Guru99.com

**Q1. What is the difference between an Inner Class and a Sub-Class?**

Ans: An Inner class is a class which is nested within another class. An Inner class has access rights for the class which is nesting it and it can access all variables and methods defined in the outer class.

A sub-class is a class which inherits from another class called super class. Sub-class can access all public and protected methods and fields of its super class.

**Q2. What are the various access specifiers for Java classes?**

Ans: In Java, access specifiers are the keywords used before a class name which defines the access scope. The types of access specifiers for classes are:

1. Public : Class,Method,Field is accessible from anywhere.

2. Protected:Method,Field can be accessed from the same class to which they belong or from the sub-classes,and from the class of same package,but not from outside.

3. Default: Method,Field,class can be accessed only from the same package and not from outside of it’s native package.

4. Private: Method,Field can be accessed from the same class to which they belong.

**Q3. What’s the purpose of Static methods and static variables?**

Ans: When there is a requirement to share a method or a variable between multiple objects of a class instead of creating separate copies for each object, we use static keyword to make a method or variable shared for all objects.

**Q4. What is data encapsulation and what’s its significance?**

Ans: Encapsulation is a concept in Object Oriented Programming for combining properties and methods in a single unit.

Encapsulation helps programmers to follow a modular approach for software development as each object has its own set of methods and variables and serves its functions independent of other objects. Encapsulation also serves data hiding purpose.

**Q5. What is a singleton class? Give a practical example of its usage.**

A singleton class in java can have only one instance and hence all its methods and variables belong to just one instance. Singleton class concept is useful for the situations when there is a need to limit the number of objects for a class.

The best example of singleton usage scenario is when there is a limit of having only one connection to a database due to some driver limitations or because of any licensing issues.

**Q6. What are Loops in Java? What are three types of loops?**

Ans: Looping is used in programming to execute a statement or a block of statement repeatedly. There are three [types of loops in Java](https://www.guru99.com/how-to-use-loops-in-javascript.html):

1) For Loops

For loops are used in java to execute statements repeatedly for a given number of times. For loops are used when number of times to execute the statements is known to programmer.

2) While Loops

While loop is used when certain statements need to be executed repeatedly until a condition is fulfilled. In while loops, condition is checked first before execution of statements.

3) Do While Loops

Do While Loop is same as While loop with only difference that condition is checked after execution of block of statements. Hence in case of do while loop, statements are executed at least once.

**Q7: What is an infinite Loop? How infinite loop is declared?**

Ans: An infinite loop runs without any condition and runs infinitely. An infinite loop can be broken by defining any breaking logic in the body of the statement blocks.

Infinite loop is declared as follows:

for (;;)

{

// Statements to execute

// Add any loop breaking logic

}

**Q8. What is the difference between continue and break statement?**

Ans: break and continue are two important keywords used in Loops. When a break keyword is used in a loop, loop is broken instantly while when continue keyword is used, current iteration is broken and loop continues with next iteration.

In below example, Loop is broken when counter reaches 4.

for (counter = 0; counter & lt; 10; counter++)

system.out.println(counter);

if (counter == 4) {

break;

}

}

In the below example when counter reaches 4, loop jumps to next iteration and any statements after the continue keyword are skipped for current iteration.

for (counter = 0; counter < 10; counter++)

system.out.println(counter);

if (counter == 4) {

continue;

}

system.out.println("This will not get printed when counter is 4");

}

**Q9. What is the difference between double and float variables in Java?**

Ans: In java, float takes 4 bytes in memory while Double takes 8 bytes in memory. Float is single precision floating point decimal number while Double is double precision decimal number.

**Q10. What is Final Keyword in Java? Give an example.**

Ans: In java, a constant is declared using the keyword Final. Value can be assigned only once and after assignment, value of a constant can’t be changed.

In below example, a constant with the name const\_val is declared and assigned avalue:

Private Final int const\_val=100

When a method is declared as final,it can NOT be overridden by the subclasses.This method are faster than any other method,because they are resolved at complied time.

When a class is declares as final,it cannot be subclassed. Example String,Integer and other wrapper classes.

**Q11. What is ternary operator? Give an example.**

Ans: Ternary operator , also called conditional operator is used to decide which value to assign to a variable based on a Boolean value evaluation. It’s denoted as ?

In the below example, if rank is 1, status is assigned a value of “Done” else “Pending”.

public class conditionTest {

public static void main(String args[]) {

String status;

int rank = 3;

status = (rank == 1) ? "Done" : "Pending";

System.out.println(status);

}

}

**Q12: How can you generate random numbers in Java?**

Ans:

Using Math.random() you can generate random numbers in the range greater than or equal to 0.1 and less than 1.0

Using Random class in package java.util

**Q13. What is default switch case? Give example.**

Ans: In a [switch statement](https://www.guru99.com/switch-java.html), default case is executed when no other switch condition matches. Default case is an optional case .It can be declared only once all other switch cases have been coded.

In the below example, when score is not 1 or 2, default case is used.

public class switchExample {

int score = 4;

public static void main(String args[]) {

switch (score) {

case 1:

system.out.println("Score is 1");

break;

case 2:

system.out.println("Score is 2");

break;

default:

system.out.println("Default Case");

}

}

}

**Q14. What’s the base class in Java from which all classes are derived?**

Ans: java.lang.object

**Q15. Can main() method in Java can return any data?**

Ans: In java, main() method can’t return any data and hence, it’s always declared with a void return type.

**Q16. What are Java Packages? What’s the significance of packages?**

Ans: In Java, package is a collection of classes and interfaces which are bundled together as they are related to each other. Use of packages helps developers to modularize the code and group the code for proper re-use. Once code has been packaged in Packages, it can be imported in other classes and used.

**Q17. Can we declare a class as Abstract without having any abstract method?**

Ans: Yes we can create an abstract class by using abstract keyword before class name even if it doesn’t have any abstract method. However, if a class has even one abstract method, it must be declared as abstract otherwise it will give an error.

**Q18. What’s the difference between an Abstract Class and Interface in Java?**

Ans: The primary difference between an abstract class and interface is that an interface can only possess declaration of public static methods with no concrete implementation while an abstract class can have members with any access specifiers (public, private etc) with or without concrete implementation.

Another key difference in the use of abstract classes and interfaces is that a class which implements an interface must implement all the methods of the interface while a class which inherits from an abstract class doesn’t require implementation of all the methods of its super class.

A class can implement multiple interfaces but it can extend only one abstract class.

**AD**

**Q19. What are the performance implications of Interfaces over abstract classes?**

Ans: Interfaces are slower in performance as compared to abstract classes as extra indirections are required for interfaces. Another key factor for developers to take into consideration is that any class can extend only one abstract class while a class can implement many interfaces.

Use of interfaces also puts an extra burden on the developers as any time an interface is implemented in a class; developer is forced to implement each and every method of interface.

**Q20. Does Importing a package imports its sub-packages as well in Java?**

Ans: In java, when a package is imported, its sub-packages aren’t imported and developer needs to import them separately if required.

For example, if a developer imports a package university.\*, all classes in the package named university are loaded but no classes from the sub-package are loaded. To load the classes from its sub-package ( say department), developer has to import it explicitly as follows:

Import university.department.\*

**Q21. Can we declare the main method of our class as private?**

Ans: In java, main method must be public static in order to run any application correctly. If main method is declared as private, developer won’t get any compilation error however, it will not get executed and will give a runtime error.

**Q22. How can we pass argument to a function by reference instead of pass by value?**

Ans: In java, we can pass argument to a function only by value and not by reference.

**Q23. How an object is serialized in java?**

Ans: In java, to convert an object into byte stream by serialization, an interface with the name Serializable is implemented by the class. All objects of a class implementing serializable interface get serialized and their state is saved in byte stream.

