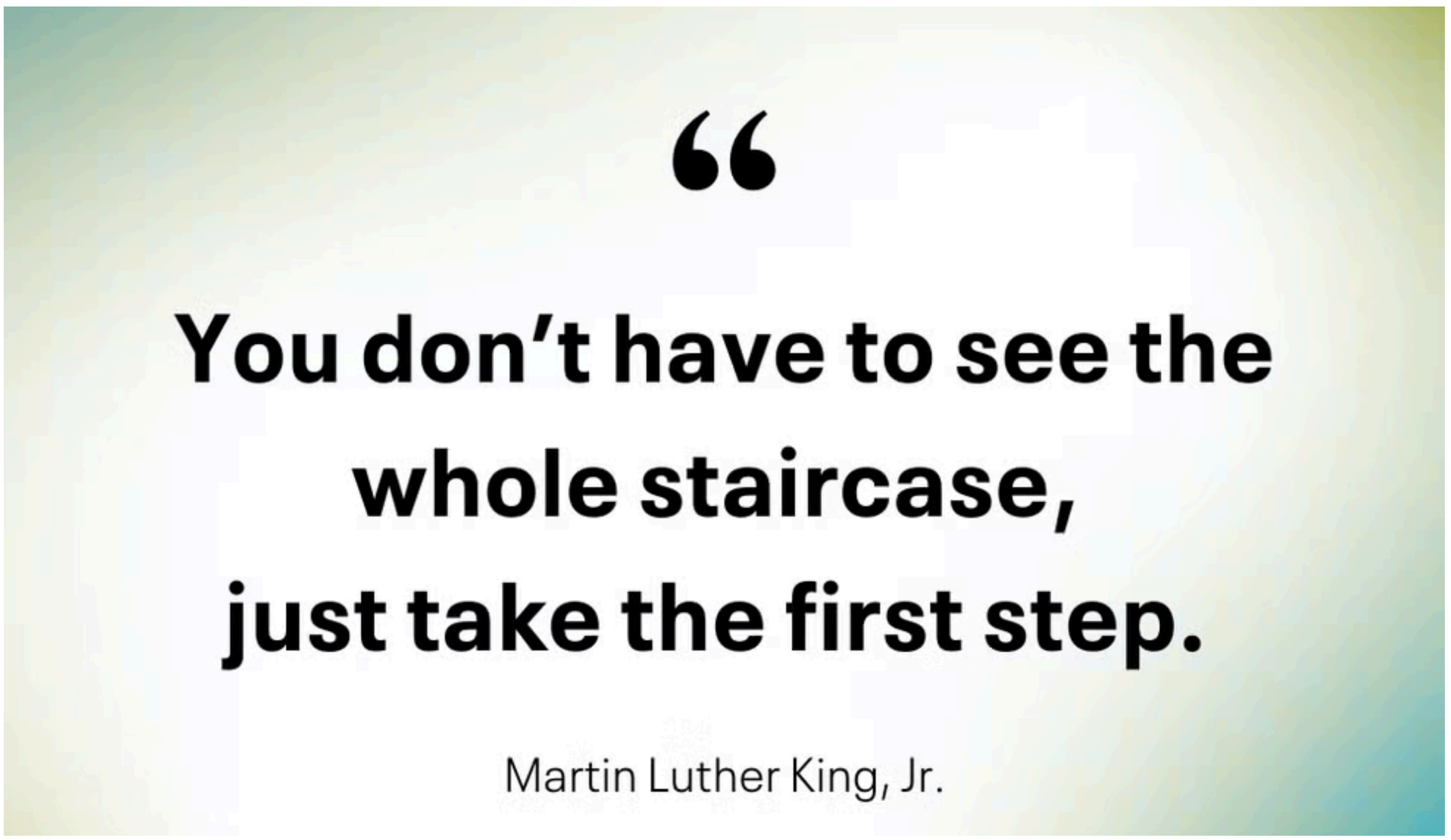
Ans: **this()** is used to access one constructor from another with in the same class while

super() is used to access superclass constructor. Either this() or super() exists it must be the first statement in the constructor.



Q 12. What is JIT compiler ?

Ans: Java JIT compiler or Just-In-Time compiler is one of the key parts of Java Runtime Environment (JRE). Its role is the compilation of bytecode to native machine code when it's in runtime. As a result, the overall performance of the Java apps is optimized.



