Java Interview Questions

~Navnath Jadhav

1) What is Java?

- *Java is a general purpose, class based, high-level, object oriented programming language and it is platform independent.*
- *Java is fast, secure and reliable.*
- *It was developed by sun microsystem*
- *There are a lot of applications, websites and games that are developed using java.*

2) What are the features of Java?

- It is one of the easy-to-use programming language to learn.
- Write code once and run it on almost any computing platform
- Java is platform independent. Some programs developed in one machine can be executed in another machine
- It is designed for building object oriented applications
- It is a multithreaded language with automatic memory management.

3) What is Java used for?

- It is used for developing android apps
- Helps you to create Enterprise software
- Wise range of mobile java applications
- Used for Big Data Analytics
- Why Java is called multithreaded language?
- *Multithreading is a Java feature that allows concurrent execution of two or more parts of a program for maximum utilization of CPU. Each part of such program is called a thread. So, threads are light-weight processes within a process.*

4) What are components of Java?

Components of Java Architecture

There are three main components of Java language: **JVM, JRE, and JDK**.

Java Virtual Machine, Java Runtime Environment and Java Development Kit respectively.

Let me elaborate each one of them one by one:

Java Virtual Machine:

Ever heard about WORA? (Write once Run Anywhere). Well, Java applications are called WORA because of their ability to run a code on any platform. This is done only because of JVM. The JVM is a Java platform component that provides an environment for executing Java programs. JVM interprets the bytecode into machine code which is executed in the machine in which the Java program runs.

So, in a nutshell, JVM performs the following functions:

- Loads the code
- Verifies the code
- Executes the code
- Provides runtime environment

Now, let me show you the JVM architecture. Here goes!

Explanation:

Class Loader: Class loader is a subsystem of JVM. It is used to load class files. Whenever we run the java program, class loader loads it first.

Class method area: It is one of the Data Area in JVM, in which Class data will be stored. Static Variables, Static Blocks, Static Methods, Instance Methods are stored in this area.

Heap: A heap is created when the JVM starts up. It may increase or decrease in size while the application runs.

Stack: JVM stack is known as a thread stack. It is a data area in the JVM memory which is created for a single execution thread. The JVM stack of a thread is used by the thread to store various elements i.e.; local variables, partial results, and data for calling method and returns.

Native stack: It subsumes all the native methods used in your application.

Execution Engine:

- JIT compiler
- Garbage collector

JIT compiler: The Just-In-Time (JIT) compiler is a part of the runtime environment. It helps in improving the performance of Java applications by compiling bytecodes to machine code at run time. The JIT compiler is enabled by default. When a method is compiled, the JVM calls the compiled code of that method directly. The JIT compiler compiles the bytecode of that method into machine code, compiling it “just in time” to run.

Garbage collector: As the name explains that Garbage Collector means to collect the unused material. Well, in JVM this work is done by Garbage collection. It tracks each and every object available in the JVM heap space and removes

unwanted ones.
Garbage collector works in two simple steps known as Mark and Sweep:

- Mark – it is where the garbage collector identifies which piece of memory is in use and which are not
- Sweep – it removes objects identified during the "mark" phase.

Java Runtime Environment:

The JRE software builds a runtime environment in which Java programs can be executed. The JRE is the on-disk system that takes your Java code, combines it with the needed libraries, and starts the JVM to execute it. The JRE contains libraries and software needed by your Java programs to run. JRE is a part of JDK (which we will study later) but can be downloaded separately.

Java Development Kit:

The Java Development Kit (JDK) is a software development environment used to develop Java applications and applets. It contains JRE and several development tools, an interpreter/loader (java), a compiler (javac), an archiver (jar), a documentation generator (javadoc) accompanied with another tool.

7) Why is Java not a pure object oriented language?

Java supports primitive data types - byte, boolean, char, short, int, float, long, and double and hence it is not a pure **object oriented language**

8) What are the differences between C++ and Java?

Comparison Index	C++	Java
Platform-independent	C++ is platform-dependent.	Java is platform-independent.
Mainly used for	C++ is mainly used for system programming.	Java is mainly used for application programming. It is widely used in window, web-based, enterprise and mobile applications.
Design Goal	C++ was designed for systems and applications programming. It was an extension of C programming language.	Java was designed and created as an interpreter for printing systems but later extended as a support network computing. It was designed with a goal of being easy to use and accessible to a broader audience.
Goto	C++ supports the goto statement.	Java doesn't support the goto statement.
Multiple inheritance	C++ supports multiple inheritance.	Java doesn't support multiple inheritance through class. It can be achieved by interfaces in java.
Operator Overloading	C++ supports operator overloading.	Java doesn't support operator overloading.
Pointers	C++ supports pointers. You can write pointer program in C++.	Java supports pointer internally. However, you can't write the pointer program in java.

		It means java has restricted pointer support in Java.
Compiler and Interpreter	C++ uses compiler only. C++ is compiled and run using the compiler which converts source code into machine code so, C++ is platform dependent.	Java uses compiler and interpreter both. Java source code is converted into bytecode at compilation time. The interpreter executes this bytecode at runtime and produces output. Java is interpreted that is why it is platform independent.
Call by Value and Call by reference	C++ supports both call by value and call by reference.	Java supports call by value only. There is no call by reference in java.
Structure and Union	C++ supports structures and unions.	Java doesn't support structures and unions.
Thread Support	C++ doesn't have built-in support for threads. It relies on third-party libraries for thread support.	Java has built-in thread support.
Documentation comment	C++ doesn't support documentation comment.	Java supports documentation comment (/** ... */) to create documentation for java source code.
Virtual Keyword	C++ supports virtual keyword so that we can decide whether or not override a function.	Java has no virtual keyword. We can override all non-static methods by default. In other words, non-static methods are virtual by default.
unsigned right shift >>>	C++ doesn't support >>> operator.	Java supports unsigned right shift >>> operator that fills zero at the top for the negative numbers. For positive numbers, it works same like >> operator.
Inheritance Tree	C++ creates a new inheritance tree always.	Java uses a single inheritance tree always because all classes are the child of Object class in java. The object class is the root of the inheritance tree in java.
Hardware	C++ is nearer to hardware.	Java is not so interactive with hardware.
Object-oriented	C++ is an object-oriented language. However, in C language, single root hierarchy is not	Java is also an object-oriented language. However, everything (except fundamental types) is an object in Java. It is a single root

	possible.	hierarchy as everything gets derived from java.lang.Object.

9) List the features of Java Programming language.

There are the following features in Java Programming Language.

- **Simple:** Java is easy to learn. The syntax of Java is based on C++ which makes easier to write the program in it.

- **Object-Oriented:** Java follows the object-oriented paradigm which allows us to maintain our code as the combination of different type of objects that incorporates both data and behavior.

- **Portable:** Java supports read-once-write-anywhere approach. We can execute the Java program on every machine. Java program (.java) is converted to bytecode (.class) which can be easily run on every machine.

- **Platform Independent:** Java is a platform independent programming language. It is different from other programming languages like C and C++ which needs a platform to be executed. Java comes with its platform on which its code is executed. Java doesn't depend upon the operating system to be executed.

- **Secured:** Java is secured because it doesn't use explicit pointers. Java also provides the concept of ByteCode and Exception handling which makes it more secured.

- **Robust:** Java is a strong programming language as it uses strong memory management. The concepts like Automatic garbage collection, Exception handling, etc. make it more robust.

- **Architecture Neutral:** Java is architectural neutral as it is not dependent on the architecture. In C, the size of data types may vary according to the architecture (32 bit or 64 bit) which doesn't exist in Java.

- **Interpreted:** Java uses the Just-in-time (JIT) interpreter along with the compiler for the program execution.

- **High Performance:** Java is faster than other traditional interpreted programming languages because Java bytecode is "close" to native code. It is still a little bit slower than a compiled language (e.g., C++).

- **Multithreaded:** We can write Java programs that deal with many tasks at once by defining multiple threads. The main advantage of multi-threading is that it doesn't occupy memory for each thread. It shares a common memory area. Threads are important for multi-media, Web applications, etc.

- **Distributed:** Java is distributed because it facilitates users to create distributed applications in Java. RMI and EJB are used for creating distributed applications. This feature of Java makes us able to access files by calling the methods from any machine on the internet.

- **Dynamic:** Java is a dynamic language. It supports dynamic loading of classes. It means classes are loaded on demand. It also supports functions from its native languages, i.e., C and C++.

10) What do you understand by Java virtual machine?

Java Virtual Machine is a virtual machine that enables the computer to run the Java program. JVM acts like a run-time engine which calls the main method present in the Java code. JVM is the specification which must be implemented in the computer system. The Java code is compiled by JVM to be a Bytecode which is machine independent and close to the native code.

11) Difference Between JDK, JRE, and JVM

Parameter	JDK	JRE	JVM
Full-Form	The JDK is an abbreviation for Java Development Kit.	The JRE is an abbreviation for Java Runtime Environment.	The JVM is an abbreviation for Java Virtual Machine.
Definition	The JDK (Java Development Kit) is a software development kit that develops applications in Java. Along with JRE, the JDK also consists of various development tools (Java Debugger, JavaDoc, compilers, etc.)	The Java Runtime Environment (JRE) is an implementation of JVM. It is a type of software package that provides class libraries of Java, JVM, and various other components for running the applications written in Java programming.	The Java Virtual Machine (JVM) is a platform-independent abstract machine that has three notions in the form of specifications. This document describes the requirement of JVM implementation.
Functionality	The JDK primarily assists in executing codes. It primarily functions in development.	JRE has a major responsibility for creating an environment for the execution of code.	JVM specifies all of the implementations. It is responsible for providing all of these implementations to the JRE.
Platform Dependency	The JDK is platform-dependent. It means that for every	JRE, just like JDK, is also platform-dependent. It means	The JVM is platform-independent. It means that you

	different platform, you require a different JDK.	that for every different platform, you require a different JRE.	won’t require a different JVM for every different platform.
Tools	Since JDK is primarily responsible for the development, it consists of various tools for debugging, monitoring, and developing java applications.	JRE, on the other hand, does not consist of any tool- like a debugger, compiler, etc. It rather contains various supporting files for JVM, and the class libraries that help JVM in running the program.	JVM does not consist of any tools for software development.
Implementation	**JDK** = Development Tools + JRE (Java Runtime Environment)	**JRE** = Libraries for running the application + JVM (Java Virtual Machine)	**JVM** = Only the runtime environment that helps in executing the Java bytecode.
Why Use It?	Why use JDK? Some crucial reasons to use JDK are: • It consists of various tools required for writing Java programs. • JDK also contains JRE for executing Java programs.	Why use JRE? Some crucial reasons to use JRE are: • If a user wants to run the Java applets, then they must install JRE on their system. • The JRE consists of class libraries along	Why use JVM? Some crucial reasons to use JVM are: • It provides its users with a platform-independent way for executing the Java source code. • JVM consists of