**Q24. When we should use serialization?**

Ans: Serialization is used when data needs to be transmitted over the network. Using serialization, object’s state is saved and converted into byte stream .The byte stream is transferred over the network and the object is re-created at destination.

**Q25. Is it compulsory for a Try Block to be followed by a Catch Block in Java for Exception handling?**

Ans: Try block needs to be followed by either Catch block or Finally block or both. Any exception thrown from try block needs to be either caught in the catch block or else any specific tasks to be performed before code abortion are put in the Finally block.

**Q26. Is there any way to skip Finally block of exception even if some exception occurs in the exception block?**

Ans: If an exception is raised in Try block, control passes to catch block if it exists otherwise to finally block. Finally block is always executed when an exception occurs and the only way to avoid execution of any statements in Finally block is by aborting the code forcibly by writing following line of code at the end of try block:

System.exit(0);

**Q27. When the constructor of a class is invoked?**

Ans: The constructor of a class is invoked every time an object is created with new keyword.

For example, in the following class two objects are created using new keyword and hence, constructor is invoked two times.

public class const\_example {

const\_example() {

system.out.println("Inside constructor");

}

public static void main(String args[]) {

const\_example c1 = new const\_example();

const\_example c2 = new const\_example();

}

}

**Q28. Can a class have multiple constructors?**

Ans: Yes, a class can have multiple constructors with different parameters. Which constructor gets used for object creation depends on the arguments passed while creating the objects.

**AD**

**Q29. Can we override static methods of a class?**

Ans: We cannot override static methods. Static methods belong to a class and not to individual objects and are resolved at the time of compilation (not at runtime).Even if we try to override static method,we will not get an complitaion error,nor the impact of overriding when running the code.

**Q30. In the below example, what will be the output?**

public class superclass {

public void displayResult() {

system.out.println("Printing from superclass");

}

}

public class subclass extends superclass {

public void displayResult() {

system.out.println("Displaying from subClass");

super.displayResult();

}

public static void main(String args[]) {

subclass obj = new subclass();

obj.displayResult();

}

}

**Ans**: Output will be:

Displaying from subClass

Printing from superclass

**Q31. Is String a data type in java?**

Ans: String is not a primitive data type in java. When a string is created in java, it’s actually an object of Java.Lang.String class that gets created. After creation of this string object, all built-in methods of String class can be used on the string object.

**Q32. In the below example, how many String Objects are created?**

String s1="I am Java Expert";

String s2="I am C Expert";

String s3="I am Java Expert";

Ans: In the above example, two objects of Java.Lang.String class are created. s1 and s3 are references to same object.

**Q33. Why Strings in Java are called as Immutable?**

Ans: In java, string objects are called immutable as once value has been assigned to a string, it can’t be changed and if changed, a new object is created.

In below example, reference str refers to a string object having value “Value one”.

String str="Value One";

When a new value is assigned to it, a new String object gets created and the reference is moved to the new object.

str="New Value";

**Q34. What’s the difference between an array and Vector?**

Ans: An array groups data of same primitive type and is static in nature while vectors are dynamic in nature and can hold data of different data types.

**Q35. What is multi-threading?**

Ans: Multi threading is a programming concept to run multiple tasks in a concurrent manner within a single program. Threads share same process stack and running in parallel. It helps in performance improvement of any program.

**Q36. Why Runnable Interface is used in Java?**

Ans: Runnable interface is used in java for implementing multi threaded applications. Java.Lang.Runnable interface is implemented by a class to support multi threading.

**Q37. What are the two ways of implementing multi-threading in Java?**

Ans: Multi threaded applications can be developed in Java by using any of the following two methodologies:

1. By using Java.Lang.Runnable Interface. Classes implement this interface to enable multi threading. There is a Run() method in this interface which is implemented.

2. By writing a class that extend Java.Lang.Thread class.

**Q38. When a lot of changes are required in data, which one should be a preference to be used? String or StringBuffer?**

Ans: Since StringBuffers are dynamic in nature and we can change the values of StringBuffer objects unlike String which is immutable, it’s always a good choice to use StringBuffer when data is being changed too much. If we use String in such a case, for every data change a new String object will be created which will be an extra overhead.

**Q39. What’s the purpose of using Break in each case of Switch Statement?**

Ans: Break is used after each case (except the last one) in a switch so that code breaks after the valid case and doesn’t flow in the proceeding cases too.

If break isn’t used after each case, all cases after the valid case also get executed resulting in wrong results.

**Q40. How garbage collection is done in Java?**

Ans: In java, when an object is not referenced any more, garbage collection takes place and the object is destroyed automatically. For automatic garbage collection java calls either System.gc() method or Runtime.gc() method.

### What is Garbage Collection?

Garbage collection is a process of reclaiming the unused runtime objects. It is performed for memory management. In other words, we can say that It is the process of removing unused objects from the memory to free up space and make this space available for Java Virtual Machine. Due to garbage collection java gives 0 as output to a variable whose value is not set, i.e., the variable has been defined but not initialized. For this purpose, we were using free() function in the C language and delete() in C++. In Java, it is performed automatically. So, java provides better memory management.

**Q41. How we can execute any code even before main method?**

Ans: If we want to execute any statements before even creation of objects at load time of class, we can use a static block of code in the class. Any statements inside this static block of code will get executed once at the time of loading the class even before creation of objects in the main method.

**Q42. Can a class be a super class and a sub-class at the same time? Give example.**

Ans: If there is a hierarchy of inheritance used, a class can be a super class for another class and a sub-class for another one at the same time.

In the example below, continent class is sub-class of world class and it’s super class of country class.

public class world {

..........

}

public class continenet extends world {

............

}

public class country extends continent {

......................

}

**Q43. How objects of a class are created if no constructor is defined in the class?**

Ans: Even if no explicit constructor is defined in a java class, objects get created successfully as a default constructor is implicitly used for object creation. This constructor has no parameters.

**Q44. In multi-threading how can we ensure that a resource isn’t used by multiple threads simultaneously?**

Ans: In multi-threading, access to the resources which are shared among multiple threads can be controlled by using the concept of synchronization. Using synchronized keyword, we can ensure that only one thread can use shared resource at a time and others can get control of the resource only once it has become free from the other one using it.

**Q45. Can we call the constructor of a class more than once for an object?**

Ans: Constructor is called automatically when we create an object using new keyword. It’s called only once for an object at the time of object creation and hence, we can’t invoke the constructor again for an object after its creation.

**Q46. There are two classes named classA and classB. Both classes are in the same package. Can a private member of classA can be accessed by an object of classB?**

Ans: Private members of a class aren’t accessible outside the scope of that class and any other class even in the same package can’t access them.

**Q47. Can we have two methods in a class with the same name?**

Ans: We can define two methods in a class with the same name but with different number/type of parameters. Which method is to get invoked will depend upon the parameters passed.

For example in the class below we have two print methods with same name but different parameters. Depending upon the parameters, appropriate one will be called:

public class methodExample {

public void print() {

system.out.println("Print method without parameters.");

}

public void print(String name) {

system.out.println("Print method with parameter");

}

public static void main(String args[]) {

methodExample obj1 = new methodExample();

obj1.print();

obj1.print("xx");

}

}

**Q48. How can we make copy of a java object?**

Ans: We can use the concept of cloning to create copy of an object. Using clone, we create copies with the actual state of an object.

Clone() is a method of Cloneable interface and hence, Cloneable interface needs to be implemented for making object copies.

**Q49. What’s the benefit of using inheritance?**

Ans: Key benefit of using inheritance is reusability of code as inheritance enables sub-classes to reuse the code of its super class. Polymorphism (Extensibility ) is another great benefit which allow new functionality to be introduced without effecting existing derived classes.

