1. How does Java achieve platform independence?

Java platform independency means that the Java program can run on any platform or operating system without suffering any changes. This is possible because of the bytecode.

2. What are the various access specifiers for Java classes?

Access specifiers are the keywords used before a class name that defines the access scope.

There are four types of access specifiers:

1. **Private:** When a variable is declared as private, it can be accessed only from the same class to which they belong.
2. **Public:** Methods and variables that have public access can be accessed by all the other classes in the project.
3. **Protected:** When a variable is declared as protected it means that it can be accessed within the same package classes and subclasses of any other packages.
4. **Default:** Methods and variables can be accessed only from the same package and not from outside of its native package.

3. What is the Difference Between ArrayList and HashSet in Java?

The following ones are the main differences between ArrayList and HashSet:

Implementation

ArrayList: Implements List interface.

HashSet: Implements Set inter face.

Duplicates

ArrayList: Allows duplicates values.

HashSet: Doesn’t allow duplicates values.

Q1. Explain JDK, JRE and JVM?

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.

Q3. Why Java is platform independent?

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

Q4. Why Java is not 100% Object-oriented?

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

Q5. What are wrapper classes in Java?

Wrapper classes convert the Java primitives into the reference types (objects).

Q6. What are constructors in Java?

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.

Q7. What is singleton class in Java and how can we make a class singleton?

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.

Q8. What is the difference between Array list and vector in Java?

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 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.

Q9. What is the difference between equals() and == in Java?

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.

Q10. When can you use the super keyword?

In Java, the super keyword is a reference variable that refers to an immediate parent class object.

When you create a subclass instance, you're also creating an instance of the parent class, which is referenced to by the super reference variable.

The uses of the Java super Keyword are-

1. To refer to an immediate parent class instance variable, use super.

2. The keyword super can be used to call the method of an immediate parent class.
3. Super() can be used to call the constructor of the immediate parent class.

Q11)What makes a HashSet different from a TreeSet?

HashSet is faster than TreeSet. HashSet is Implemented using a hash table. TreeSet takes O(Log n) for search, insert and delete which is higher than HashSet. But TreeSet keeps sorted data.

Q12. What are the differences between HashMap and HashTable in Java?

HashMap	Hashtable
It is non synchronized. It cannot be shared between many threads without proper synchronization code.	It is synchronized. It is thread-safe and can be shared with many threads.
It permits one null key and multiple null values.	It does not permit any null key or value.
is a new class introduced in JDK 1.2.	It was present in earlier versions of java as well.
It is faster.	It is slower.
It is traversed through the iterator.	It is traversed through Enumerator and Iterator.
It uses fail fast iterator.	It uses an enumerator which is not fail fast.
It inherits AbstractMap class.	It inherits Dictionary class.

Q26. Define a Java Class.

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.

```
class Abc {
member variables // class body
methods}
```

Q27. What is an object in Java and how is it created?

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();

Q31. Differentiate between the constructors and methods in Java?

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

In case you are facing any challenges with these Core Java interview questions, please comment on your problems in the section below.

Q32. What is final keyword in Java?

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

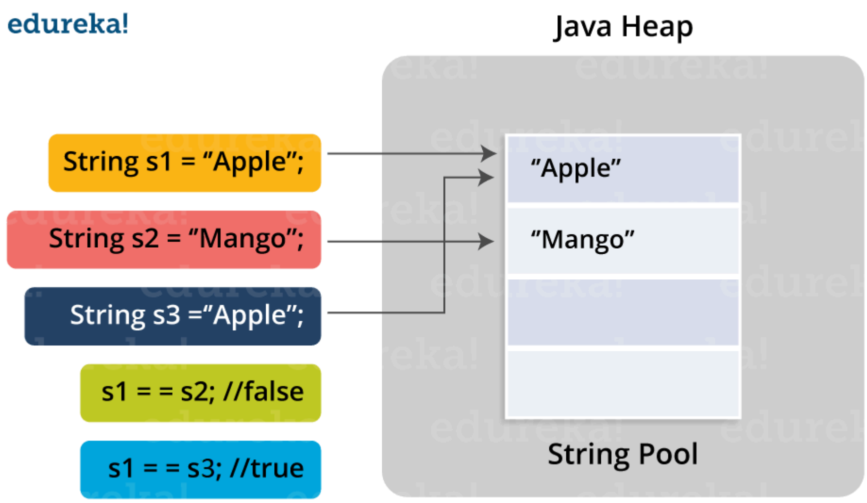
When a method is declared final then it can’t be overridden by the inheriting class.