	• It includes an Appletviewer, Java application launcher, compiler, etc. • The compiler helps in converting the code written in Java into bytecodes. • The Java application launcher helps in opening a JRE. It then loads all of the necessary details and then executes all of its main methods.	with JVM and its supporting files. It has no other tools like a compiler or a debugger for Java development. • JRE uses crucial package classes like util, math, awt, lang, and various runtime libraries.	various tools, libraries, and multiple frameworks. • The JVM also comes with a Just-in-Time (JIT) compiler for converting the Java source code into a low-level machine language. Thus, it ultimately runs faster than any regular application. • Once you run the Java program, you can run JVM on any given platform to save your time.
Features	Features of JDK • Here are a few crucial features of JDK: • It has all the features that	Features of JRE • Here are a few crucial features of JRE: • It is a set of tools that	Features of JVM Here are a few crucial features of JVM: • The JVM enables a user

	JRE does. • JDK enables a user to handle multiple extensions in only one catch block. • It basically provides an environment for developing and executing the Java source code. • It has various development tools like the debugger, compiler, etc. • One can use the Diamond operator to specify a generic interface in place of writing the exact one. • Any user can easily install JDK on Unix, Mac, and Windows OS (Operating	actually helps the JVM to run. • The JRE also consists of deployment technology. It includes Java Plug-in and Java Web Start as well. • A developer can easily run a source code in JRE. But it does not allow them to write and compile the concerned Java program. • JRE also contains various integration libraries like the JDBC (Java Database Connectivity), JNDI (Java Naming and Directory Interface), RMI (Remote Method Invocation), and many more.	to run applications on their device or in a cloud environment. • It helps in converting the bytecode into machine-specific code. • JVM also provides some basic Java functions, such as garbage collection, security, memory management, and many more. • It uses a library along with the files given by JRE (Java Runtime Environment) for running the program. • Both JRE and JDK contain JVM. • It is easily customizable.

	Systems).	- It consists of the JVM and virtual machine client for Java HotSpot.	For instance, a user can feasibly allocate a maximum and minimum memory to it. - JVM can also execute a Java program line by line. It is thus also known as an interpreter. - JVM is also independent of the OS and hardware. It means that once a user writes a Java program, they can easily run it anywhere.

12) What is JIT compiler?

Just-In-Time(JIT) compiler: It is used to improve the performance. JIT compiles parts of the bytecode that have similar functionality at the same time, and hence reduces the amount of time needed for compilation. Here the term “compiler” refers to a translator from the instruction set of a Java virtual machine (JVM) to the instruction set of a specific CPU.

13) What is the platform?

A platform is the hardware or software environment in which a piece of software is executed. There are two types of platforms, software-based and hardware-based. Java provides the software-based platform.

14) What gives Java its 'write once and run anywhere' nature?

The bytecode. Java compiler converts the Java programs into the class file (Byte Code) which is the intermediate language between source code and machine code. This bytecode is not platform specific and can be executed on any computer.

15) What is classloader?

Classloader is a subsystem of JVM which is used to load class files. Whenever we run the java program, it is loaded first by the classloader. There are three built-in classloaders in Java.

1. **Bootstrap ClassLoader**: This is the first classloader which is the superclass of Extension classloader. It loads the rt.jar file which contains all class files of Java Standard Edition like java.lang package classes, java.net package classes, java.util package classes, java.io package classes, java.sql package classes, etc.
2. **Extension ClassLoader**: This is the child classloader of Bootstrap and parent classloader of System classloader. It loads the jar files located inside $JAVA_HOME/jre/lib/ext directory.
3. **System/Application ClassLoader**: This is the child classloader of Extension classloader. It loads the class files from the classpath. By default, the classpath is set to the current directory. You can change the classpath using "-cp" or "-classpath" switch. It is also known as Application classloader.

16) Is Empty .java file name a valid source file name?

Yes, Java allows to save our java file by **.java** only, we need to compile it by **javac .java** and run by **java classname** Let's take a simple example:

```
//save by .java only
class A{
```

```
public static void main(String args[]){
System.out.println("Hello java");
}
}
//compile by javac .java
//run by     java A
```

compile it by **javac .java**

run it by **java A**

17) Is delete, next, main, exit or null keyword in java?

No.

18) If I don't provide any arguments on the command line, then what will the value stored in the String array passed into the main() method, empty or NULL?

It is empty, but not null.

19) What if I write static public void instead of public static void?

The program compiles and runs correctly because the order of specifiers doesn't matter in Java.

20) What is the default value of the local variables?

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

21) What are the various access specifiers in Java?

In Java, access specifiers are the keywords which are used to define the access scope of the method, class, or a variable. In Java, there are four access specifiers given below.

- **Public** The classes, methods, or variables which are defined as public, can be accessed by any class or method.
- **Protected** Protected can be accessed by the class of the same package, or by the sub-class of this class, or within the same class.
- **Default** Default are accessible within the package only. By default, all the classes, methods, and variables are of default scope.
- **Private** The private class, methods, or variables defined as private can be accessed within the class only.

22) What is the purpose of static methods and variables?

The methods or variables defined as static are shared among all the objects of the class. The static is the part of the class and not of the object. The static variables are stored in the class area, and we do not need to create the object to access such variables. Therefore, static is used in the case, where we need to define variables or methods which are common to all the objects of the class.

For example, In the class simulating the collection of the students in a college, the name of the college is the common attribute to all the students. Therefore, the college name will be defined as **static**.

23) What is an object?

The Object is the real-time entity having some state and behavior. In Java, Object is an instance of the class having the instance variables as the state of the object and the methods as the behavior of the object. The object of a class can be created by using the **new** keyword.

24) What is the difference between an object-oriented programming language and object-based programming language?

There are the following basic differences between the object-oriented language and object-based language.

- Object-oriented languages follow all the concepts of OOPs whereas, the object-based language doesn't follow all the concepts of OOPs like inheritance and polymorphism.

- Object-oriented languages do not have the inbuilt objects whereas Object-based languages have the inbuilt objects, for example, JavaScript has window object.
- Examples of object-oriented programming are Java, C#, Smalltalk, etc. whereas the examples of object-based languages are JavaScript, VBScript, etc.

25) What will be the initial value of an object reference which is defined as an instance variable?

All object references are initialized to null in Java.

26)Constructors in Java

In Java, a constructor is a block of codes similar to the method. It is called when an instance of the class is created. At the time of calling constructor, memory for the object is allocated in the memory.

It is a special type of method which is used to initialize the object.Every time an object is created using the new() keyword, at least one constructor is called.

It calls a default constructor if there is no constructor available in the class. In such case, Java compiler provides a default constructor by default.

There are two types of constructors in Java: no-arg constructor, and parameterized constructor.

Rules for creating Java constructor

There are two rules defined for the constructor.

1. Constructor name must be the same as its class name
2. A Constructor must have no explicit return type
3. A Java constructor cannot be abstract, static, final, and synchronized

Types of Java constructors

There are two types of constructors in Java:

1. Default constructor (no-arg constructor)
2. Parameterized constructor

Java Default Constructor

A constructor is called "Default Constructor" when it doesn't have any parameter.

Syntax of default constructor:

In this example, we are creating the no-arg constructor in the Bike class. It will be invoked at the time of object creation.

```
<class_name>(){}
```

Example of default constructor

```
//Java Program to create and call a default constructor
class Bike1{
//creating a default constructor
Bike1(){System.out.println("Bike is created");}
//main method
public static void main(String args[]){
//calling a default constructor
Bike1 b=new Bike1();
}
}
```

Test it Now

Output:

```
Bike is created
```

> *Rule: If there is no constructor in a class, compiler automatically creates a default constructor.*

The image part with relationship ID rId4 was not found in the file.

Q) What is the purpose of a default constructor?

The default constructor is used to provide the default values to the object like 0, null, etc., depending on the type.

Example of default constructor that displays the default values

```
//Let us see another example of default constructor
//which displays the default values
class Student3{
int id;
String name;
//method to display the value of id and name
void display(){System.out.println(id+" "+name);}

public static void main(String args[]){
//creating objects
Student3 s1=new Student3();
Student3 s2=new Student3();
//displaying values of the object
s1.display();
s2.display();
}
}
```

Test it Now

Output:

```
0 null

0 null
```

Explanation: In the above class,you are not creating any constructor so compiler provides you a default constructor. Here 0 and null values are provided by default constructor.

Java Parameterized Constructor

A constructor which has a specific number of parameters is called a parameterized constructor.

Why use the parameterized constructor?

The parameterized constructor is used to provide different values to distinct objects. However, you can provide the same values also.

Example of parameterized constructor

In this example, we have created the constructor of Student class that have two parameters. We can have any number of parameters in the constructor.

```
//Java Program to demonstrate the use of the parameterized constructor.
class Student4{
    int id;
    String name;
    //creating a parameterized constructor
    Student4(int i,String n){
    id = i;
    name = n;
    }
    //method to display the values
    void display(){System.out.println(id+" "+name);}

    public static void main(String args[]){
    //creating objects and passing values
    Student4 s1 = new Student4(111,"Karan");
    Student4 s2 = new Student4(222,"Aryan");
    //calling method to display the values of object
    s1.display();
    s2.display();
   }
}
```

Test it Now

Output:

```
111 Karan
```

Constructor Overloading in Java

In Java, a constructor is just like a method but without return type. It can also be overloaded like Java methods.

Constructor overloading in Java is a technique of having more than one constructor with different parameter lists. They are arranged in a way that each constructor performs a different task. They are differentiated by the compiler by the number of parameters in the list and their types.

Example of Constructor Overloading

```
//Java program to overload constructors
class Student5{
   int id;
   String name;
   int age;
   //creating two arg constructor
   Student5(int i,String n){
   id = i;
   name = n;
   }
   //creating three arg constructor
   Student5(int i,String n,int a){
   id = i;
   name = n;
   age=a;
   }
   void display(){System.out.println(id+" "+name+" "+age);}

   public static void main(String args[]){
        Student5 s1 = new Student5(111,"Karan");
        Student5 s2 = new Student5(222,"Aryan",25);
        s1.display();
        s2.display();
       }
    }
```

Test it Now

Output:

```
111 Karan 0
222 Aryan 25
```

Difference between constructor and method in Java

There are many differences between constructors and methods.