**Q50. What’s the default access specifier for variables and methods of a class?**

Ans: Default access specifier for variables and method is package protected i.e variables and class is available to any other class but in the same package,not outside the package.

**Q51. Give an example of use of Pointers in Java class.**

Ans: There are no pointers in Java. So we can’t use concept of pointers in Java.

**Q52. How can we restrict inheritance for a class so that no class can be inherited from it?**

Ans: If we want a class not to be extended further by any class, we can use the keyword **Final** with the class name.

In the following example, Stone class is Final and can’t be extend

public Final Class Stone {

// Class methods and Variables

}

**Q53. What’s the access scope of Protected Access specifier?**

Ans: When a method or a variable is declared with Protected access specifier, it becomes accessible in the same class,any other class of the same package as well as a sub-class.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Modifier** | **Class** | **Package** | **Subclass** | **World** |
| public | Y | Y | Y | Y |
| protected | Y | Y | Y | N |
| no modifier | Y | Y | N | N |
| private | Y | N | N | N |

**Q54. What’s difference between Stack and Queue?**

Ans: Stack and Queue both are used as placeholder for a collection of data. The primary difference between a stack and a queue is that stack is based on Last in First out (LIFO) principle while a queue is based on FIFO (First In First Out) principle.

**Q55. In java, how we can disallow serialization of variables?**

Ans: If we want certain variables of a class not to be serialized, we can use the keyword **transient**while declaring them. For example, the variable trans\_var below is a transient variable and can’t be serialized:

public class transientExample {

private transient trans\_var;

// rest of the code

}

**Q56. How can we use primitive data types as objects?**

Ans: Primitive data types like int can be handled as objects by the use of their respective wrapper classes. For example, Integer is a wrapper class for primitive data type int. We can apply different methods to a wrapper class, just like any other object.

What are wrapper classes?

Wrapper classes are classes that allow primitive types to be accessed as objects. In other words, we can say that wrapper classes are built-in java classes which allow the conversion of objects to primitives and primitives to objects. The process of converting primitives to objects is called autoboxing, and the process of converting objects to primitives is called unboxing. There are eight wrapper classes present in **java.lang** package is given below.

|  |  |
| --- | --- |
| **Primitive Type** | **Wrapper class** |
| boolean | Boolean |
| char | Character |
| byte | Byte |
| short | Short |
| int | Integer |
| long | Long |
| float | Float |
| double | Double |

219)What are autoboxing and unboxing? When does it occur?

The autoboxing is the process of converting primitive data type to the corresponding wrapper class object, eg., int to Integer. The unboxing is the process of converting wrapper class object to primitive data type. For eg., integer to int. Unboxing and autoboxing occur automatically in Java. However, we can externally convert one into another by using the methods like valueOf() or xxxValue().

It can occur whenever a wrapper class object is expected, and primitive data type is provided or vice versa.

* Adding primitive types into Collection like ArrayList in Java.
* Creating an instance of parameterized classes ,e.g., ThreadLocal which expect Type.
* Java automatically converts primitive to object whenever one is required and another is provided in the method calling.
* When a primitive type is assigned to an object type.

220) What is the output of the below Java program?

1. **public** **class** Test1
2. {
3. **public** **static** **void** main(String[] args) {
4. Integer i = **new** Integer(201);
5. Integer j = **new** Integer(201);
6. **if**(i == j)
7. {
8. System.out.println("hello");
9. }
10. **else**
11. {
12. System.out.println("bye");
13. }
14. }
15. }

**Output**

bye

**Explanation**

The Integer class caches integer values from -127 to 127. Therefore, the Integer objects can only be created in the range -128 to 127. The operator **==** will not work for the value greater than 127; thus **bye** is printed.

**Q57. Which types of exceptions are caught at compile time?**

Ans: Checked exceptions can be caught at the time of program compilation. Checked exceptions must be handled by using try catch block in the code in order to successfully compile the code.

**Q58. Describe different states of a thread.**

Ans: A thread in Java can be in either of the following states:

Ready: When a thread is created, it’s in Ready state.

Running: A thread currently being executed is in running state.

Waiting: A thread waiting for another thread to free certain resources is in waiting state.

Dead: A thread which has gone dead after execution is in dead state.

**Q59. Can we use a default constructor of a class even if an explicit constructor is defined?**

Ans: Java provides a default no argument constructor if no explicit constructor is defined in a Java class. But if an explicit constructor has been defined, default constructor can’t be invoked and developer can use only those constructors which are defined in the class.

**Q60. Can we override a method by using same method name and arguments but different return types?**

Ans: The basic condition of method overriding is that method name, arguments as well as return type must be exactly same as is that of the method being overridden. Hence using a different return type doesn’t override a method.

**Q61.What will be the output of following piece of code?**

public class operatorExample {

public static void main(String args[]) {

int x = 4;

system.out.println(x++);

}

}

Ans: In this case postfix ++ operator is used which first returns the value and then increments. Hence it’s output will be 4.

**Q61. A person says that he compiled a java class successfully without even having a main method in it? Is it possible?**

Ans: main method is an entry point of Java class and is required for execution of the program however; a class gets compiled successfully even if it doesn’t have a main method. It can’t be run though.

**Q62. Can we call a non-static method from inside a static method?**

Ans: Non-Static methods are owned by objects of a class and have object level scope and in order to call the non-Static methods from a static block (like from a static main method), an object of the class needs to be created first. Then using object reference, these methods can be invoked.

**Q63. What are the two environment variables that must be set in order to run any Java programs?**

Ans: Java programs can be executed in a machine only once following two environment variables have been properly set:

PATH variable

CLASSPATH variable

**Q64. Can variables be used in Java without initialization?**

Ans: In Java, if a variable is used in a code without prior initialization by a valid value, program doesn’t compile and gives an error as no default value is assigned to variables in Java.

**Q65. Can a class in Java be inherited from more than one class?**

Ans: In Java, a class can be derived from only one class and not from multiple classes. Multiple inheritances is not supported by Java.

**Q66. Can a constructor have different name than a Class name in Java?**

Ans: Constructor in Java must have same name as the class name and if the name is different, it doesn’t act as a constructor and compiler thinks of it as a normal method.

**Q67. What will be the output of Round(3.7) and Ceil(3.7)?**

Ans: Round(3.7) returns 4 and Ceil(3.7) returns 4.

**Q68: Can we use goto in Java to go to a particular line?**

Ans: In Java, there is not goto keyword and java doesn’t support this feature of going to a particular labeled line.

**Q69. Can a dead thread be started again?**

Ans: In java, a thread which is in dead state can’t be started again. There is no way to restart a dead thread.

**Q70. Is the following class declaration correct?**

**Ans:**

public abstract final class testClass {

// Class methods and variables

}

Ans: The above class declaration is incorrect as an abstract class can’t be declared as Final.

**Q71. Is JDK required on each machine to run a Java program?**

Ans: JDK is development Kit of Java and is required for development only and to run a Java program on a machine, JDK isn’t required. Only JRE is required.

**Q72. What’s the difference between comparison done by equals method and == operator?**

Ans: In Java, equals() method is used to compare the contents of two string objects and returns true if the two have same value while == operator compares the references of two string objects.

In the following example, equals() returns true as the two string objects have same values. However == operator returns false as both string objects are referencing to different objects:

public class equalsTest {

public static void main(String args[]) {

String str1 = new String("Hello World");

String str2 = new String("Hello World");

if (str1.equals(str2))

{ // this condition is true

System.out.println("str1 and str2 are equal in terms of values");

}

if (str1 == str2) {

//This condition is true

System.out.println("Both strings are referencing same object");

} else

{

// This condition is NOT true

System.out.println("Both strings are referencing different objects");

}

}

}

**Q73. Is it possible to define a method in Java class but provide it’s implementation in the code of another language like C?**

Ans: Yes, we can do this by use of native methods. In case of native method based development, we define public static methods in our Java class without its implementation and then implementation is done in another language like C separately.

