

## Q1. Explain JDK, JRE and JVM?

Ans:

JDK	JRE	JVM
It stands for Java Development Kit.	It stands for Java Runtime Environment.	It stands for Java Virtual Machine.
It is the tool necessary to compile, document and package Java programs.	JRE refers to a runtime environment in which Java bytecode can be executed.	It is an abstract machine. It is a specification that provides a run-time environment in which Java bytecode can be executed.
It contains JRE + development tools.	It's an implementation of the JVM which physically exists.	JVM follows three notations: Specification, Implementation, and Runtime Instance.

## Q2. Why is the Java platform independent?

Ans: Java is called platform independent because of its byte codes which can run on any system irrespective of its underlying operating system.

## Q3. Explain public static void main(String args[]) in Java.

Ans: main() in Java is the entry point for any Java program. It is always written as public static void main(String[] args).

- public: Public is an access modifier, which is used to specify who can access this method. Public means that this Method will be accessible by any Class.

- **static:** It is a keyword in java which identifies it as class-based. `main()` is made static in Java so that it can be accessed without creating the instance of a Class. In case, `main` is not made static then the compiler will throw an error as `main()` is called by the JVM before any objects are made and only static methods can be directly invoked via the class.
- **void:** It is the return type of the method. Void defines the method which will not return any value.
- **main:** It is the name of the method which is searched by JVM as a starting point for an application with a particular signature only. It is the method where the main execution occurs.
- **String args[]:** It is the parameter passed to the main method.

#### **Q4. What are wrapper classes in Java?**

Ans: Wrapper classes convert the Java primitives into the reference types (objects). Every primitive data type has a class dedicated to it. These are known as wrapper classes because they “wrap” the primitive data type into an object of that class. Refer to the below image which displays different primitive type, wrapper class and constructor argument.

#### **Q5. Why is Java not 100% Object-oriented?**

Ans: Java is not 100% Object-oriented because it makes use of eight primitive data types such as `boolean`, `byte`, `char`, `int`, `float`, `double`, `long`, `short` which are not objects.

#### **Q6. What are constructors in Java?**

Ans: In Java, constructor refers to a block of code which is used to initialize an object. It must have the same name as that of the class. Also, it has no return type and it is automatically called when an object is created.

There are two types of constructors:

1. Default Constructor:
  - a. In Java, a default constructor is the one which does not take any inputs.
  - b. In other words, default constructors are the no argument constructors which will be created by default in case you no other constructor is defined by the user.
  - c. Its main purpose is to initialize the instance variables with the default values. Also, it is majorly used for object creation.
2. Parameterized Constructor:
  - a. The parameterized constructor in Java, is the constructor which is capable of initializing the instance variables with the provided values.
  - b. In other words, the constructors which take the arguments are called **parameterized constructors**.

## Q7. What is the difference between an Array list and vector in Java?

Ans:

ArrayList	Vector
Array List is not synchronized.	Vector is synchronized.
Array List is fast as it's non-synchronized.	Vector is slow as it is thread safe.
If an element is inserted into the Array List, it increases its Array size by 50%.	Vector defaults to doubling the size of its array.
Array List does not define the increment size.	Vector defines the increment size.
Array List can only use Iterator for traversing an Array List.	Vector can use both Enumeration and Iterator for traversing.

**Q8.** What is a singleton class in Java and how can we make a class singleton?

Ans: Singleton class is a class whose only one instance can be created at any given time, in one JVM. A class can be made singleton by making its constructor private.

**Q9.** What are the differences between Heap and Stack Memory in Java?

Ans: The major difference between Heap and Stack memory are:

Features	Stack	Heap
Memory	Stack memory is used only by one thread of execution.	Heap memory is used by all the parts of the application.
Access	Stack memory can't be accessed by other threads.	Objects stored in the heap are globally accessible.
Memory Management	Follows LIFO manner to free memory.	Memory management is based on the generation associated with each object.
Lifetime	Exists until the end of execution of the thread.	Heap memory lives from the start till the end of application execution.
Usage	Stack memory only contains local primitive and reference variables to objects in heap space.	Whenever an object is created, it's always stored in the Heap space.