Java Constructor	Java Method
A constructor is used to initialize the state of an object.	A method is used to expose the behavior of an object.
A constructor must not have a return type.	A method must have a return type.
The constructor is invoked implicitly.	The method is invoked explicitly.
The Java compiler provides a default constructor if you don't have any constructor in a class.	The method is not provided by the compiler in any case.
The constructor name must be same as the class name.	The method name may or may not be same as the class name.

27) Does constructor return any value?

Yes, it is the current class instance (You cannot use return type yet it returns a value).

28) Is there Constructor class in Java?

Yes.

29) What is the purpose of Constructor class?

Java provides a Constructor class which can be used to get the internal information of a constructor in the class. It is found in the java.lang.reflect package.

30)Is constructor inherited?

No, The constructor is not inherited.

31) Can you make a constructor final?

No, the constructor can't be final.

32) Can we overload the constructors?

Yes, the constructors can be overloaded by changing the number of arguments accepted by the constructor or by changing the data type of the parameters. Consider the following example.

```
class Test
{
    int i;
    public Test(int k)
    {
        i=k;
    }
    public Test(int k, int m)
    {
        System.out.println("Hi I am assigning the value max(k, m) to i");
        if(k>m)
        {
            i=k;
        }
        else
        {
            i=m;
        }
    }
}
public class Main
{
    public static void main (String args[])
    {
        Test test1 = new Test(10);
        Test test2 = new Test(12, 15);
        System.out.println(test1.i);
        System.out.println(test2.i);
    }
}

```

In the above program, The constructor Test is overloaded with another constructor. In the first call to the constructor, The constructor with one argument is called, and i will be initialized with the value 10. However, In the second call to the constructor, The constructor with the 2 arguments is called, and i will be initialized with the value 15.

33) What do you understand by copy constructor in Java?

There is no copy constructor in java. However, we can copy the values from one object to another like copy constructor in C++.

There are many ways to copy the values of one object into another in java. They are:

- By constructor
- By assigning the values of one object into another
- By clone() method of Object class

In this example, we are going to copy the values of one object into another using java constructor.

```
//Java program to initialize the values from one object to another
class Student6{
   int id;
   String name;
   //constructor to initialize integer and string
   Student6(int i,String n){
   id = i;
   name = n;
   }
   //constructor to initialize another object
   Student6(Student6 s){
   id = s.id;
   name =s.name;
   }
   void display(){System.out.println(id+" "+name);}

   public static void main(String args[]){
   Student6 s1 = new Student6(111,"Karan");
   Student6 s2 = new Student6(s1);
   s1.display();
   s2.display();
  }
}
```

Test it Now

Output:

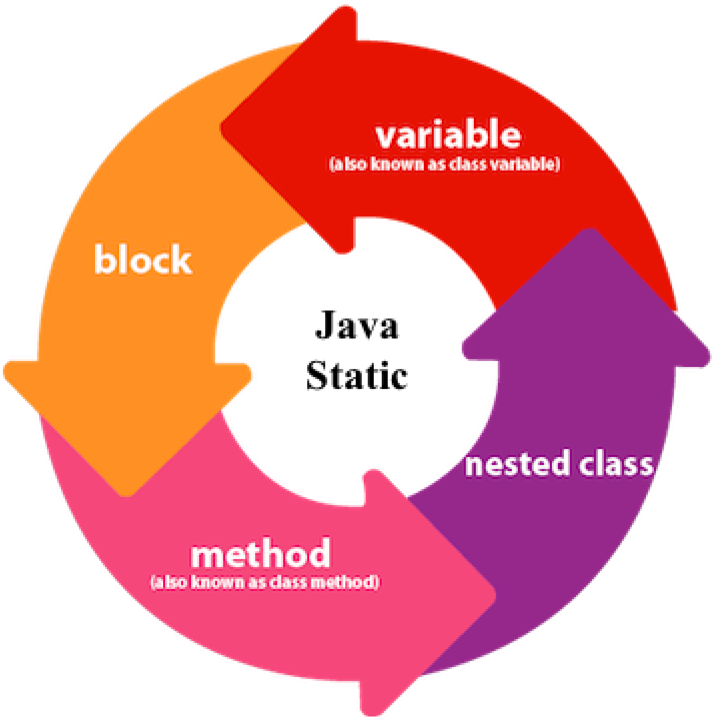
```
111 Karan
111 Karan
```

34) Java static keyword

The **static keyword** in Java is used for memory management mainly. We can apply static keyword with variables, methods, blocks and nested classes. The static keyword belongs to the class than an instance of the class.

The static can be:

1. Variable (also known as a class variable)
2. Method (also known as a class method)
3. Block
4. Nested class



1) Java static variable

If you declare any variable as static, it is known as a static variable.

- The static variable can be used to refer to the common property of all objects (which is not unique for each object), for example, the company name of employees, college name of students, etc.
- The static variable gets memory only once in the class area at the time of class loading.

Advantages of static variable

It makes your program **memory efficient** (i.e., it saves memory).

Understanding the problem without static variable

```
class Student{
   int rollno;
   String name;
   String college="ITS";
}
```

Suppose there are 500 students in my college, now all instance data members will get memory each time when the object is created. All students have its unique rollno and name, so instance data member is good in such case. Here, "college" refers to the common property of all objects. If we make it static, this field will get the memory only once.

> *Java static property is shared to all objects.*

Example of static variable

```
//Java Program to demonstrate the use of static variable
class Student{
  int rollno;//instance variable
  String name;
  static String college ="ITS";//static variable
  //constructor
  Student(int r, String n){
  rollno = r;
  name = n;
  }
  //method to display the values
  void display (){System.out.println(rollno+" "+name+" "+college);}
}
//Test class to show the values of objects
public class TestStaticVariable1{
 public static void main(String args[]){
 Student s1 = new Student(111,"Karan");
 Student s2 = new Student(222,"Aryan");
 //we can change the college of all objects by the single line of code
 //Student.college="BBDIT";
 s1.display();
 s2.display();
 }
}
```

Test it Now

Output:

```
111 Karan ITS
222 Aryan ITS
```

Program of the counter without static variable

In this example, we have created an instance variable named count which is incremented in the constructor. Since instance variable gets the memory at the time of object creation, each object will have the copy of the instance variable. If it is incremented, it won't reflect other objects. So each object will have the value 1 in the count variable.

```
//Java Program to demonstrate the use of an instance variable
//which get memory each time when we create an object of the class.
class Counter{
int count=0;//will get memory each time when the instance is created

Counter(){
count++;//incrementing value
System.out.println(count);
}

public static void main(String args[]){
//Creating objects
Counter c1=new Counter();
Counter c2=new Counter();
Counter c3=new Counter();
}
}
```

Test it Now

Output:

```
1
1
```

Program of counter by static variable

As we have mentioned above, static variable will get the memory only once, if any object changes the value of the static variable, it will retain its value.

```
//Java Program to illustrate the use of static variable which
//is shared with all objects.
class Counter2{
static int count=0;//will get memory only once and retain its value

Counter2(){
count++;//incrementing the value of static variable
System.out.println(count);
}

public static void main(String args[]){
//creating objects
Counter2 c1=new Counter2();
Counter2 c2=new Counter2();
Counter2 c3=new Counter2();
}
}
```

Test it Now

Output:

```
1
2
```

2) Java static method

If you apply static keyword with any method, it is known as static method.

- A static method belongs to the class rather than the object of a class.
- A static method can be invoked without the need for creating an instance of a class.
- A static method can access static data member and can change the value of it.

Example of static method

```
//Java Program to demonstrate the use of a static method.
class Student{
    int rollno;
    String name;
    static String college = "ITS";
    //static method to change the value of static variable
    static void change(){
    college = "BBDIT";
    }
    //constructor to initialize the variable
```

```
     Student(int r, String n){
     rollno = r;
     name = n;
     }
     //method to display values
     void display(){System.out.println(rollno+" "+name+" "+college);}
}
//Test class to create and display the values of object
public class TestStaticMethod{
    public static void main(String args[]){
    Student.change();//calling change method
    //creating objects
    Student s1 = new Student(111,"Karan");
    Student s2 = new Student(222,"Aryan");
    Student s3 = new Student(333,"Sonoo");
    //calling display method
    s1.display();
    s2.display();
    s3.display();
    }
}
```

Test it Now

```
Output:111 Karan BBDIT
       222 Aryan BBDIT
       333 Sonoo BBDIT
```

Another example of a static method that performs a normal calculation

```
//Java Program to get the cube of a given number using the static method

class Calculate{
    static int cube(int x){
    return x*x*x;
    }

    public static void main(String args[]){
    int result=Calculate.cube(5);
    System.out.println(result);
    }
}
```

Test it Now

```
Output:125
```

Restrictions for the static method

There are two main restrictions for the static method. They are:

1. The static method can not use non static data member or call non-static method directly.
2. this and super cannot be used in static context.

```
class A{
 int a=40;//non static

 public static void main(String args[]){
  System.out.println(a);
 }
}
```

Test it Now

```
Output:Compile Time Error
```

Q) Why is the Java main method static?

Ans) It is because the object is not required to call a static method. If it were a non-static method, JVM creates an object first then call main() method that will lead the problem of extra memory allocation.

3) Java static block

- Is used to initialize the static data member.
- It is executed before the main method at the time of classloading.

Example of static block

```
class A2{
  static{System.out.println("static block is invoked");}
  public static void main(String args[]){
   System.out.println("Hello main");
  }
}
```

Test it Now

```
Output:static block is invoked
       Hello main
```

Q) Can we execute a program without main() method?

Ans) No, one of the ways was the static block, but it was possible till JDK 1.6. Since JDK 1.7, it is not possible to execute a Java class without the main method.

```
class A3{
  static{
  System.out.println("static block is invoked");
  System.exit(0);
  }
}
```

Test it Now

Output:

```
static block is invoked
```

Since JDK 1.7 and above, output would be:

```
Error: Main method not found in class A3, please define the main method as:
   public static void main(String[] args)
or a JavaFX application class must extend javafx.application.Application
```

35) Why is the main method static?

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

36) Can we override the static methods?

No, we can't override static methods.

37) Can we execute a program without main() method?

Ans) No, It was possible before JDK 1.7 using the static block. Since JDK 1.7, it is not possible.

38) What if the static modifier is removed from the signature of the main method?

Program compiles. However, at runtime, It throws an error "NoSuchMethodError."