**Q74. How are destructors defined in Java?**

Ans: In Java, there are no destructors defined in the class as there is no need to do so. Java has its own garbage collection mechanism which does the job automatically by destroying the objects when no longer referenced.

**Q75. Can a variable be local and static at the same time?**

Ans: No a variable can’t be static as well as local at the same time. Defining a local variable as static gives compilation error.

**Q76. Can we have static methods in an Interface?**

Ans: Static methods can’t be overridden in any class while any methods in an interface are by default abstract and are supposed to be implemented in the classes being implementing the interface. So it makes no sense to have static methods in an interface in Java.

**Q77. In a class implementing an interface, can we change the value of any variable defined in the interface?**

Ans: No, we can’t change the value of any variable of an interface in the implementing class as all variables defined in the interface are by default public, static and Final and final variables are like constants which can’t be changed later.

**Q78. Is it correct to say that due to garbage collection feature in Java, a java program never goes out of memory?**

Ans: Even though automatic garbage collection is provided by Java, it doesn’t ensure that a Java program will not go out of memory as there is a possibility that creation of Java objects is being done at a faster pace compared to garbage collection resulting in filling of all the available memory resources.

So, garbage collection helps in reducing the chances of a program going out of memory but it doesn’t ensure that.

**Q79. Can we have any other return type than void for main method?**

Ans: No, Java class main method can have only void return type for the program to get successfully executed.

Nonetheless , if you absolutely must return a value to at the completion of main method , you can use System.exit(int status)

**Q80. I want to re-reach and use an object once it has been garbage collected. How it’s possible?**

Ans: Once an object has been destroyed by garbage collector, it no longer exists on the heap and it can’t be accessed again. There is no way to reference it again.

**Q81. In Java thread programming, which method is a must implementation for all threads?**

Ans: Run() is a method of Runnable interface that must be implemented by all threads.

**Q82. I want to control database connections in my program and want that only one thread should be able to make database connection at a time. How can I implement this logic?**

Ans: This can be implemented by use of the concept of synchronization. Database related code can be placed in a method which hs **synchronized**keyword so that only one thread can access it at a time.

**Q83. How can an exception be thrown manually by a programmer?**

Ans: In order to throw an exception in a block of code manually, **throw** keyword is used. Then this exception is caught and handled in the catch block.

public void topMethod() {

try {

excMethod();

} catch (ManualException e) {}

}

public void excMethod {

String name = null;

if (name == null) {

throw (new ManualException("Exception thrown manually ");

}

}

**Q84. I want my class to be developed in such a way that no other class (even derived class) can create its objects. How can I do so?**

Ans: If we declare the constructor of a class as private, it will not be accessible by any other class and hence, no other class will be able to instantiate it and formation of its object will be limited to itself only.

**Q85. How objects are stored in Java?**

Ans: In java, each object when created gets a memory space from a heap. When an object is destroyed by a garbage collector, the space allocated to it from the heap is re-allocated to the heap and becomes available for any new objects.

**Q86. How can we find the actual size of an object on the heap?**

Ans: In java, there is no way to find out the exact size of an object on the heap.

**Q87. Which of the following classes will have more memory allocated?**

**Class A: Three methods, four variables, no object**

**Class B: Five methods, three variables, no object**

Ans: Memory isn’t allocated before creation of objects. Since for both classes, there are no objects created so no memory is allocated on heap for any class.

**Q88. What happens if an exception is not handled in a program?**

Ans: If an exception is not handled in a program using try catch blocks, program gets aborted and no statement executes after the statement which caused exception throwing.

**Q89. I have multiple constructors defined in a class. Is it possible to call a constructor from another constructor’s body?**

Ans: If a class has multiple constructors, it’s possible to call one constructor from the body of another one using **this()**.

**Q90. What’s meant by anonymous class?**

Ans: An anonymous class is a class defined without any name in a single line of code using new keyword.

For example, in below code we have defined an anonymous class in one line of code:

public java.util.Enumeration testMethod()

{

return new java.util.Enumeration()

{

@Override

public boolean hasMoreElements()

{

// TODO Auto-generated method stub

return false;

}

@Override

public Object nextElement()

{

// TODO Auto-generated method stub

return null;

}

}

**Q91. Is there a way to increase the size of an array after its declaration?**

Ans: Arrays are static and once we have specified its size, we can’t change it. If we want to use such collections where we may require a change of size ( no of items), we should prefer vector over array.

**Q92. If an application has multiple classes in it, is it okay to have a main method in more than one class?**

Ans: If there is main method in more than one classes in a java application, it won’t cause any issue as entry point for any application will be a specific class and code will start from the main method of that particular class only.

**Q93. I want to persist data of objects for later use. What’s the best approach to do so?**

Ans: The best way to persist data for future use is to use the concept of serialization.

**Q94. What is a Local class in Java?**

Ans: In Java, if we define a new class inside a particular block, it’s called a local class. Such a class has local scope and isn’t usable outside the block where its defined.

**Q95. String and StringBuffer both represent String objects. Can we compare String and StringBuffer in Java?**

Ans: Although String and StringBuffer both represent String objects, we can’t compare them with each other and if we try to compare them, we get an error.

**Q96. Which API is provided by Java for operations on set of objects?**

Ans: Java provides a Collection API which provides many useful methods which can be applied on a set of objects. Some of the important classes provided by Collection API include ArrayList, HashMap, TreeSet and TreeMap.

**Q97. Can we cast any other type to Boolean Type with type casting?**

Ans: No, we can neither cast any other primitive type to Boolean data type nor can cast Boolean data type to any other primitive data type.

**Q98. Can we use different return types for methods when overridden?**

Ans: The basic requirement of method overriding in Java is that the overridden method should have same name, and parameters.But a method can be overridden with a different return type as long as the new return type extends the original.

For example , method is returning a reference type.

Class B extends A {

A method(int x) {

//original method

}

B method(int x) {

//overridden method

}

}

**Q99. What’s the base class of all exception classes?**

Ans: In Java, **Java.lang.Throwable** is the super class of all exception classes and all exception classes are derived from this base class.

**Q100. What’s the order of call of constructors in inheritance?**

Ans: In case of inheritance, when a new object of a derived class is created, first the constructor of the super class is invoked and then the constructor of the derived class is invoked.

^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

Simplilearn

**Top 70 Core Java Interview Questions Answered for Freshers and Experienced in 2022**

Lesson 29 of 33[By Ravikiran A S](https://www.simplilearn.com/authors/ravikiran-a-s)

Last updated on Jun 5, 202211601618



[Previous](https://www.simplilearn.com/tutorials/java-tutorial/java-projects-for-beginner)[Next](https://www.simplilearn.com/tutorials/java-tutorial/java-programming)

Table of Contents

[Basic Java Interview Questions for Freshers](https://www.simplilearn.com/tutorials/java-tutorial/java-interview-questions#basic_java_interview_questions_for_freshers)

[Intermediate Java Interview Questions](https://www.simplilearn.com/tutorials/java-tutorial/java-interview-questions#intermediate_java_interview_questions)

[Advanced Java Interview Questions for Experienced](https://www.simplilearn.com/tutorials/java-tutorial/java-interview-questions#advanced_java_interview_questions_for_experienced)

[Next Steps](https://www.simplilearn.com/tutorials/java-tutorial/java-interview-questions#next_steps)

[Java](https://www.simplilearn.com/tutorials/java-tutorial/what-is-java) is the most widely used [programming language](https://www.simplilearn.com/tutorials/programming-tutorial/first-programming-language) in the current IT industry. One major reason for the vast number of beginners and professionals in the field of[programming](https://www.simplilearn.com/how-to-learn-programming-article) is the career potential that Java knowledge comes with. This article is dedicated to the same purpose. Here is a complete guide on how to help you crack the most frequently asked Java Interview questions.