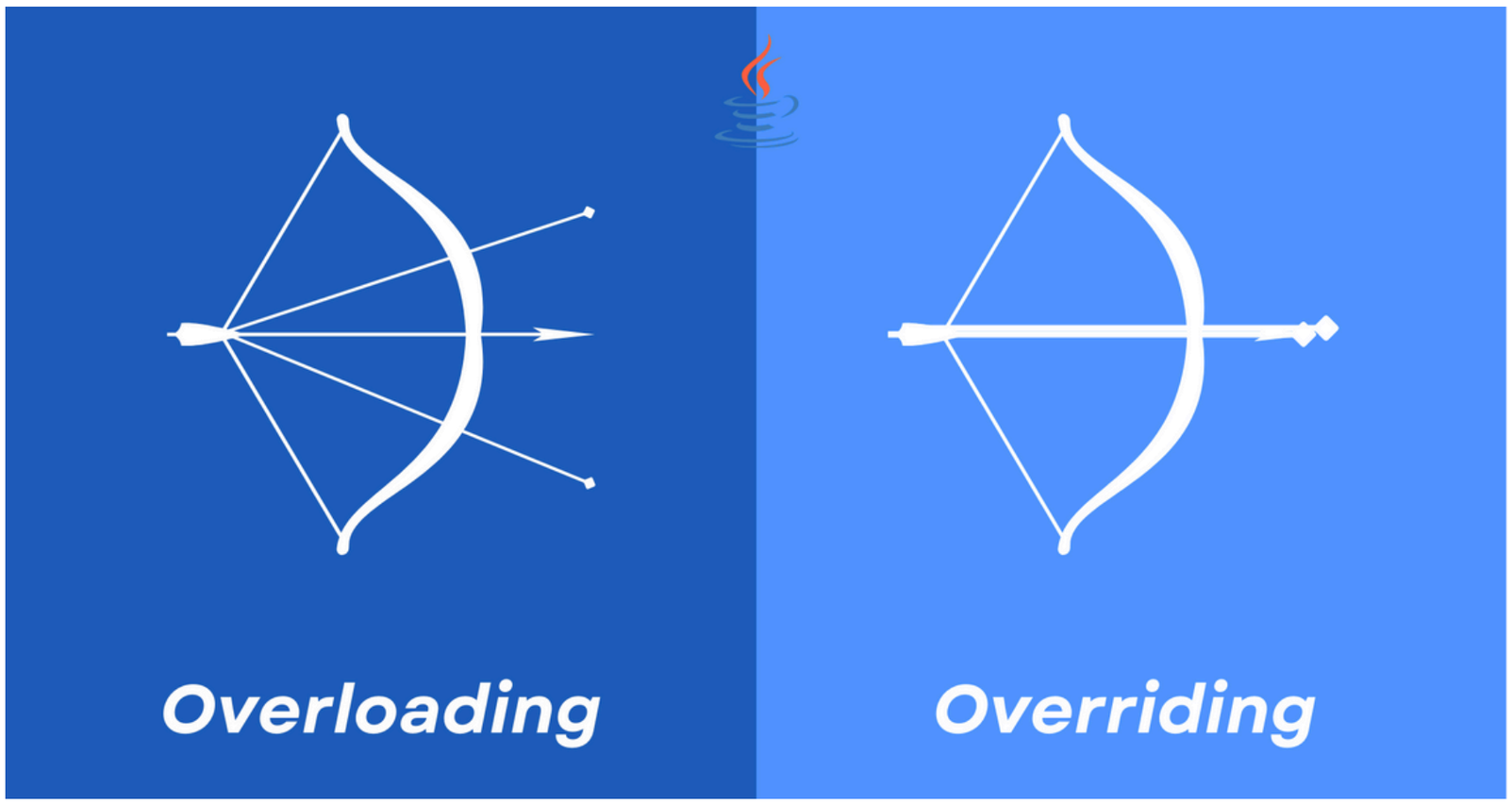
Q 13. Difference between overriding and overloading in java?

Ans: **Overriding**:

Overriding occurs when a subclass provides a specific implementation of a method that is already defined in its superclass. The method in the subclass must have the same name, return type, and parameters as the method in the superclass. It is used to achieve runtime polymorphism.

Overloading:

Overloading happens when multiple methods in the same class have the same name but different parameter lists (different type or number of parameters). It is used to increase the readability of the program and is resolved at compile time.