39) What is the difference between static (class) method and instance method?

static or class method	instance method
1)A method that is declared as static is known as the static method.	A method that is not declared as static is known as the instance method.
2)We don't need to create the objects to call the static methods.	The object is required to call the instance methods.

3)Non-static (instance) members cannot be accessed in the static context (static method, static block, and static nested class) directly.	Static and non-static variables both can be accessed in instance methods.
4)For example: public static int cube(int n){ return n*n*n;}	For example: public void msg(){...}.

40) Can we make constructors static?

As we know that the static context (method, block, or variable) belongs to the class, not the object. Since Constructors are invoked only when the object is created, there is no sense to make the constructors static. However, if you try to do so, the compiler will show the compiler error.

41) Can we make the abstract methods static in Java?

In Java, if we make the abstract methods static, It will become the part of the class, and we can directly call it which is unnecessary. Calling an undefined method is completely useless therefore it is not allowed.

42) Can we declare the static variables and methods in an abstract class?

Yes, we can declare static variables and methods in an abstract method. As we know that there is no requirement to make the object to access the static context, therefore, we can access the static context declared inside the abstract class by using the name of the abstract class. Consider the following example.

```
abstract class Test
{
    static int i = 102;
    static void TestMethod()
    {
        System.out.println("hi !! I am good !!");
    }
}
public class TestClass extends Test
{
    public static void main (String args[])
    {
        Test.TestMethod();
        System.out.println("i = "+Test.i);
    }
}
```

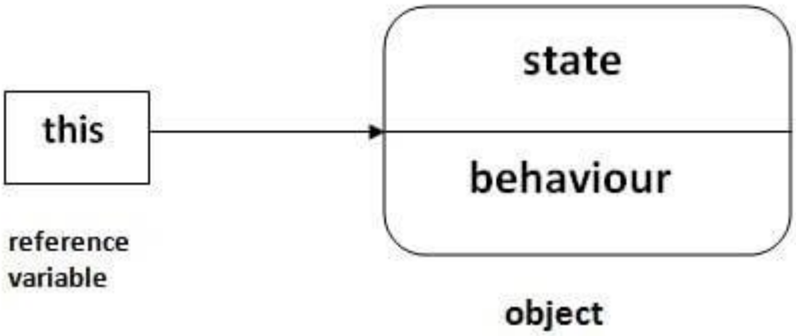
Output

```
hi !! I am good !!
i = 102
```

Core Java - OOPs Concepts: Inheritance Interview Questions

43) What is this keyword in java?

The **this** keyword is a reference variable that refers to the current object. There are the various uses of this keyword in Java. It can be used to refer to current class properties such as instance methods, variable, constructors, etc. It can also be passed as an argument into the methods or constructors. It can also be returned from the method as the current class instance.



44) What are the main uses of this keyword?

There are the following uses of **this** keyword.

- **this** can be used to refer to the current class instance variable.

- **this** can be used to invoke current class method (implicitly)
- **this()** can be used to invoke the current class constructor.
- **this** can be passed as an argument in the method call.
- **this** can be passed as an argument in the constructor call.
- **this** can be used to return the current class instance from the method

45) Can this keyword be used to refer static members?

Yes, It is possible to use this keyword to refer static members because this is just a reference variable which refers to the current class object. However, as we know that, it is unnecessary to access static variables through objects, therefore, it is not the best practice to use this to refer static members. Consider the following example.

```
public class Test
{
   static int i = 10;
   public Test ()
   {
      System.out.println(this.i);
   }
   public static void main (String args[])
   {
      Test t = new Test();
   }
}
```

Output

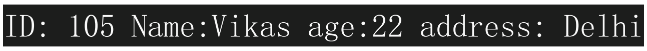
```
10
```

46) How can constructor chaining be done using this keyword?

Constructor chaining enables us to call one constructor from another constructor of the class with respect to the current class object. We can use this keyword to perform constructor chaining within the same class. Consider the following example which illustrates how can we use this keyword to achieve constructor chaining.

```java
public class Employee
{
    int id,age;
    String name, address;
    public Employee (int age)
    {
        this.age = age;
    }
    public Employee(int id, int age)
    {
        this(age);
        this.id = id;
    }
    public Employee(int id, int age, String name, String address)
    {
        this(id, age);
        this.name = name;
        this.address = address;
    }
    public static void main (String args[])
    {
        Employee emp = new Employee(105, 22, "Vikas", "Delhi");
        System.out.println("ID: "+emp.id+" Name:"+emp.name+" age:"+emp.age+" address: "+emp.address);
    }

}
```

Output



```
ID: 105 Name:Vikas age:22 address: Delhi
```

47) What is the Inheritance?

Inheritance is a mechanism by which one object acquires all the properties and behavior of another object of another class. It is used for Code Reusability and Method Overriding. The idea behind inheritance in Java is that you can create new classes that are built upon existing classes. When you inherit from an existing class, you can reuse methods and fields of the parent class. Moreover, you can add new methods and fields in your current class also. Inheritance represents the IS-A relationship which is also known as a parent-child relationship.

There are five types of inheritance in Java.

- Single-level inheritance
- Multi-level inheritance
- Multiple Inheritance
- Hierarchical Inheritance
- Hybrid Inheritance

Multiple inheritance is not supported in Java through class.

Inheritance in Java

- Inheritance
- Types of Inheritance
- Why multiple inheritance is not possible in Java in case of class?

Inheritance in Java is a mechanism in which one object acquires all the properties and behaviors of a parent object. It is an important part of OOPs (Object Oriented programming system).

The idea behind inheritance in Java is that you can create new classes that are built upon existing classes. When you inherit from an existing class, you can reuse methods and fields of the parent class. Moreover, you can add new methods and fields in your current class also.

Inheritance represents the **IS-A relationship** which is also known as a parent-child relationship.

Why use inheritance in java

- For Method Overriding (so runtime polymorphism can be achieved).
- For Code Reusability.

Terms used in Inheritance

- **Class:** A class is a group of objects which have common properties. It is a template or blueprint from which objects are created.
- **Sub Class/Child Class:** Subclass is a class which inherits the other class. It is also called a derived class, extended class, or child class.
- **Super Class/Parent Class:** Superclass is the class from where a subclass inherits the features. It is also called a base class or a parent class.
- **Reusability:** As the name specifies, reusability is a mechanism which facilitates you to reuse the fields and methods of the existing class when you create a new class. You can use the same fields and methods already defined in the previous class.

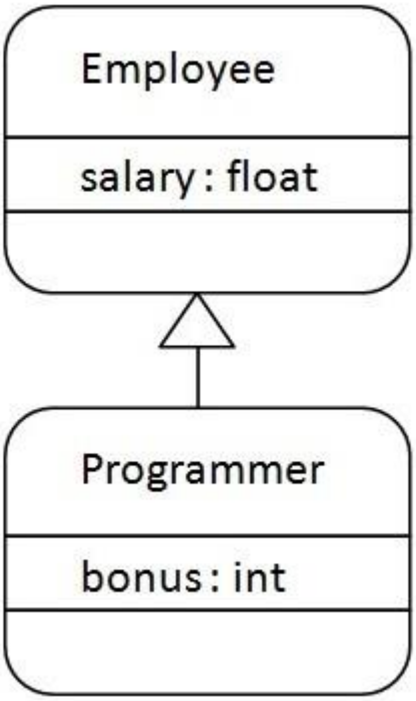
The syntax of Java Inheritance

```java
class Subclass-name extends Superclass-name
{
   //methods and fields
}
```

The **extends keyword** indicates that you are making a new class that derives from an existing class. The meaning of "extends" is to increase the functionality.

In the terminology of Java, a class which is inherited is called a parent or superclass, and the new class is called child or subclass.

Java Inheritance Example



As displayed in the above figure, Programmer is the subclass and Employee is the superclass. The relationship between the two classes is **Programmer IS-A Employee**. It means that Programmer is a type of Employee.

```
class Employee{
 float salary=40000;
}
class Programmer extends Employee{
 int bonus=10000;
 public static void main(String args[]){
   Programmer p=new Programmer();
   System.out.println("Programmer salary is:"+p.salary);
   System.out.println("Bonus of Programmer is:"+p.bonus);
}
}
```

Test it Now

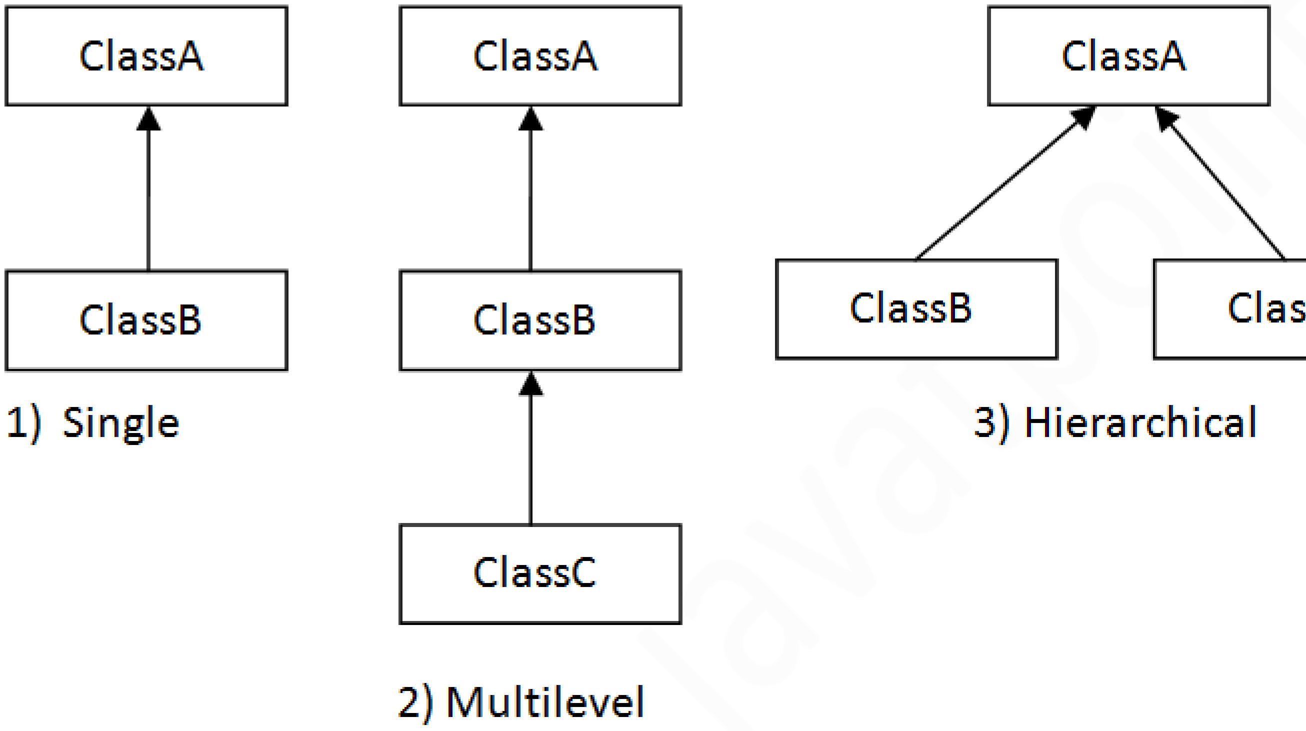
```
Programmer salary is:40000.0
Bonus of programmer is:10000
```

In the above example, Programmer object can access the field of own class as well as of Employee class i.e. code reusability.