**Full Stack Java Developer Course**

In Partnership with HIRIST and HackerEarth[EXPLORE COURSE](https://www.simplilearn.com/java-full-stack-developer-certification-training-course?source=GhPreviewCTABanner)



Basic Java Interview Questions for Freshers

So let’s get started with the first set of basic Java Interview Questions which is primarly useful for freshers.

1. What are the differences between C++ and Java?

Concept.

[C++](https://www.simplilearn.com/tutorials/cpp-tutorial/learn-cpp-basics) is not platform-independent; the principle behind C++ programming is “write once, compile anywhere.”

In contrast, because the byte code generated by the Java compiler is platform-independent, it can run on any machine, Java programs are written once and run everywhere.

Languages Compatibility.

C++ is a programming language that is based on the [C programming language](https://www.simplilearn.com/c-programming-article). Most other high-level languages are compatible with C++.

Most of the languages of Java are incompatible. Java is comparable to those of C and C++.

Interaction with the library.

It can access the native system libraries directly in C++. As a result, it’s better for programming at the system level.

Java’s native libraries do not provide direct call support. You can use Java Native Interface or access the libraries.

Characteristics.

C++ distinguishes itself by having features that are similar to procedural and object-oriented languages. The characteristic that sets Java apart is automatic garbage collection. Java doesn’t support destructors at the moment.

The semantics of the type.

Primitive and object types in C++ have the same kind of semantics. The primitive and [object classes of Java](https://www.simplilearn.com/tutorials/java-tutorial/java-classes-and-objects), on the other hand, are not consistent.

In the context of Compiler and Interpreter.

Java refers to a compiled and interpreted language. In contrast, C++ is only a compiled language.

In Java, the source code is the compiled output is a platform-independent byte code.

In C++, the source program is compiled into an object code that is further executed to produce an output.

2. List the features of the Java Programming language?

A few of the significant features of [Java Programming Language](https://www.simplilearn.com/tutorials/java-tutorial/java-programming) are:

Easy: Java is a language that is considered easy to learn. One fundamental concept of OOP Java has a catch to understand.

Secured Feature: Java has a secured feature that helps develop a virus-free and tamper-free system for the users.

OOP: OOP stands for [Object-Oriented Programming language](https://www.simplilearn.com/tutorials/cpp-tutorial/oops-concepts-in-cpp). OOP signifies that, in Java, everything is considered an object.

Independent Platform: Java is not compiled into a platform-specific machine; instead, it is compiled into platform-independent bytecode. This code is interpreted by the Virtual Machine on which the platform runs.

3. What do you get in the Java download file? How do they differ from one another?

We get two major things along with the Java Download file.

JDK - [Java Development Kit](https://www.simplilearn.com/tutorials/java-tutorial/jdk-in-java)

JRE - Java Runtime Environment

|  |  |
| --- | --- |
| JDK | JRE |
| Abbreviation for JavaDevelopment Kit | Abbreviation for Java Runtime Environment |
| JDK is a dedicated kit for solely software development | JRE is a set of software and library designed for executing Java Programs |
| Unlike JVM, JDK is Platform Dependent | Unlike JVM, JRE is also Platform Dependent |
| JDK package is a set of tools for debugging and Developing | JRE Package is one that only supports files and libraries for a runtime environment |
| JDK package will be provided with an installer file | JRE Package does not get an installer but has only a runtime environment |

**FREE Java Certification Training**

Learn A-Z of Java like never before[ENROLL NOW](https://www.simplilearn.com/learn-java-basics-skillup?utm_source=frs&utm_medium=skillup-course-banner&utm_campaign=frs-tutorial-skillup-course-promotion)



4. What is a ClassLoader?

A classloader in Java is a subsystem of Java Virtual Machine, dedicated to loading class files when a program is executed; ClassLoader is the first to load the executable file.

Java has [Bootstrap](https://www.simplilearn.com/tutorials/angular-tutorial/angular-bootstrap), Extension, and Application classloaders.

5. What are the Memory Allocations available in JavaJava?

Java has five significant types of memory allocations.

Class Memory

Heap Memory

Stack Memory

Program Counter-Memory

Native Method Stack Memory

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

[Stack](https://www.simplilearn.com/tutorials/data-structure-tutorial/stacks-in-data-structures) is generally used to store the order of method execution and local variables. In contrast, Heap memory is used to store the objects. After storing, they use dynamic memory allocation and deallocation.

7. Will the program run if we write static public void main?

Yes, the program will successfully execute if written so. Because, in Java, there is no specific rule for the order of specifiers

8. What is the default value stored in Local Variables?

Neither the Local Variables nor any primitives and Object references have any default value stored in them.

9. Explain the expected output of the following code segment?

public class Simplilearn

{

    public static void main (String args[])

    {

        System.out.println(100 + 100 +“Simplilearn");

        System.out.println(“E-Learning Company" + 100 + 100);

    }

}

The answers for the two print statements are as follows.

200Simplilearn

E-Learning Company100100

10. What is an Association?

An Association can be defined as a relationship that has no ownership over another. For example, a person can be associated with multiple banks, and a bank can be related to various people, but no one can own the other.

11. What do you mean by aggregation?

The term aggregation refers to the relationship between two classes best described as a “whole/part” and “has-a” relationship. This kind is the most specialized version of an association relationship. It contains the reference to another class and is said to have ownership of that class.

12. Define Copy Constructor in Java

A Copy Constructor in Java is a constructor that initializes an object through another object of the same class.

13. What is a Marker Interface?

An empty[interface in Java](https://www.simplilearn.com/tutorials/java-tutorial/java-interface) is referred to as a Marker interface. Serializable and Cloneable are some famous examples of Marker Interface.

14. What is Object Cloning?

An ability to recreate an object entirely similar to an existing object is known as Object Cloning in Java. Java provides a clone() method to clone a current object offering the same functionality as the original object.

15. Why is Java not completely object-oriented?

Java is not considered as a 100% object-oriented [programming language](https://www.simplilearn.com/best-programming-languages-start-learning-today-article) because it still makes use of eight or more primitive data types like int, float double, etc.

16. What is an object-oriented paradigm?

A Paradigm that is based on the concepts of “Objects.” It contains data and code. Data that is in the form of fields, and regulation, that is in the form of procedures. The exciting feature of this paradigm is that the object’s procedures can access and often modify the data fields themselves.

17. Define Wrapper Classes in Java.

In Java, when you declare primitive datatypes, then Wrapper classes are responsible for converting them into objects(Reference types).

18. Define Singleton Classes in Java.

In Java, when you make the constructor of a class private, that particular class can generate only one object. This type of class is popularly known as a Singleton Class.

19. Define package in Java.

The package is a collective bundle of classes and interfaces and the necessary libraries and JAR files. The use of packages helps in code reusability.

**Post Graduate Program: Full Stack Web Development**

in Collaboration with Caltech CTME[ENROLL NOW](https://www.simplilearn.com/pgp-full-stack-web-development-certification-training-course?source=GhPreviewCTABanner)



20. Can you implement pointers in a Java Program?

Java Virtual Machine takes care of memory management implicitly. Java's primary motto was to keep programming simple. So, accessing memory directly through pointers is not a recommended action. Hence, pointers are eliminated in Java.