- final class

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

Q36. What is Java String Pool?

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 new object will be created in the String pool and the respective reference will be returned.



Q39. What is constructor chaining in Java?

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()

Q40. Difference between String, StringBuilder, and StringBuffer.

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

If you think this article on Java Interview Questions is helpful, you can check out Edureka's Java Training in Chennai as well.

Q41. What is a classloader in Java?

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:

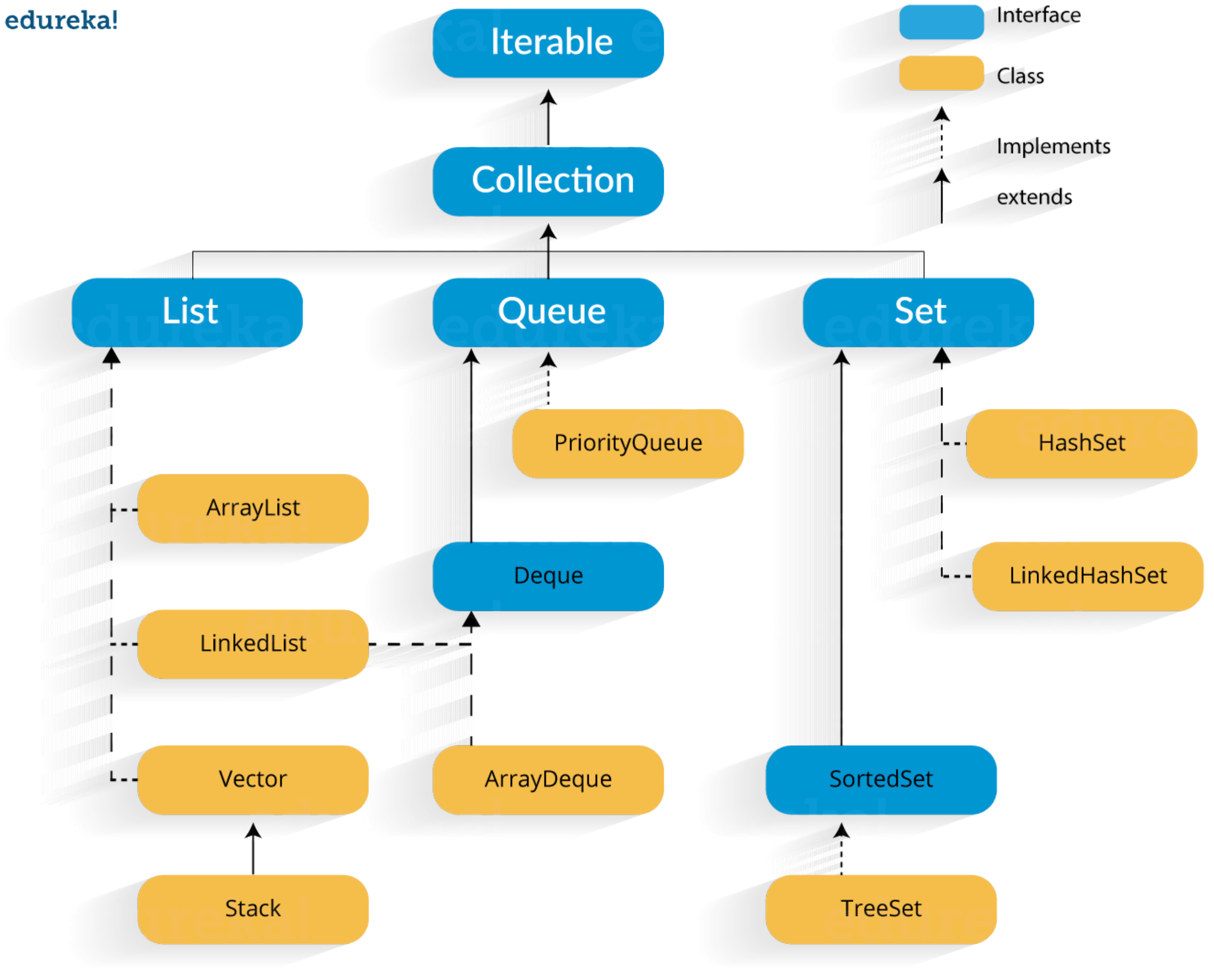
1. Bootstrap ClassLoader
2. Extension ClassLoader
3. System/Application ClassLoader

Q45. What is collection class in Java? List down its methods and interfaces.

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

The below image shows the complete hierarchy of the Java Collection.



… linked : https://www.edureka.co/blog/interview-questions/java-interview-questions/