Types of inheritance in java

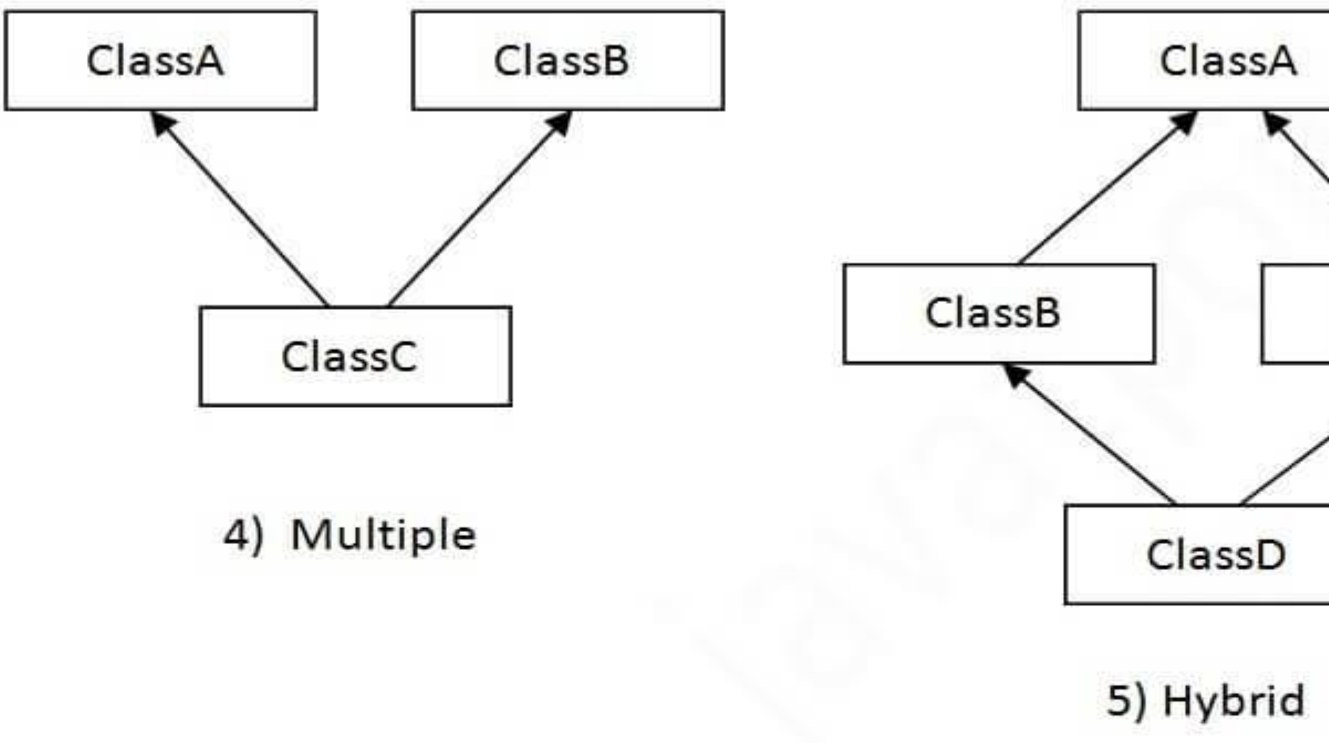
On the basis of class, there can be three types of inheritance in java: single, multilevel and hierarchical.

In java programming, multiple and hybrid inheritance is supported through interface only. We will learn about interfaces later.



Note: Multiple inheritance is not supported in Java through class.

When one class inherits multiple classes, it is known as multiple inheritance. For Example:



Single Inheritance Example

When a class inherits another class, it is known as a single inheritance. In the example given below, Dog class inherits the Animal class, so there is the single inheritance.

File: TestInheritance.java

```
class Animal{
void eat(){System.out.println("eating...");}
}
class Dog extends Animal{
void bark(){System.out.println("barking...");}
}
class TestInheritance{
public static void main(String args[]){
Dog d=new Dog();
d.bark();
d.eat();
```

```java
}}
```

Output:

```
barking...
eating...
```

Multilevel Inheritance Example

When there is a chain of inheritance, it is known as multilevel inheritance. As you can see in the example given below, BabyDog class inherits the Dog class which again inherits the Animal class, so there is a multilevel inheritance.

File: TestInheritance2.java

```java
class Animal{
void eat(){System.out.println("eating...");}
}
class Dog extends Animal{
void bark(){System.out.println("barking...");}
}
class BabyDog extends Dog{
void weep(){System.out.println("weeping...");}
}
class TestInheritance2{
public static void main(String args[]){
BabyDog d=new BabyDog();
d.weep();
d.bark();
d.eat();
}}
```

Output:

```
weeping...
barking...
```

```
eating...
```

Hierarchical Inheritance Example

When two or more classes inherits a single class, it is known as hierarchical inheritance. In the example given below, Dog and Cat classes inherits the Animal class, so there is hierarchical inheritance.

File: TestInheritance3.java

```java
class Animal{
void eat(){System.out.println("eating...");}
}
class Dog extends Animal{
void bark(){System.out.println("barking...");}
}
class Cat extends Animal{
void meow(){System.out.println("meowing...");}
}
class TestInheritance3{
public static void main(String args[]){
Cat c=new Cat();
c.meow();
c.eat();
//c.bark();//C.T.Error
}}
```

Output:

```
meowing...
eating...
```

Q) Why multiple inheritance is not supported in java?

To reduce the complexity and simplify the language, multiple inheritance is not supported in java.

Consider a scenario where A, B, and C are three classes. The C class inherits A and B classes. If A and B classes have the same method and you call it from child class object, there will be ambiguity to call the method of A or B class.

Since compile-time errors are better than runtime errors, Java renders compile-time error if you inherit 2 classes. So whether you have same method or different, there will be compile time error.

```java
class A{
void msg(){System.out.println("Hello");}
}
class B{
void msg(){System.out.println("Welcome");}
}
class C extends A,B{//suppose if it were

 public static void main(String args[]){
   C obj=new C();
   obj.msg();//Now which msg() method would be invoked?
}
}
```

Test it Now

```
Compile Time Error
```

48) Why is Inheritance used in Java?

There are various advantages of using inheritance in Java that is given below.

- Inheritance provides code reusability. The derived class does not need to redefine the method of base class unless it needs to provide the specific implementation of the method.
- Runtime polymorphism cannot be achieved without using inheritance.
- We can simulate the inheritance of classes with the real-time objects which makes OOPs more realistic.
- Inheritance provides data hiding. The base class can hide some data from the derived class by making it private.
- Method overriding cannot be achieved without inheritance. By method overriding, we can give a specific implementation of some basic method contained by the base class.

48) Which class is the superclass for all the classes?

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

49) What is aggregation?

Aggregation can be defined as the relationship between two classes where the aggregate class contains a reference to the class it owns. Aggregation is best described as a **has-a** relationship. For example, The aggregate class Employee having various fields such as age, name, and salary also contains an object of Address class having various fields such as Address-Line 1, City, State, and pin-code. In other words, we can say that Employee (class) has an object of Address class. Consider the following example.

Address.java

```
public class Address {
String city,state,country;

public Address(String city, String state, String country) {
    this.city = city;
    this.state = state;
    this.country = country;
}

}
```

Employee.java

```
public class Emp {
int id;
String name;
Address address;

public Emp(int id, String name,Address address) {
    this.id = id;
    this.name = name;
    this.address=address;
}

void display(){
System.out.println(id+" "+name);
System.out.println(address.city+" "+address.state+" "+address.country);

}

public static void main(String[] args) {
Address address1=new Address("gzb","UP","india");
Address address2=new Address("gno","UP","india");

Emp e=new Emp(111,"varun",address1);
Emp e2=new Emp(112,"arun",address2);

e.display();
e2.display();

}
}
```

Output

```
111 varun
gzb UP india
112 arun
gno UP india
```

50) What is composition?

Holding the reference of a class within some other class is known as composition. When an object contains the other object, if the contained object cannot exist without the existence of container object, then it is called composition. In other words, we can say that composition is the particular case of aggregation which represents a stronger relationship between two objects. Example: A class contains students. A student cannot exist without a class. There exists composition between class and students.

51) What is the difference between aggregation and composition?

Aggregation represents the weak relationship whereas composition represents the strong relationship. For example, the bike has an indicator (aggregation), but the bike has an engine (composition).

52) Why does Java not support pointers?

The pointer is a variable that refers to the memory address. They are not used in Java because they are unsafe(unsecured) and complex to understand.

53) What is super in java?

The **super** keyword in Java is a reference variable that is used to refer to the immediate parent class object. Whenever you create the instance of the subclass, an instance of the parent class is created implicitly which is referred by super reference variable. The super() is called in the class constructor implicitly by the compiler if there is no super or this.

```
class Animal{
Animal(){System.out.println("animal is created");}
}
class Dog extends Animal{
Dog(){
System.out.println("dog is created");
}
}
class TestSuper4{
public static void main(String args[]){
Dog d=new Dog();
}
}
```

Test it Now

Output:

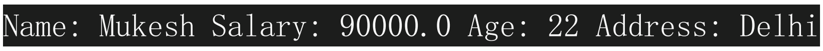
```
animal is created
dog is created
```

54) How can constructor chaining be done by using the super keyword?

```
class Person
{
    String name,address;
    int age;
    public Person(int age, String name, String address)
    {
        this.age = age;
        this.name = name;
        this.address = address;
    }
}
```

```
class Employee extends Person
{
    float salary;
    public Employee(int age, String name, String address, float salary)
    {
        super(age,name,address);
        this.salary = salary;
    }
}
public class Test
{
    public static void main (String args[])
    {
        Employee e = new Employee(22, "Mukesh", "Delhi", 90000);
        System.out.println("Name: "+e.name+" Salary: "+e.salary+" Age: "+e.age+" Address: "+e.address);
    }
}
```

Output



```
Name: Mukesh Salary: 90000.0 Age: 22 Address: Delhi
```

55) What are the main uses of the super keyword?