Q 14. What is multithreading in Java?

Ans: **Multithreading** is a process in which a Java program is divided into multiple smaller parts called threads. These threads run parallelly and simultaneously, allowing tasks to be performed concurrently. The main purpose of multithreading in Java is to create lightweight threads that share the same memory space, enabling better use of the system's processing power and improving the performance of applications.

Q 15. What is classloader?

Ans: A Class Loader in Java is a part of the Java Runtime Environment (JRE) that loads classes into memory when required during the execution of a program. It is responsible for dynamically loading, linking, and initializing classes and interfaces.

Java uses three main types of class loaders:

- Bootstrap ClassLoader
- Extension ClassLoader
- System/Application ClassLoader

Q 16. What is Java enum?

Ans: An enum (short for enumeration) in Java is a special data type that represents a group of named constant values. It was introduced in Java 5 to provide a type-safe way to define and use constants. Enums are declared using the enum keyword, and the constant values are separated by commas.

```
enum Day {
    MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY
}
```

Q 17. What is the difference between equals() method and equality (==) operator in Java?

Ans:

equals	==
it is a method	it is an operator
used to compare objects only.	used to compare primitive and object.
checks equality of constant of the object.	check equality of contant of the object and the references.

Q 18. How Lambda expressions and functional interface are interrelated?

Ans: We can call the functional interface a large platform that comes with numerous expressions. The lambda expressions are one such part of this interface. This is the interrelation between the two.

Q 19. What are the roles of final, finally, and finalize keywords in Java?

Ans:

final	finally	finalize
It is a keyword.	It is a block.	It is a method.
Used to apply restrictions on class, methods & variables.	Used to place an important code.	Used to perform clean-up processing just before the object is garbage collected.
final class can't be inherited, method can't be overridden & the variable value can't be changed.	It will be executed whether the exception is handled or not.	-

Q 20. What is the difference between a class and an object?

Ans:

- A class is a blueprint or template for creating objects.
- An object is an instance of a class, representing a real-world entity.

Q 21. What is inheritance in Java?

Ans: Inheritance is a mechanism where a child class inherits properties and methods from a parent class using the extends keyword.
It promotes code reuse.

```
class Animal {
    void eat() { System.out.println("Eating..."); }
}
class Dog extends Animal {
    void bark() { System.out.println("Barking..."); }
}
```


Q 22. What is polymorphism?

Ans: Polymorphism means many forms. It allows a single method name to behave differently based on the object.

- Compile-time polymorphism: Method overloading.
- Runtime polymorphism: Method overriding.

Q 23. What is encapsulation?

Ans: Encapsulation is the process of wrapping data and code together into a single unit (class).

It hides the data from outside access using private variables and public getters/setters.

Q 24. What is abstraction?

Ans: Abstraction is the process of hiding complex implementation and showing only essential features. It is achieved using **abstract** classes and **interfaces**.

Q 25. What is the difference between abstract class and interface?

Ans:

Abstract Class	Interface
1. *abstract* keyword	1. *interface* keyword
2. Subclasses *extends* abstract class	2. Subclasses *implements* interfaces
3. Abstract class can have implemented methods and 0 or more abstract methods	3. Java 8 onwards, Interfaces can have default and static methods
4. We can extend only one abstract class	4. We can implement multiple interfaces

Q 26. Can Java support multiple inheritance?

Ans: Java does not support multiple inheritance with classes to avoid ambiguity.

it allows multiple inheritance using interfaces.

Q 27. What is method overloading and method overriding?

Ans:

- **Overloading**: Same method name with different parameters (compile-time polymorphism).
- **Overriding**: Subclass redefines parent class method with the same signature (runtime polymorphism).

Overriding

```
class Dog{
    public void bark(){
        System.out.println("woof ");
    }
}
class Hound extends Dog{
    public void sniff(){
        System.out.println("sniff ");
    }

    public void bark(){
        System.out.println("bowl");
    }
}
```

Same Method Name, Same parameter

Overloading

```
class Dog{
    public void bark(){
        System.out.println("woof ");
    }

    //overloading method
    public void bark(int num){
        for(int i=0; i<num; i++)
            System.out.println("woof ");
    }
}
```

Same Method Name, Different Parameter

Q 28. Can we write any code after throw statement?