21. Differentiate between instance and local variables.

For instance, [variables](https://www.simplilearn.com/tutorials/javascript-tutorial/scope-of-variables-in-javascript) are declared inside a class, and the scope is limited to only a specific object.

A local variable can be anywhere inside a method or a specific block of code. Also, the scope is limited to the code segment where the variable is declared.

22. Explain Java String Pool.

A collection of strings in Java's Heap memory is referred to as Java String Pool. In case you try to create a new string object, JVM first checks for the presence of the object in the pool. If available, the same object reference is shared with the variable, else a new object is created.

23. What is an Exception?

An [Exception in Java](https://www.simplilearn.com/tutorials/java-tutorial/exception-handling-in-java) is considered an unexpected event that can disrupt the program's normal flow. These events can be fixed through the process of Exception Handling.

24. What is the final keyword in Java?

The term final is a predefined word in Java that is used while declaring values to variables. When a value is declared using the final keyword, then the variable's value remains constant throughout the program's execution.

25. What happens when the main() isn't declared as static?

When the main method is not declared as static, then the program may be compiled correctly but ends up with a severe ambiguity and throws a run time error that reads "NoSuchMethodError."

Get a firm foundation in Java, the most commonly used programming language in software development with the [Java Certification Training Course](https://www.simplilearn.com/mobile-and-software-development/java-javaee-soa-development-training?source=GhPreviewCTAText).

With this we are done with the first section that is Basic Java Interview Question, Now, lets move on to our next section of Intermediate Java Interview Questions.

Intermediate Java Interview Questions

26. What is JDK? Mention the variants of JDK?

JDK is an abbreviation for Java Development Kit. It is a combined Package of JRE and Developer tools used for designing [Java Applications](https://www.simplilearn.com/popular-java-applications-article) and Applets. Oracle has the following variants.

JDK Standard Edition

JDK Enterprise Edition

JDK Micro Edition

27. What is the difference between JDK, JRE, and JVM?

JVM has a Just in Time (JIT) compiler tool that converts all the Java source code into the low-level compatible machine language. Therefore, it runs faster than the regular application.

JRE has class libraries and other JVM supporting files. But it doesn’t have any tool for java development such as compiler or debugger.

JDK has tools that are required to write Java Programs and uses JRE to execute them. It has a compiler, Java application launcher, and an applet viewer.

28. What is a JIT compiler?

JIT compiler refers to Just in Time compiler. It is the simplest way of executing the computer code that takes in compilation during the execution of a program rather than before performance. It commonly uses bytecode translation to machine code. It is then executed directly.

29. What are Brief Access Specifiers and Types of Access Specifiers?

Access Specifiers are predefined keywords used to help JVM understand the scope of a variable, method, and class. We have four access specifiers.

Public Access Specifier

Private Access Specifier

Protected Access Specifier

Default Access Specifier

**New Course: Full Stack Development for Beginners**

Learn Git Command, Angular, NodeJS, Maven & More[ENROLL NOW](https://www.simplilearn.com/free-online-full-stack-development-course-skillup?utm_source=frs&utm_medium=skillup-course-banner&utm_campaign=frs-tutorial-skillup-course-promotion)



30. How many types of constructors are used in Java?

There are two types of [constructors](https://www.simplilearn.com/tutorials/java-tutorial/constructor-in-java) that are used in Java.

Parameterized Constructors: Parameterized constructor accepts the parameters with which users can initialize the instance variables. Users can initialize the class variables dynamically at the time of instantiating the class.

Default constructors: This type doesn’t accept any parameters; rather, it instantiates the class variables with their default values. It is used mainly for object creation.

31. Can a constructor return a value?

Yes, A constructor can return a value. It replaces the class's current instance implicitly; you cannot make a constructor return a value explicitly.

32. Explain ‘this’ keyword in Java.

The term "this" is a particular keyword designated as a reference keyword. The "this" keyword is used to refer to the current class properties like method, instance, variable, and constructors.

33. Explain ‘super’ keyword in Java.

The term "super" is a particular keyword designated as a reference keyword. The "super" keyword refers to the immediate parent class object.

34. Explain Method Overloading in Java.

The process of creating multiple method signatures using one method name is called Method Overloading in Java. Two ways to achieve method overloading are:

Varying the number of arguments

Changing the return type of the Method

35. Can we overload a static method?

No, Java does not support the Overloading of a [static method.](https://www.simplilearn.com/tutorials/java-tutorial/static-keyword-in-java) The process would throw an error reading "static method cannot be referenced."

36. Define Late Binding.

Binding is a process of unifying the method call with the method's code segment. Late binding happens when the method's code segment is unknown until it is called during the runtime.

37. Define Dynamic Method Dispatch.

The Dynamic method dispatch is a process where the method call is executed during the runtime. A reference variable is used to call the super-class. This process is also known as Run-Time Polymorphism.

38. Why is the delete function faster in the linked list than an array?

Delete Function is faster in [linked lists](https://www.simplilearn.com/tutorials/java-tutorial/linked-list-in-java) as the user needs to make a minor update to the pointer value so that the node can point to the next successor in the list

39. Give a briefing on the life cycle of a thread.

The life cycle of a thread includes five stages, as mentioned below.

New Born State

Runnable State

Running State

Blocked State

Dead State

40. Explain the difference between >> and >>> operators.

Although they look similar, there is a massive difference between both.

>> operator does the job of right shifting the sign bits

>>> operator is used in shifting out the zero-filled bits

41. Brief the life cycle of an applet.

The life cycle of an applet involves the following.

Initialization

Start

Stop

Destroy

Paint

42. Why are generics used in Java Programming?

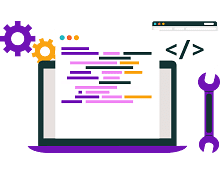
Compile-time type safety is provided by using generics. Compile-time type safety allows users to catch unnecessary invalid types at compile time. Generic methods and classes help programmers specify a single method declaration, a set of related methods, or related types with an available class declaration.

43. Explain the Externalizable interface.

The Externalizable interface helps with control over the process of serialization. An "externalisable" interface incorporates readExternal and writeExternal methods.

**Full Stack Web Developer Course**

To become an expert in MEAN Stack[VIEW COURSE](https://www.simplilearn.com/full-stack-web-developer-mean-stack-certification-training?source=GhPreviewCTABanner)



44. What is the Daemon Thread?

The Daemon thread can be defined as a thread with the least priority. This Daemon thread is designed to run in the background during the Garbage Collection in Java.

The setDaemon() method creates a Daemon thread in Java.

45. Explain the term enumeration in Java.

Enumeration or [enum](https://www.simplilearn.com/tutorials/java-tutorial/enum-in-java" \t "_blank" \o "enum) is an interface in Java. Enum allows the sequential access of the elements stored in a collection in Java.

46. Why is Java is Dynamic?

Java is designed to adapt to an evolving environment. Java programs include a large amount of runtime information that is used to resolve access to objects in real-time.

47. Can you run a code before executing the main method?

Yes, we can execute any code, even before the main method. We will be using a static block of code when creating the objects at the class's load time. Any statements within this static block of code will get executed at once while loading the class, even before creating objects in the main method.

48. How many times is the finalize method called?

The finalize method is called the Garbage collector. For every object, the Garbage Collector calls the finalize() method just for one time.

What is the purpose of the finalize() method?

The finalize() method is invoked just before the object is garbage collected. It is used to perform cleanup processing. The Garbage collector of JVM collects only those objects that are created by new keyword. So if you have created an object without new, you can use the finalize method to perform cleanup processing (destroying remaining objects). The cleanup processing is the process to free up all the resources, network which was previously used and no longer needed. It is essential to remember that it is not a reserved keyword, finalize method is present in the object class hence it is available in every class as object class is the superclass of every class in java. Here, we must note that neither finalization nor garbage collection is guaranteed. Consider the following example.