There are the following uses of super keyword.

- super can be used to refer to the immediate parent class instance variable.
- super can be used to invoke the immediate parent class method.
- super() can be used to invoke immediate parent class constructor.

56) What are the differences between this and super keyword?

There are the following differences between this and super keyword.

- The super keyword always points to the parent class contexts whereas this keyword always points to the current class context.
- The super keyword is primarily used for initializing the base class variables within the derived class constructor whereas this keyword primarily used to differentiate between local and instance variables when passed in the class constructor.
- The super and this must be the first statement inside constructor otherwise the compiler will throw an error.

57) Can you use this() and super() both in a constructor?

No, because this() and super() must be the first statement in the class constructor.

Example:

```
public class Test{
    Test()
     {
         super();
         this();
         System.out.println("Test class object is created");
     }
     public static void main(String []args){
     Test t = new Test();
     }
}
```

Output:

```
Test.java:5: error: call to this must be first statement in
constructor
```

58)What is object cloning?

The object cloning is used to create the exact copy of an object. The clone() method of the Object class is used to clone an object. The **java.lang.Cloneable** interface must be implemented by the class whose object clone we want to create. If we don't implement Cloneable interface, clone() method generates CloneNotSupportedException.

```
protected Object clone() throws CloneNotSupportedException
```

Core Java - OOPs Concepts: Method Overloading Interview Questions

59) What is method overloading?

Method overloading is the polymorphism technique which allows us to create multiple methods with the same name but different signature. We can achieve method overloading in two ways.

- By Changing the number of arguments
- By Changing the data type of arguments

Method overloading increases the readability of the program. Method overloading is performed to figure out the program quickly.

60) Why is method overloading not possible by changing the return type in java?

In Java, method overloading is not possible by changing the return type of the program due to avoid the ambiguity.

```
class Adder{
static int add(int a,int b){return a+b;}
static double add(int a,int b){return a+b;}
}
class TestOverloading3{
public static void main(String[] args){
System.out.println(Adder.add(11,11));//ambiguity
}}
```

Test it Now

Output:

```
Compile Time Error: method add(int, int) is already defined in class
Adder
```

61) Can we overload the methods by making them static?

No, We cannot overload the methods by just applying the static keyword to them(number of parameters and types are the same). Consider the following example.

```java
public class Animal
{
    void consume(int a)
    {
        System.out.println(a+" consumed!!");
    }
    static void consume(int a)
    {
        System.out.println("consumed static "+a);
    }
    public static void main (String args[])
    {
        Animal a = new Animal();
        a.consume(10);
        Animal.consume(20);
    }
}
```

Output

```
Animal.java:7: error: method consume(int) is already defined in class
Animal
    static void consume(int a)
```

```
                ^
Animal.java:15: error: non-static method consume(int) cannot be referenced from a static context
        Animal.consume(20);
              ^
2 errors
```

62) Can we overload the main() method?

Yes, we can have any number of main methods in a Java program by using method overloading.

Core Java - OOPs Concepts: Method Overriding Interview Questions

63) What is method overriding:

If a subclass provides a specific implementation of a method that is already provided by its parent class, it is known as Method Overriding. It is used for runtime polymorphism and to implement the interface methods.

Rules for Method overriding

- The method must have the same name as in the parent class.
- The method must have the same signature as in the parent class.
- Two classes must have an IS-A relationship between them.

64) Can we override the static method?

No, you can't override the static method because they are the part of the class, not the object.

65) Why can we not override static method?

It is because the static method is the part of the class, and it is bound with class whereas instance method is bound with the object, and static gets memory in class area, and instance gets memory in a heap.

66) Can we override the overloaded method?

Yes.

67) Difference between method Overloading and Overriding.

Method Overloading	Method Overriding
1) Method overloading increases the readability of the program.	Method overriding provides the specific implementation of the method that is already provided by its superclass.
2) Method overloading occurs within the class.	Method overriding occurs in two classes that have IS-A relationship between them.
3) In this case, the parameters must be different.	In this case, the parameters must be the same.

68) Can we override the private methods?

No, we cannot override the private methods because the scope of private methods is limited to the class and we cannot access them outside of the class.

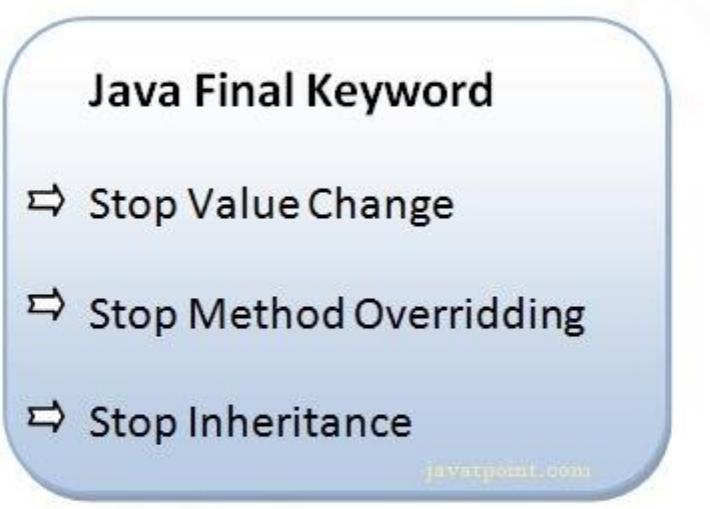
69) Can you have virtual functions in Java?

Yes, all functions in Java are virtual by default.

Core Java - OOPs Concepts: final keyword Interview Questions

70) What is the final variable?

In Java, the final variable is used to restrict the user from updating it. If we initialize the final variable, we can't change its value. In other words, we can say that the final variable once assigned to a value, can never be changed after that. The final variable which is not assigned to any value can only be assigned through the class constructor.



Java Final Keyword

⇨ Stop Value Change

⇨ Stop Method Overridding

⇨ Stop Inheritance

```
class Bike9{
 final int speedlimit=90;//final variable
 void run(){
  speedlimit=400;
 }
 public static void main(String args[]){
 Bike9 obj=new  Bike9();
 obj.run();
 }
}//end of class
```

Test it Now

```
Output:Compile Time Error
```

More Details.

71) What is the final method?

If we change any method to a final method, we can't override it. More Details.

```
class Bike{
  final void run(){System.out.println("running");}
}

class Honda extends Bike{
   void run(){System.out.println("running safely with 100kmph");}

   public static void main(String args[]){
   Honda honda= new Honda();
   honda.run();
   }
}
```

Test it Now

```
Output:Compile Time Error
```

72) What is the final class?

If we make any class final, we can't inherit it into any of the subclasses.

```
final class Bike{}

class Honda1 extends Bike{
  void run(){System.out.println("running safely with 100kmph");}

  public static void main(String args[]){
  Honda1 honda= new Honda1();
  honda.run();
  }
}
```

Test it Now

```
Output:Compile Time Error
```

More Details.

73) What is the final blank variable?

A final variable, not initialized at the time of declaration, is known as the final blank variable. We can't initialize the final blank variable directly. Instead, we have to initialize it by using the class constructor. It is useful in the case when the user has some data which must not be changed by others, for example, PAN Number. Consider the following example:

```
class Student{
int id;
String name;
final String PAN_CARD_NUMBER;
...
}
```

More Details.

74) Can we initialize the final blank variable?

Yes, if it is not static, we can initialize it in the constructor. If it is static blank final variable, it can be initialized only in the static block. More Details.

75) Can you declare the main method as final?

Yes, We can declare the main method as public static final void main(String[] args){}.

76) Can we declare a constructor as final?

The constructor can never be declared as final because it is never inherited. Constructors are not ordinary methods; therefore, there is no sense to declare constructors as final. However, if you try to do so, The compiler will throw an error.

77) Can we declare an interface as final?

No, we cannot declare an interface as final because the interface must be implemented by some class to provide its definition. Therefore, there is no sense to make an interface final. However, if you try to do so, the compiler will show an error.

78) What is the difference between the final method and abstract method?

The main difference between the final method and abstract method is that the abstract method cannot be final as we need to override them in the subclass to give its definition.

Core Java - OOPs: Polymorphism Interview Questions

79) What is the difference between compile-time polymorphism and runtime polymorphism?

SN	compile-time polymorphism	Runtime polymorphism
1	In compile-time polymorphism, call to a method is resolved at compile-time.	In runtime polymorphism, call to an overridden method is resolved at runtime.
2	It is also known as static binding, early binding, or overloading.	It is also known as dynamic binding, late binding, overriding, or dynamic method dispatch.
3	Overloading is a way to achieve compile-time polymorphism in which, we can define multiple methods or constructors with different signatures.	Overriding is a way to achieve runtime polymorphism in which, we can redefine some particular method or variable in the derived class. By using overriding, we can give some specific implementation to the base class properties in the derived class.
4	It provides fast execution because the	It provides slower execution as compare to compile-

	type of an object is determined at compile-time.	time because the type of an object is determined at run-time.
5	Compile-time polymorphism provides less flexibility because all the things are resolved at compile-time.	Run-time polymorphism provides more flexibility because all the things are resolved at runtime.

80) What is Runtime Polymorphism?

Runtime polymorphism or dynamic method dispatch is a process in which a call to an overridden method is resolved at runtime rather than at compile-time. In this process, an overridden method is called through the reference variable of a superclass. The determination of the method to be called is based on the object being referred to by the reference variable.

```java
class Bike{
  void run(){System.out.println("running");}
}
class Splendor extends Bike{
  void run(){System.out.println("running safely with 60km");}
  public static void main(String args[]){
    Bike b = new Splendor();//upcasting
    b.run();
  }
}
```

Test it Now

Output:

```
running safely with 60km.
```

In this process, an overridden method is called through the reference variable of a superclass. The determination of the method to be called is based on the object being referred to by the reference variable.

81) Can you achieve Runtime Polymorphism by data members?

No, because method overriding is used to achieve runtime polymorphism and data members cannot be overridden. We can override the member functions but not the data members. Consider the example given below.

```
class Bike{
 int speedlimit=90;
}
class Honda3 extends Bike{
 int speedlimit=150;
 public static void main(String args[]){
  Bike obj=new Honda3();
  System.out.println(obj.speedlimit);//90
   }
```

Test it Now

Output:

```
90
```

82) What is the difference between static binding and dynamic binding?