Ans: After throw statement jvm stop execution and subsequent statements are not executed. If we try to write any statement after throw we do get compile time error saying unreachable code.

Q 29. Can we nested try statements in java?

Ans: Yes try statements can be nested. We can declare try statements inside the block of another try statement.

Q 30. What is a constructor?

Ans: A special method used to initialize objects.

Q 31. What is the default value of the local variables?

Ans: The local variables are not initialized to any default value, neither primitives nor object references.

Q 32. What is an object?

Ans: An object in Java is a runtime instance of a class that encapsulates state and behavior. It represents a real-world entity and serves as the fundamental building block of object-oriented programming.

Q 33. What are the advantages of Packages in Java?

Ans: There are various advantages of defining packages in Java.

- o Packages avoid the name clashes.
- o The Package provides easier access control.
- o We can also have the hidden classes that are not visible outside and used by the package.
- o It is easier to locate the related classes.

Q 34. Can you make a constructor final?

Ans: No, the constructor can't be final.

Q 35. what will happen when a constructor is declared as protected?

Ans: When a constructor is protected, it can be accessed only within the same package or by subclasses (even if they are in different packages).
It restricts object creation from other classes.

Q 36. What is the static method?

Ans: A static method belongs to the class rather than an instance. It can be called without creating an object of the class and can only access other static members directly.

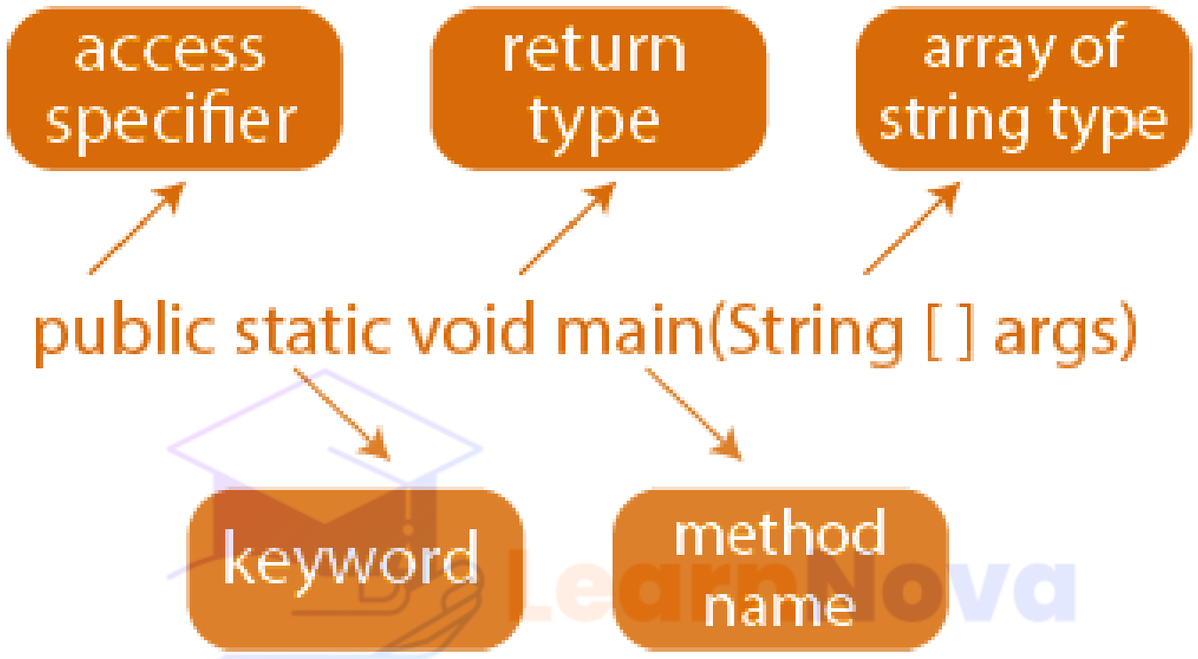
```
class Calculator {
    static int add(int a, int b) {
        return a + b;
    }

    static int subtract(int a, int b) {
        return a - b;
    }
}

public class Main {
    public static void main(String[] args) {
        int sum = Calculator.add(5, 3);
        int difference = Calculator.subtract(5, 3);
        System.out.println("Sum: " + sum);
        System.out.println("Difference: " + difference);
    }
}
```