**public** **class** FinalizeTest {

**int** j=12;

**void** add()

    {

        j=j+12;

        System.out.println("J="+j);

    }

**public** **void** finalize()

    {

        System.out.println("Object is garbage collected");

    }

**public** **static** **void** main(String[] args) {

**new** FinalizeTest().add();

        System.gc();

**new** FinalizeTest().add();

    }

}

Advanced Java Interview Questions for Experienced

Now, lets move on to our last section of Advanced Java Interview Questions which is primarly useful for experienced and working professionals.

49. Can "this" and "super" keywords be used together?

No, "this" and "super" keywords should be used in the first statement in the class constructor. The following code gives you a brief idea.

public class baseClass {

     baseClass() {

         super();

         this();

         System.out.println(" baseClass object is created");

     }

     public static void main(String []args){

         baseClass bclass = new baseClass();

     }

}

50. What is a JSP page?

JSP is an abbreviation for Java Servlet Page. The JSP page consists of two types of text.

Static Data

JSP elements

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Find Our Java Training in Top Cities   |  |  |  | | --- | --- | --- | | India | United States | Other Countries | | [Java Training in Bangalore](https://www.simplilearn.com/advanced-java-training-course-bangalore-city) | [Java Training New York](https://www.simplilearn.com/advanced-java-training-course-new-york-city) | [Java Course London](https://www.simplilearn.com/advanced-java-training-course-london-city) | | [Java Training in Chennai](https://www.simplilearn.com/advanced-java-training-course-chennai-city) | [Java Training San Diego](https://www.simplilearn.com/advanced-java-training-course-san-diego-city) | [Java Course Singapore](https://www.simplilearn.com/advanced-java-training-course-singapore-city) | | [Java Training in Hyderabad](https://www.simplilearn.com/advanced-java-training-course-hyderabad-city) | [Java Training Dallas](https://www.simplilearn.com/advanced-java-training-course-dallas-city) | [Java Course Melbourne](https://www.simplilearn.com/advanced-java-training-course-melbourne-city) | |

**Free Course: JavaScript for Beginners**

Learn the Basics of JavaScript[ENROLL NOW](https://www.simplilearn.com/learn-javascript-basics-free-course-skillup?utm_source=frs&utm_medium=skillup-course-banner&utm_campaign=frs-tutorial-skillup-course-promotion)



51. What is JDBC?

JDBC is an abbreviation for Java Database Connector.

JDBC is an abstraction layer used to establish connectivity between an existing database and a Java application

52. Explain the various directives in JSP.

Directives are instructions processed by JSP Engine. After the JSP page is compiled into a Servlet, Directives set page-level instructions, insert external files, and define customized tag libraries. Directives are defined using the symbols below:

start with "< %@" and then end with "% >"

The various types of directives are shown below:

Include directive

It includes a file and combines the content of the whole file with the currently active pages.

Page directive

Page Directive defines specific attributes in the JSP page, like the buffer and error page.

Taglib

Taglib declares a custom tag library, which is used on the page.

53. What are the observer and observable classes?

Objects that inherit the "Observable class" take care of a list of "observers."

When an Observable object gets upgraded, it calls the update() method of each of its observers.

After that, it notifies all the observers that there is a change of state.

The Observer interface gets implemented by objects that observe Observable objects.

54. What is Session Management in Java?

A session is essentially defined as the random conversation's dynamic state between the client and the server. The virtual communication channel includes a string of responses and requests from both sides. The popular way of implementing session management is establishing a session ID in the client's communicative discourse and the server.

55. Briefly explain the term Spring Framework.

Spring is essentially defined as an application [framework](https://www.simplilearn.com/tutorials/java-tutorial/java-frameworks) and inversion of control containers for Java. The spring framework creates enterprise applications in Java. Especially useful to keep in mind that the spring framework's central features are essentially conducive to any Java application.

56. How to handle exceptions in Spring MVC Framework?

[Spring MVC](https://www.simplilearn.com/tutorials/spring-boot-tutorial/spring-vs-spring-boot) has two approaches for handling the exceptions:

Exception handler method: In this kind of exception handling, the user will get the @ExceptionHandler annotation type used to annotate a method to handle exceptions.

XML Configuration: The user can use the SimpleMappingExceptionResolver bean in Spring’s application file and map the exception.

57. What is JCA in Java?

Java Cryptography Architecture gives a platform and provides architecture and application programming interfaces that enable decryption and encryption.

Developers use Java Cryptography Architecture to combine the application with the security applications. Java Cryptography Architecture helps in implementing third party security rules and regulations.

Java Cryptography Architecture uses the hash table, encryption message digest, etc. to implement the security.

58. Explain JPA in Java.

The Java Persistence API enables us to create the persistence layer for desktop and web applications. Java Persistence deals in the following:

Java Persistence API

Query Language

Java Persistence Criteria API

Object Mapping Metadata

59. Explain the different authentications in Java Servlets.

Authentication options are available in Servlets: There are four different options for authentication in servlet:

Basic Authentication:

Usernames and passwords are given by the client to authenticate the user.

Form-based authentication:

In this, the login form is made by the programmer by using [HTML](https://www.simplilearn.com/tutorials/html-tutorial/what-is-html).

Digest Authentication:

It is similar to basic authentication, but the passwords are encrypted using the Hash formula. Hash Formula makes digest more secure.

Client certificate Authentication:

It requires that each client accessing the resource has a certificate that it sends to authenticate itself. Client Authentication requires the SSL protocol.

Learn top skills demanded in the industry, including Angular, Spring Boot, Hibernate, Servlets, and JSPs, and SOA to build highly web scalable apps with the [Full Stack Java Developer Masters Program](https://www.simplilearn.com/java-full-stack-developer-certification-training-course?source=GhPreviewCTAText).

60. Explain FailFast iterator and FailSafe iterator along with examples for each.

FailFast iterators and FailSafe iterators are used in Java Collections.

FailFast iterators do not allow changes or modifications to the Java Collections, which means they fail when the latest element is added to the collection or an existing element gets removed from the collection. The FailFast iterators tend to fail and throw an exception called ConcurrentModificationException.

Ex: ArrayList, HashMap

Whereas, on the other hand, FailSafe iterators allow changes or modifications to be done on the Java Collections. It is possible, as the FailSafe iterators usually operate on the cloned copy of the collection. Hence, they do not throw any specific exception.

Ex: CopyOnWriteArrayList

61. How do we reverse a string?

The [string can be reversed](https://www.simplilearn.com/tutorials/java-tutorial/reverse-a-string-in-java) by using the following program.

package simplilearnJava;

public class StringReverse {

public static void main(String args[]) {

String str = "Simplilearn";

String reverse = new StringBuffer(str).reverse().toString();

System.out.printf("Actual Word: %s, Word after reversing %s", str, reverse);

}

public static String reverse(String source) {

if (source == null || source.isEmpty()) {

return source;

}

String reverse = "";

for (int i = source.length() - 1; i >= 0; i--) {

reverse = reverse + source.charAt(i);

}

return reverse;

}

}

Expected Output:

Actual Word: Simplilearn, Word after reversing nraelilpmiS

62. Write a program to find the square root of a number.

The Square root of a number can be found by using the following program.

package simplilearnJava;

import java.util.Scanner;

public class SRoot {

public static void main(String args[]) {

try (Scanner sc = new Scanner(System.in)) {

System.out.println("Input a number to find square root: ");

double square = sc.nextDouble();

double squareRoot = Math.sqrt(square);

System.out.printf("The square root is: %f ", squareRoot);