In case of the static binding, the type of the object is determined at compile-time whereas, in the dynamic binding, the type of the object is determined at runtime.

Static Binding

```
class Dog{
 private void eat(){System.out.println("dog is eating...");}

 public static void main(String args[]){
  Dog d1=new Dog();
  d1.eat();
 }
}
```

Dynamic Binding

```java
class Animal{
 void eat(){System.out.println("animal is eating...");}
}

class Dog extends Animal{
 void eat(){System.out.println("dog is eating...");}

 public static void main(String args[]){
  Animal a=new Dog();
  a.eat();
 }
}
```

83) What is Java instanceOf operator?

The instanceof in Java is also known as type comparison operator because it compares the instance with type. It returns either true or false. If we apply the instanceof operator with any variable that has a null value, it returns false. Consider the following example.

```java
class Simple1{
 public static void main(String args[]){
 Simple1 s=new Simple1();
 System.out.println(s instanceof Simple1);//true
 }
}
```

Test it Now

Output

```
true
```

An object of subclass type is also a type of parent class. For example, if Dog extends Animal then object of Dog can be referred by either Dog or Animal class.

Core Java - OOPs Concepts: Abstraction Interview Questions

84) What is the abstraction?

Abstraction is a process of hiding the implementation details and showing only functionality to the user. It displays just the essential things to the user and hides the internal information, for example, sending SMS where you type the text and send the message. You don't know the internal processing about the message delivery. Abstraction enables you to focus on what the object does instead of how it does it. Abstraction lets you focus on what the object does instead of how it does it.

In Java, there are two ways to achieve the abstraction.

- Abstract Class
- Interface

More details.

85) What is the difference between abstraction and encapsulation?

Abstraction hides the implementation details whereas encapsulation wraps code and data into a single unit.

More details.

86) What is the abstract class?

A class that is declared as abstract is known as an abstract class. It needs to be extended and its method implemented. It cannot be instantiated. It can have abstract methods, non-abstract methods, constructors, and static methods. It can also have the final methods which will force the subclass not to change the body of the method. Consider the following example.

```
abstract class Bike{
  abstract void run();
}
class Honda4 extends Bike{
void run(){System.out.println("running safely");}
public static void main(String args[]){
 Bike obj = new Honda4();
 obj.run();
}
}
```

Test it Now

Output

```
running safely
```

More details.

87) Can there be an abstract method without an abstract class?

No, if there is an abstract method in a class, that class must be abstract.

88) Can you use abstract and final both with a method?

No, because we need to override the abstract method to provide its implementation, whereas we can't override the final method.

89) Is it possible to instantiate the abstract class?

No, the abstract class can never be instantiated even if it contains a constructor and all of its methods are implemented.

90) What is the interface?

The interface is a blueprint for a class that has static constants and abstract methods. It can be used to achieve full abstraction and multiple inheritance. It is a mechanism to achieve abstraction. There can be only abstract methods in the Java interface, not method body. It is used to achieve abstraction and multiple inheritance in Java. In other words, you can say that interfaces can have abstract methods and variables. Java Interface also represents the IS-A relationship. It cannot be instantiated just like the abstract class. However, we need to implement it to define its methods. Since Java 8, we can have the default, static, and private methods in an interface.

91) Can you declare an interface method static?

No, because methods of an interface are abstract by default, and we can not use static and abstract together.

92) Can the Interface be final?

No, because an interface needs to be implemented by the other class and if it is final, it can't be implemented by any class.

93) What is a marker interface?

A Marker interface can be defined as the interface which has no data member and member functions. For example, Serializable, Cloneable are marker interfaces. The marker interface can be declared as follows.

```java
public interface Serializable{
}
```

94) What are the differences between abstract class and interface?

Abstract class	Interface

An abstract class can have a method body (non-abstract methods).	The interface has only abstract methods.
An abstract class can have instance variables.	An interface cannot have instance variables.
An abstract class can have the constructor.	The interface cannot have the constructor.
An abstract class can have static methods.	The interface cannot have static methods.
You can extend one abstract class.	You can implement multiple interfaces.
The abstract class **can provide the implementation of the interface**.	The Interface **can't provide the implementation of the abstract class**.
The **abstract keyword** is used to declare an abstract class.	The **interface keyword** is used to declare an interface.
An **abstract class** can extend another Java class and implement multiple Java interfaces.	An **interface** can extend another Java interface only.
An **abstract class** can be extended using keyword **extends**	An **interface class** can be implemented using keyword **implements**
A Java **abstract class** can have class members like private, protected, etc.	Members of a Java interface are public by default.

Example:
public abstract class Shape{
public abstract void draw();
}
	Example:
	public interface Drawable{
	void draw();
	}

95) Can we define private and protected modifiers for the members in interfaces?

No, they are implicitly public.

96) When can an object reference be cast to an interface reference?

An object reference can be cast to an interface reference when the object implements the referenced interface.

97) How to make a read-only class in Java?

A class can be made read-only by making all of the fields private. The read-only class will have only getter methods which return the private property of the class to the main method. We cannot modify this property because there is no setter method available in the class. Consider the following example.

```
  //A Java class which has only getter methods.
public class Student{
//private data member
private String college="AKG";
//getter method for college
public String getCollege(){
return college;
}
}
```

98) How to make a write-only class in Java?

A class can be made write-only by making all of the fields private. The write-only class will have only setter methods which set the value passed from the main method to the private fields. We cannot read the properties of the class because there is no getter method in this class. Consider the following example.

```
  //A Java class which has only setter methods.
public class Student{
//private data member
private String college;
//getter method for college
public void setCollege(String college){
this.college=college;
}
}
```

99) What are the advantages of Encapsulation in Java?

There are the following advantages of Encapsulation in Java?

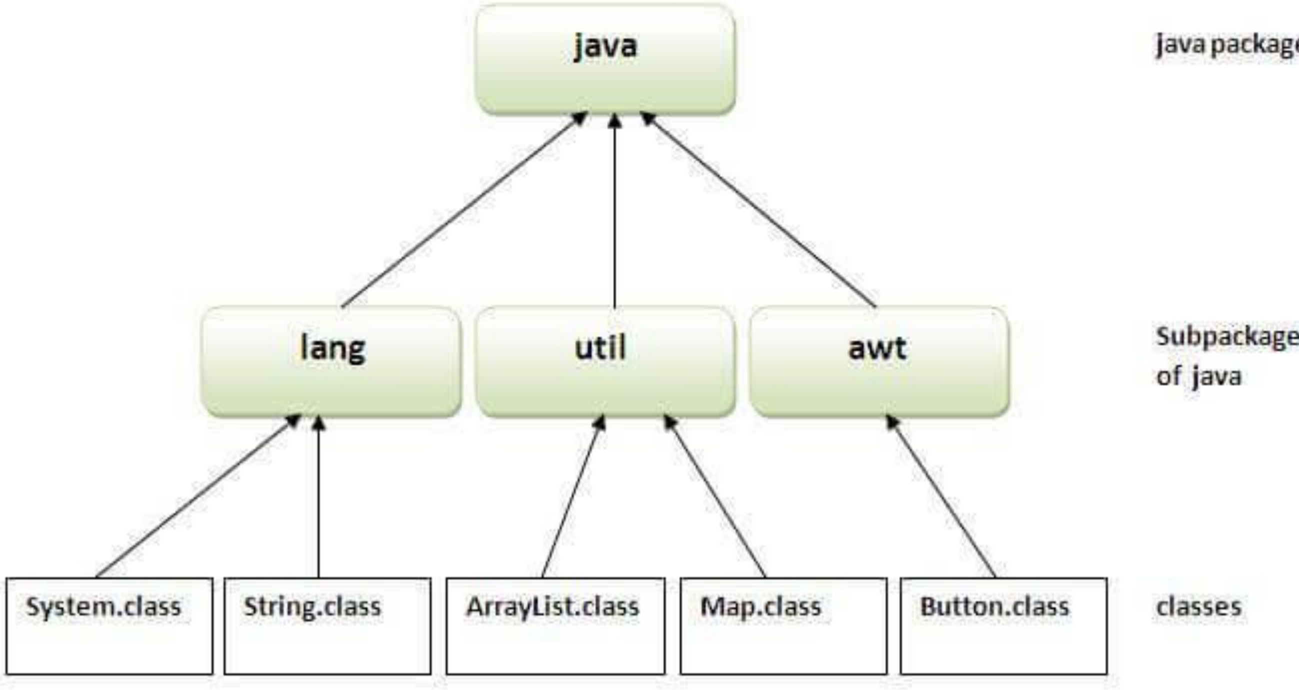
- By providing only the setter or getter method, you can make the class read-only or write-only. In other words, you can skip the getter or setter methods.
- It provides you the control over the data. Suppose you want to set the value of id which should be greater than 100 only, you can write the logic inside the setter method. You can write the logic not to store the negative numbers in the setter methods.
- It is a way to achieve data hiding in Java because other class will not be able to access the data through the private data members.
- The encapsulate class is easy to test. So, it is better for unit testing.
- The standard IDE's are providing the facility to generate the getters and setters. So, it is easy and fast to create an encapsulated class in Java.

Core Java - OOPs Concepts: Package Interview Questions

100) What is the package?

A package is a group of similar type of classes, interfaces, and sub-packages. It provides access protection and removes naming collision. The packages in Java can be categorized into two forms, inbuilt package, and user-defined package. There are many built-in packages such as Java, lang, awt, javax, swing, net, io, util, sql, etc. Consider the following example to create a package in Java.

```
//save as Simple.java
package mypack;
public class Simple{
 public static void main(String args[]){
        System.out.println("Welcome to package");
       }
     }
```



101) What are the advantages of defining packages in Java?

By defining packages, we can avoid the name conflicts between the same class names defined in different packages. Packages also enable the developer to organize the similar classes more effectively. For example, one can clearly understand that the classes present in java.io package are used to perform io related operations.

102) How to create packages in Java?

If you are using the programming IDEs like Eclipse, NetBeans, MyEclipse, etc. click on **file->new->project** and eclipse will ask you to enter the name of the package. It will create the project package containing various directories such as src, etc. If you are using an editor like notepad for java programming, use the following steps to create the package.

- Define a package **package_name**. Create the class with the name **class_name** and save this file with **your_class_name.java**.