## Q10. What is the difference between equals() and == in Java?

Ans: Equals() method is defined in Object class in Java and used for checking equality of two objects defined by business logic.

- “==” or equality operator in Java is a binary operator provided by Java programming language and used to compare primitives and objects. public boolean equals(Object o) is the method provided by the Object class.
- The default implementation uses == operator to compare two objects.
- For example: method can be overridden like String class. equals() method is used to compare the values of two objects.

## Q11. What is a package in Java? List down various advantages of packages.

Ans: Packages in Java, are the collection of related classes and interfaces which are bundled together.

- By using packages, developers can easily modularize the code and optimize its reuse.
- Also, the code within the packages can be imported by other classes and reused.
- Below I have listed down a few of its advantages:
  - Packages help in avoiding name clashes
  - They provide easier access control on the code
  - Packages can also contain hidden classes which are not visible to the outer classes and only used within the package
  - Creates a proper hierarchical structure which makes it easier to locate the related classes

## **Q12. What is a JIT compiler in Java?**

Ans: JIT stands for Just-In-Time compiler in Java. It is a program that helps in converting the Java bytecode into instructions that are sent directly to the processor.

- By default, the JIT compiler is enabled in Java and is activated whenever a Java method is invoked.
- The JIT compiler then compiles the bytecode of the invoked method into native machine code, compiling it “just in time” to execute.
- Once the method has been compiled, the JVM summons the compiled code of that method directly rather than interpreting it.
- This is why it is often responsible for the performance optimization of Java applications at the run time.

## **Q13. Why are pointers not used in Java?**

Ans: Java doesn't use pointers because they are unsafe and increases the complexity of the program.

- Since Java is known for its simplicity of code, adding the concept of pointers will be contradicting.
- Moreover, since JVM is responsible for implicit memory allocation, thus in order to avoid direct access to memory by the user, pointers are discouraged in Java.

## **Q14. What are access modifiers in Java?**

Ans: In Java, access modifiers are special keywords which are used to restrict the access of a class, constructor, data member and method in another class. Java supports four types of access modifiers:

1. Default
2. Private
3. Protected
4. Public

Modifier	Default	Private	Protected	Public
Same class	YES	YES	YES	YES
Same Package subclass	YES	NO	YES	YES
Same Package non-subclass	YES	NO	YES	YES
Different package subclass	NO	NO	YES	YES
Different package non-subclass	NO	NO	NO	YES

## Q15. Define a Java Class.

Ans: A class in Java is a blueprint which includes all your data. A class contains fields (variables) and methods to describe the behavior of an object. Let's have a look at the syntax of a class.

```
1  class Abc {
2  member variables // class body
3  methods}
```

## Q16. What is an object in Java and how is it created?

Ans: An object is a real-world entity that has a state and behavior. An object has three characteristics:

1. State
2. Behavior
3. Identity

An object is created using the 'new' keyword. For example:

```
ClassName obj = new ClassName();
```

## **Q17. What is Object Oriented Programming?**

Ans: Object-oriented programming or popularly known as OOPs is a programming model or approach where the programs are organized around objects rather than logic and functions.

- In other words, OOP mainly focuses on the objects that are required to be manipulated instead of logic.
- This approach is ideal for the programs large and complex codes and needs to be actively updated or maintained.

## **Q18. What are the main concepts of OOPs in Java?**

Ans: Object-Oriented Programming or OOPs is a programming style that is associated with concepts like:

1. Inheritance: Inheritance is a process where one class acquires the properties of another.
2. Encapsulation: Encapsulation in Java is a mechanism of wrapping up the data and code together as a single unit.
3. Abstraction: Abstraction is the methodology of hiding the implementation details from the user and only providing the functionality to the users.
4. Polymorphism: Polymorphism is the ability of a variable, function or object to take multiple forms.