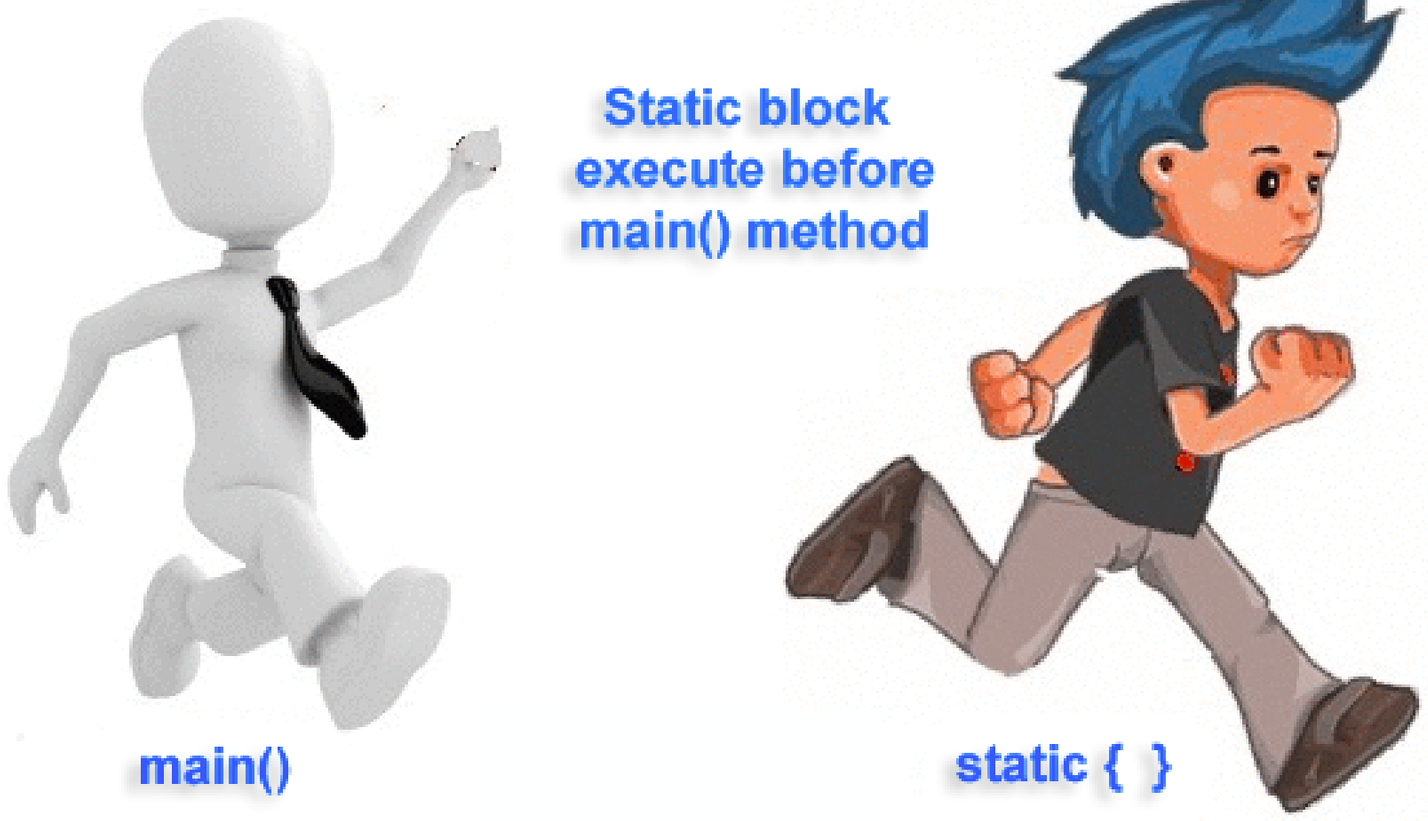
Q 37. Why is the main method static?