}

}

}

Expected Output:

Input a number to find square root:

25

The square root is: 5

**Free Course: Java Hibernate Fundamentals**

Learn Java Hibernate Basic Skills for Free[ENROLL NOW](https://www.simplilearn.com/learn-java-hibernate-basics-skillup?utm_source=frs&utm_medium=skillup-course-banner&utm_campaign=frs-tutorial-skillup-course-promotion)



63. Write a program that detects the duplicate characters in a string.

The program that finds the duplicate elements in a string is written below:

package simplilearnJava;

import java.util.HashMap;

import java.util.Map;

import java.util.Set;

public class FindDuplicate {

public static void main(String args[]) {

printDuplicateCharacters("Simplilearn");

}

public static void printDuplicateCharacters(String word) {

char[] characters = word.toCharArray();

Map<Character, Integer> charMap = new HashMap<Character, Integer>();

for (Character ch : characters) {

if (charMap.containsKey(ch)) {

charMap.put(ch, charMap.get(ch) + 1);

} else {

charMap.put(ch, 1);

}

}

Set<Map.Entry<Character, Integer>> entrySet = charMap.entrySet();

System.out.printf("List of duplicate characters in String '%s' %n", word);

for (Map.Entry<Character, Integer> entry : entrySet) {

if (entry.getValue() > 1) {

System.out.printf("%s: %d %n", entry.getKey(), entry.getValue());

}

}

}

}

Expected output:

List of duplicate characters in String 'Simplilearn.'

i: 2

l: 2

64. Write a Program to remove duplicates in an ArrayList.

The following program can be implemented to remove duplicate elements in an ArrayList

package simplilearnJava;

import java.util.ArrayList;

import java.util.LinkedHashSet;

import java.util.List;

import java.util.Set;

public class ArrayDuplicate {

public static void main(String args[]) {

List<Integer> num = new ArrayList<Integer>();

num.add(1);

num.add(2);

num.add(3);

num.add(4);

num.add(5);

num.add(6);

num.add(3);

num.add(4);

num.add(5);

num.add(6);

System.out.println("Your list of elements in ArrayList : " + num);

Set<Integer> primesWithoutDuplicates = new LinkedHashSet<Integer>(num);

num.clear();

num.addAll(primesWithoutDuplicates);

System.out.println("list of original numbers without duplication: " + num);

}

}

Expected Output:

Your list of elements in ArrayList : [1, 2, 3, 4, 5, 6, 3, 4, 5, 6]

list of original numbers without duplication: [1, 2, 3, 4, 5, 6]

65. Find the word count in a [string](https://www.simplilearn.com/tutorials/java-tutorial/java-strings) using HashMap Collection.

The following program can be used for word count.

package simplilearnJava;

import java.util.HashMap;

public class WordCount {

public static void main(String[] args) {

String str = "Hello World, Welcome to Simplilearn";

String[] split = str.split(" ");

HashMap<String, Integer> map = new HashMap<String, Integer>();

for (int i = 0; i < split.length; i++) {

if (map.containsKey(split[i])) {

int count = map.get(split[i]);

map.put(split[i], count + 1);

} else {

map.put(split[i], 1);

}

}

System.out.println(map);

}

}

Expected Output:

{Hello=1, Simplilearn=1, Welcome=1, to=1, World,=1}

66. Write a program to find the Second Highest number in an ArrayList

The following program can be used to find the second biggest number in an array list.

package simplilearnJava;

public class NextHighest {

public static void main(String[] args)

    {

        int array[] = { 1, 2, 3, 4, 11, 12, 13, 14, 21, 22, 23, 24, 31, 32};

        int high = 0;

        int nextHigh = 0;

        System.out.println("The given array is:");

        for (int i = 0; i < array.length; i++)

        {

            System.out.print(array[i] + "\t");

        }

        for (int i = 0; i < array.length; i++)

        {

            if (array[i] > high)

            {

                nextHigh = high;

                high = array[i];

            }

            else if (array[i] > nextHigh)

            {

                nextHigh = array[i];

            }

        }

        System.out.println("Second Highest is:" + nextHigh);

        System.out.println("Highest Number is: "  +high);

    }

}

Expected Output:

The given array is:

1 2 3 4 11 12 13 14 21 22 23 24 31 32

Second Highest is:31

The highest number is: 32

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

System.out and System.err represent the monitor by default and thus can be used to send data or results to the monitor. System.out is used to display normal messages and results. System.eerr is used to display error messages. System.in represents InputStream object which by default represents standard input device, i.e., keyboard.

68. Could you provide some implementation of a Dictionary having a large number of words?

The simplest implementation that can be given is that of a List wherein one can place ordered words and perform a Binary search. The other implementation with a better search performance is HashMap where the key is used as the first character of the word and the value as a LinkedList.

Up another level, there are HashMaps like:

hashmap {

a (key) -> hashmap (key-aa , value (hashmap(key-aaa,value)

b (key) -> hashmap (key-ba , value (hashmap(key-baa,value)

z (key) -> hashmap (key-za , value (hashmap(key-zaa,value)

}

Up to n levels where n is the average size of the word in the dictionary.

69. How would you tackle it if you might have to encounter pattern programs in Java?

Solution - [Top 25 Most Frequently asked Pattern Programs in Java](https://www.simplilearn.com/tutorials/java-tutorial/pattern-programs-in-java)

With this, we have come to the end of this Java Interview Questions article. Moving ahead, we will look into the next crucial steps that you could pursue, to master Java.

We offer complete Job Guarantee and money-back if you don't secure a job within 6 months of graduation. Get interview ready with intense and comprehensive career mentoring sessions in our [Full Stack Java Developer Program.](https://www.simplilearn.com/java-full-stack-developer-certification-training-course?source=GhPreviewCTAText) Enroll TODAY!

Next Steps

Java Interview Questions are really important to go through before attending an interview. It helps you to put yourself on the safer side by getting you ready to be able to answer the questions asked in your interview. Now, the next step is to learn [How to Become a Software Developer](https://m.youtube.com/watch?v=Fq4GSr5-Qbs&t=4s).

You can also explore and get familiar with interview questions related to other backend languages like PHP, Python and [Node.js Interview Questions.](https://www.simplilearn.com/tutorials/nodejs-tutorial/nodejs-interview-questions)

If you're looking for more in-depth knowledge about the Java programming language and information on how to get certified as a [professional developer](https://www.simplilearn.com/how-to-become-complete-web-development-professional-article), explore our Java training and certification programs, which Simplilearn’s experienced industry experts offer in real-time. In particular, check out our [Full Stack Java Developer Master’s Program](https://www.simplilearn.com/java-full-stack-developer-certification-training-course) today!

If you have any questions about this “Java Interview Questions” article, please leave them in the comments section, and our experts will answer them for you, at the earliest!

JavaTPoint

### 133) Explain the hierarchy of Java Exception classes?

The java.lang.Throwable class is the root class of Java Exception hierarchy which is inherited by two subclasses: Exception and Error. A hierarchy of Java Exception classes are given below:



What are the ways to instantiate the Class class?

There are three ways to instantiate the Class class.

* **forName() method of Class class:** The forName() method is used to load the class dynamically. It returns the instance of Class class. It should be used if you know the fully qualified name of the class. This cannot be used for primitive types.
* **getClass() method of Object class:** It returns the instance of Class class. It should be used if you know the type. Moreover, it can be used with primitives.
* **the .class syntax:** If a type is available, but there is no instance then it is possible to obtain a Class by appending ".class" to the name of the type. It can be used for primitive data type also.

### What is the purpose of using javap?

The javap command disassembles a class file. The javap command displays information about the fields, constructors and methods present in a class file.

**Syntax**

javap fully\_class\_name