## **Q19. What is the difference between a local variable and an instance variable?**

Ans: In Java, a local variable is typically used inside a method, constructor, or a block and has only local scope. Thus, this variable can be used only within the scope of a block. The best benefit of having a local variable is that other methods in the class won't be even aware of that variable.

Example



```

1    if(x > 100)
2    {
3    String test = "abc";
4    }

```

Whereas, an instance variable in Java, is a variable which is bound to its object itself. These variables are declared within a class, but outside a method. Every object of that class will create it's own copy of the variable while using it. Thus, any changes made to the variable won't reflect in any other instances of that class and will be bound to that particular instance only.

```

1    class Test{
2    public String EmpName;
3    public int empAge;
4    }

```

## Q20. Differentiate between the constructors and methods in Java?

Ans:

Methods	Constructors
1. Used to represent the behavior of an object	1. Used to initialize the state of an object
2. Must have a return type	2. Do not have any return type
3. Needs to be invoked explicitly	3. Is invoked implicitly
4. No default method is provided by the compiler	4. A default constructor is provided by the compiler if the class has none
5. Method name may or may not be same as class name	5. Constructor name must always be the same as the class name

## Q21. What is the final keyword in Java?

Ans: Final is a special keyword in Java that is used as a non-access modifier. A final variable can be used in different contexts such as:  
final variable

- When the final keyword is used with a variable then its value can't be changed once assigned. In case the no value has been assigned to the final variable then using only the class constructor a value can be assigned to it.

final method

- When a method is declared final then it can't be overridden by the inheriting class.

final class

- When a class is declared as final in Java, it can't be extended by any subclass but it can extend another class.

## Q22. What is the difference between break and continue statements?

break	continue
1. Can be used in switch and loop (for, while, do while) statements	1. Can be only used with loop statements
2. It causes the switch or loop statements to terminate the moment it is executed	2. It doesn't terminate the loop but causes the loop to jump to the next iteration
3. It terminates the innermost enclosing loop or switch immediately	3. A continue within a loop nested with a switch will cause the next loop iteration to execute

Ans: Example break:

```

1   for (int i = 0; i < 5; i++)
2   {
3   if (i == 3)
4   {
5   break;
6   }
7   System.out.println(i);
8   }

```

Example continue:

```

1   for (int i = 0; i < 5; i++)
2   {
3   if(i == 2)
4   {
5   continue;
6   }
7   System.out.println(i);
8   }

```

## Q23. What is an infinite loop in Java? Explain with an example.

Ans: An infinite loop is an instruction sequence in Java that loops endlessly when a functional exit isn't met. This type of loop can be the result of a programming error or may also be a deliberate action based on the application behavior. An infinite loop will terminate automatically once the application exits.

For example:

```

public class InfiniteForLoopDemo
{
public static void main(String[] arg) {
for(;;)
System.out.println("Welcome!");
// To terminate this program press ctrl + c in the console.
}

```

## Q24. What is the difference between this() and super() in Java?

Ans: In Java, super() and this(), both are special keywords that are used to call the constructor.

this()	super()
1. this() represents the current instance of a class	1. super() represents the current instance of a parent/base class
2. Used to call the default constructor of the same class	2. Used to call the default constructor of the parent/base class
3. Used to access methods of the current class	3. Used to access methods of the base class
4. Used for pointing the current class instance	4. Used for pointing the superclass instance
5. Must be the first line of a block	5. Must be the first line of a block

## Q25. What is Java String Pool?

Ans: Java String pool refers to a collection of Strings which are stored in heap memory.

- In this, whenever a new object is created, String pool first checks whether the object is already present in the pool or not.
- If it is present, then the same reference is returned to the variable else a new object will be created in the String pool and the respective reference will be returned.