Ans: Because the object is not required to call the static method. If we make the main method non-static, JVM will have to create its object first and then call main() method which will lead to the extra memory allocation



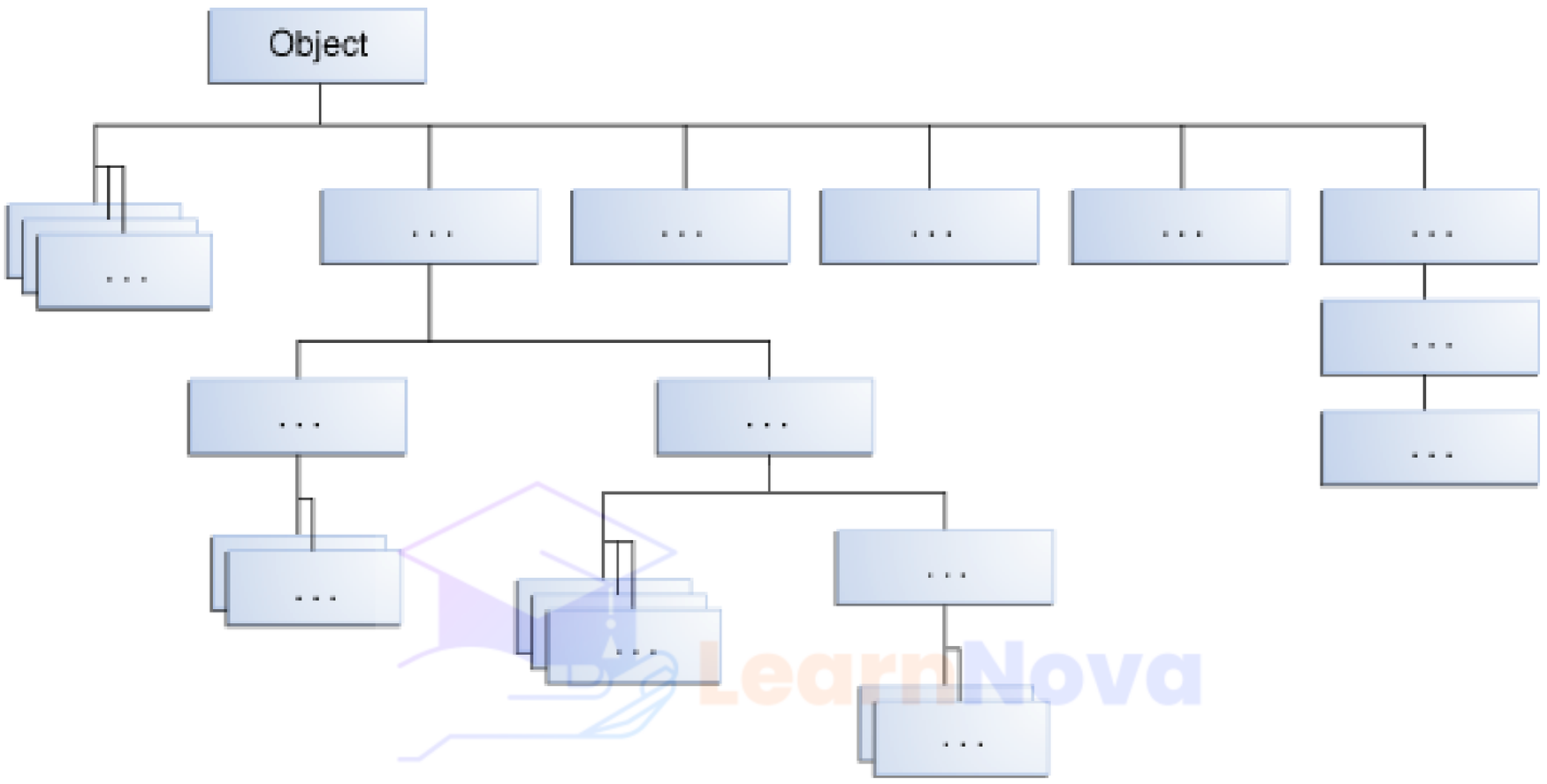
Q 38. What is the static block?

Ans: A static block is a block of code inside a class that is executed only once when the class is loaded into memory



Q 39. Which class is the superclass for all the classes?

Ans: The object class is the superclass of all other classes in Java.



Q 40. What is the difference between System.out, System.err, and System.in?

Ans:

- System.out: Regular output (e.g., program results).
- System.err: Error output (e.g., exceptions or error messages).
- System.in: Input (e.g., reading data from the user).

Our students have gone on to work at renowned companies, innovative startups, and leading unicorns.

TATA CONSULTANCY SERVICES · Infosys · wipro · HCL · Capgemini · cognizant · make my trip · ZOHO · Tech Mahindra · amazon · facebook

Explore our Programs