## Q26. Differentiate between static and non-static methods in Java.

Ans:

Static Method	Non-Static Method
1. The static keyword must be used before the method name	1. No need to use the static keyword before the method name
2. It is called using the class (className.methodName)	2. It is can be called like any general method
3. They can't access any non-static instance variables or methods	3. It can access any static method and any static variable without creating an instance of the class

## Q27. What is constructor chaining in Java?

Ans: In Java, constructor chaining is the process of calling one constructor from another with respect to the current object.

- Constructor chaining is possible only through legacy where a subclass constructor is responsible for invoking the superclass' constructor first.
- There could be any number of classes in the constructor chain.
- Constructor chaining can be achieved in two ways:
  1. Within the same class using this()
  2. From base class using super()

## Q28. Difference between String, StringBuilder, and StringBuffer.

Ans:

Factor	String	StringBuilder	StringBuffer
Storage Area	Constant String Pool	Heap Area	Heap Area
Mutability	Immutable	Mutable	Mutable
Thread Safety	Yes	No	Yes
Performance	Fast	More efficient	Less efficient

## Q29. What is a classloader in Java?

Ans: The Java ClassLoader is a subset of JVM (Java Virtual Machine) that is responsible for loading the class files. Whenever a Java program is executed it is first loaded by the classloader. Java provides three built-in classloaders:

1. Bootstrap ClassLoader
2. Extension ClassLoader
3. System/Application ClassLoader

## Q30. Why are Java Strings immutable in nature?

Ans: In Java, string objects are immutable in nature which simply means once the String object is created its state cannot be modified.

- Whenever you try to update the value of that object instead of updating the values of that particular object, Java creates a new string object. Java String objects are immutable as String objects are generally cached in the String pool.
- Since String literals are usually shared between multiple clients, action from one client might affect the rest. It enhances security, caching, synchronization, and performance of the application.

### Q31. What is the difference between an array and an array list?

Ans:

Array	ArrayList
Cannot contain values of different data types	Can contain values of different data types.
Size must be defined at the time of declaration	Size can be dynamically changed
Need to specify the index in order to add data	No need to specify the index
Arrays are not type parameterized	ArrayLists are type
Arrays can contain primitive data types as well as objects	ArrayLists can contain only objects, no primitive data types are allowed

### Q32. What is a Map in Java?

Ans: In Java, Map is an interface of Util package which maps unique keys to values. The Map interface is not a subset of the main Collection interface and thus it behaves little differently from the other collection types. Below are a few of the characteristics of Map interface:

1. Map doesn't contain duplicate keys.
2. Each key can map at max one value.

### Q33. What is a collection class in Java? List down its methods and interfaces.

Ans: In Java, the collection is a framework that acts as an architecture for storing and manipulating a group of objects. Using Collections you can

perform various tasks like searching, sorting, insertion, manipulation, deletion, etc. Java collection framework includes the following:

- Interfaces
- Classes
- Methods

### Q34. Define JDBC Drivers?

Ans: The driver is simply a Java library containing classes that implement the JDBC API. Because all JDBC drivers have to implement the same interfaces, it's not difficult to change the data source an application uses.

- For example, if an application uses an SQLite database, and then we decide later that we want to use a MySQL database, all we have to do use the MySQL JDBC driver instead of the SQLite one (in addition to migrating the data to a MySQL DB, of course).
- JDBC consists of two packages: `java.sql` (core JDBC) and `javax.sql` (optional JDBC). The APIs in the `javax.sql` package is required when working with database servers.
- The JDK ships with a database called a derby, which can be used for desktop applications or when prototyping. The derby JDBC driver is also included in the JDK.

### Q35. What is the use of garbage collection in Java?

Ans: The use of garbage collection in Java is to recognize and remove objects that are no more required by a program and are unnecessary consuming the program resources that could be freed and reused by other objects.

- An object in Java is subject to garbage collection when it turns inaccessible to the Java program in which it is initiated.



### **Q36.** What are Lambda Expressions?

Ans: Lambda Expressions, often called lambdas, are a new feature introduced with Java 8. They provide an easier way to work with interfaces that have only one method, and they are often used in places where the programmer uses anonymous classes.