- Now compile the file by running the following command on the terminal.

```
1.          javac -d . your_class_name.java
```

  The above command creates the package with the name **package_name** in the present working directory.

- Now, run the class file by using the absolute class file name, like following.

```
1.          java package_name.class_name
```

103) How can we access some class in another class in Java?

There are two ways to access a class in another class.

- **By using the fully qualified name:** To access a class in a different package, either we must use the fully qualified name of that class, or we must import the package containing that class.
- **By using the relative path**, We can use the path of the class that is related to the package that contains our class. It can be the same or subpackage.

104) Do I need to import java.lang package any time? Why?

No. It is by default loaded internally by the JVM.

105) Can I import same package/class twice? Will the JVM load the package twice at runtime?

One can import the same package or the same class multiple times. Neither compiler nor JVM complains about it. However, the JVM will internally load the class only once no matter how many times you import the same class.

106) What is the static import?

By static import, we can access the static members of a class directly, and there is no to qualify it with the class name.

More details.

Java: Exception Handling Interview Questions

There is given a list of exception handling interview questions with answers. If you know any exception handling interview question, kindly post it in the comment section.

107) How many types of exception can occur in a Java program?

There are mainly two types of exceptions: checked and unchecked. Here, an error is considered as the unchecked exception. According to Oracle, there are three types of exceptions:

- **Checked Exception:** Checked exceptions are the one which are checked at compile-time. For example, SQLException, ClassNotFoundException, etc.

- **Unchecked Exception:** Unchecked exceptions are the one which are handled at runtime because they can not be checked at compile-time. For example, ArithmaticException, NullPointerException, ArrayIndexOutOfBoundsException, etc.

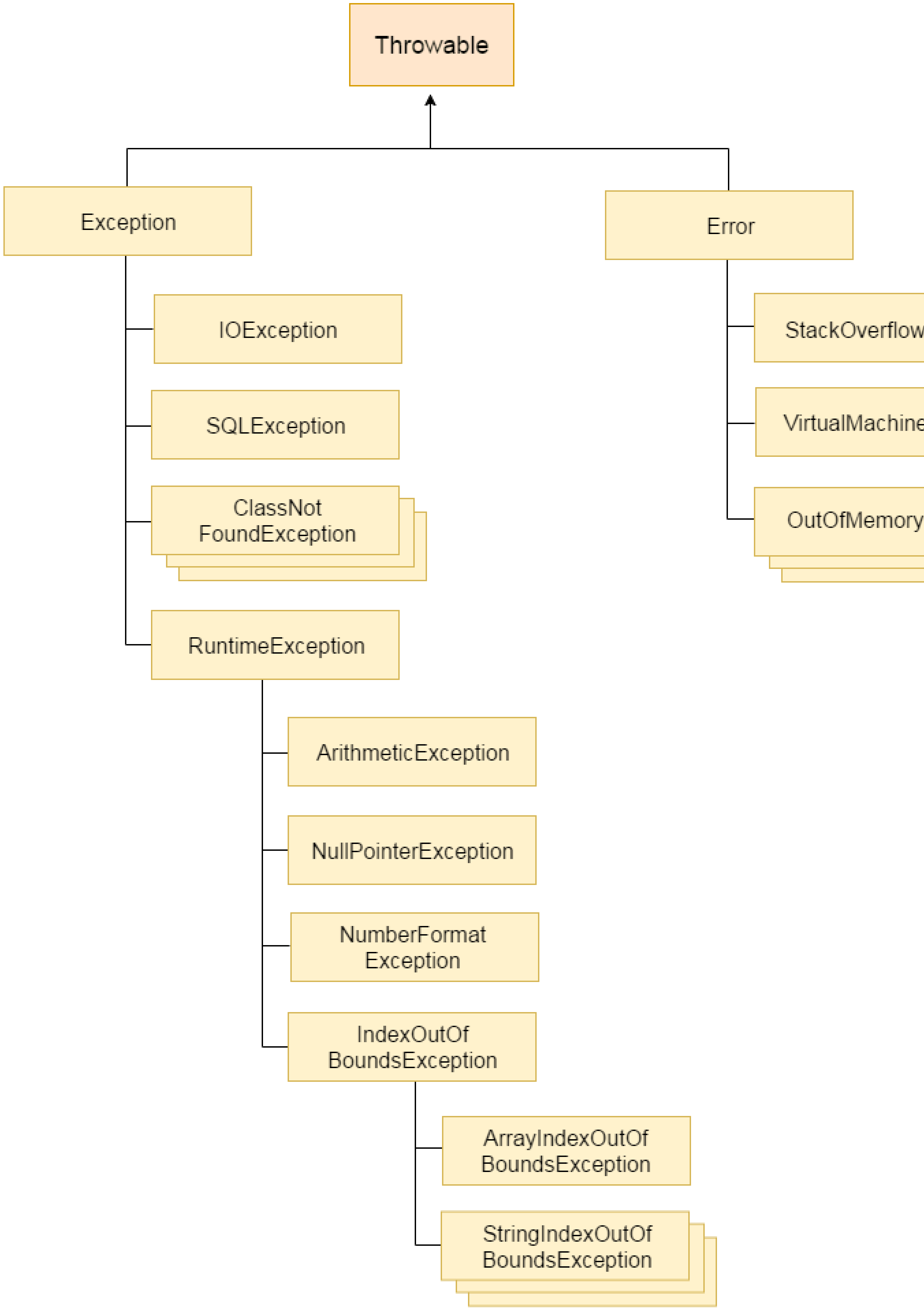
- **Error:** Error cause the program to exit since they are not recoverable. For Example, OutOfMemoryError, AssertionError, etc.

108) What is Exception Handling?

Exception Handling is a mechanism that is used to handle runtime errors. It is used primarily to handle checked exceptions. Exception handling maintains the normal flow of the program. There are mainly two types of exceptions: checked and unchecked. Here, the error is considered as the unchecked exception.

109) 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:


Throwable
Exception
Error
IOException
SQLException
ClassNot FoundException
RuntimeException
ArithmeticException
NullPointerException
NumberFormat Exception
IndexOutOf BoundsException
ArrayIndexOutOf BoundsException
StringIndexOutOf BoundsException
StackOverflow
VirtualMachine
OutOfMemory

110) What is the difference between Checked Exception and Unchecked Exception?

1) Checked Exception

The classes that extend Throwable class except RuntimeException and Error are known as checked exceptions, e.g., IOException, SQLException, etc. Checked exceptions are checked at compile-time.

2) Unchecked Exception

The classes that extend RuntimeException are known as unchecked exceptions, e.g., ArithmeticException, NullPointerException, etc. Unchecked exceptions are not checked at compile-time.

111) What is the base class for Error and Exception?

The Throwable class is the base class for Error and Exception.

112) Is it necessary that each try block must be followed by a catch block?

It is not necessary that each try block must be followed by a catch block. It should be followed by either a catch block OR a finally block. So whatever exceptions are likely to be thrown should be declared in the throws clause of the method. Consider the following example.

```
public class Main{
    public static void main(String []args){
      try{
         int a = 1;
         System.out.println(a/0);
      }
      finally
      {
```

```
                System.out.println("rest of the code...");
            }
        }
    }

```

Output:

```
Exception in thread main java.lang.ArithmeticException:/ by zero
rest of the code...
```

113) What is finally block?

The "finally" block is used to execute the important code of the program. It is executed whether an exception is handled or not. In other words, we can say that finally block is the block which is always executed. Finally block follows try or catch block. If you don't handle the exception, before terminating the program, JVM runs finally block, (if any). The finally block is mainly used to place the cleanup code such as closing a file or closing a connection. Here, we must know that for each try block there can be zero or more catch blocks, but only one finally block. The finally block will not be executed if program exits(either by calling System.exit() or by causing a fatal error that causes the process to abort).

More details.

114) Can finally block be used without a catch?

Yes, According to the definition of finally block, it must be followed by a try or catch block, therefore, we can use try block instead of catch. More details.

115) Is there any case when finally will not be executed?

Finally block will not be executed if program exits(either by calling System.exit() or by causing a fatal error that causes the process to abort).More details.

116) What is the difference between throw and throws?

throw keyword	throws keyword
1) The **throw** keyword is used to throw an exception explicitly.	The **throws** keyword is used to declare an exception.
2) The checked exceptions cannot be propagated with throw only.	The checked exception can be propagated with throws
3) The **throw** keyword is followed by an instance.	The **throws** keyword is followed by class.
4) The **throw** keyword is used within the method.	The **throws** keyword is used with the method signature.
5) You cannot throw multiple exceptions.	You can declare multiple exceptions, e.g., public void method()throws IOException, SQLException.

117) Can an exception be rethrown?

Yes.

118) Can subclass overriding method declare an exception if parent class method doesn't throw an exception?

Yes but only unchecked exception not checked.

More details.

119) What is exception propagation?

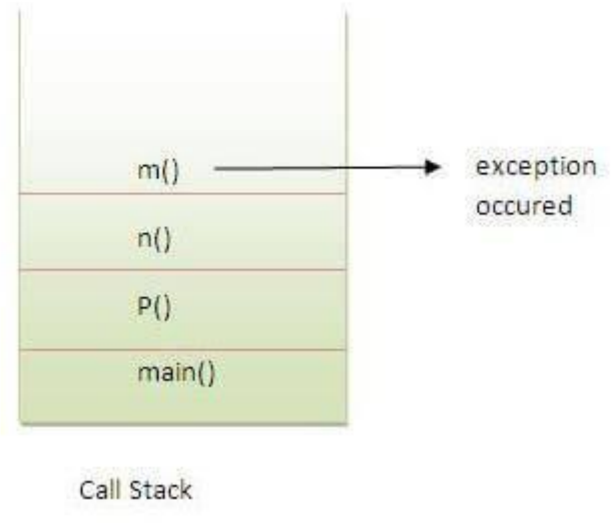
An exception is first thrown from the top of the stack and if it is not caught, it drops down the call stack to the previous method, If not caught there, the exception again drops down to the previous method, and so on until they are caught or until they reach the very bottom of the call stack. This procedure is called exception propagation. By default, checked exceptions are not propagated.

```java
class TestExceptionPropagation1{
  void m(){
    int data=50/0;
  }
  void n(){
    m();
  }
  void p(){
   try{
    n();
   }catch(Exception e){System.out.println("exception handled");}
  }
  public static void main(String args[]){
   TestExceptionPropagation1 obj=new TestExceptionPropagation1();
   obj.p();
   System.out.println("normal flow...");
  }
}
```

Test it Now

Output:

```
exception handled
normal flow...
```

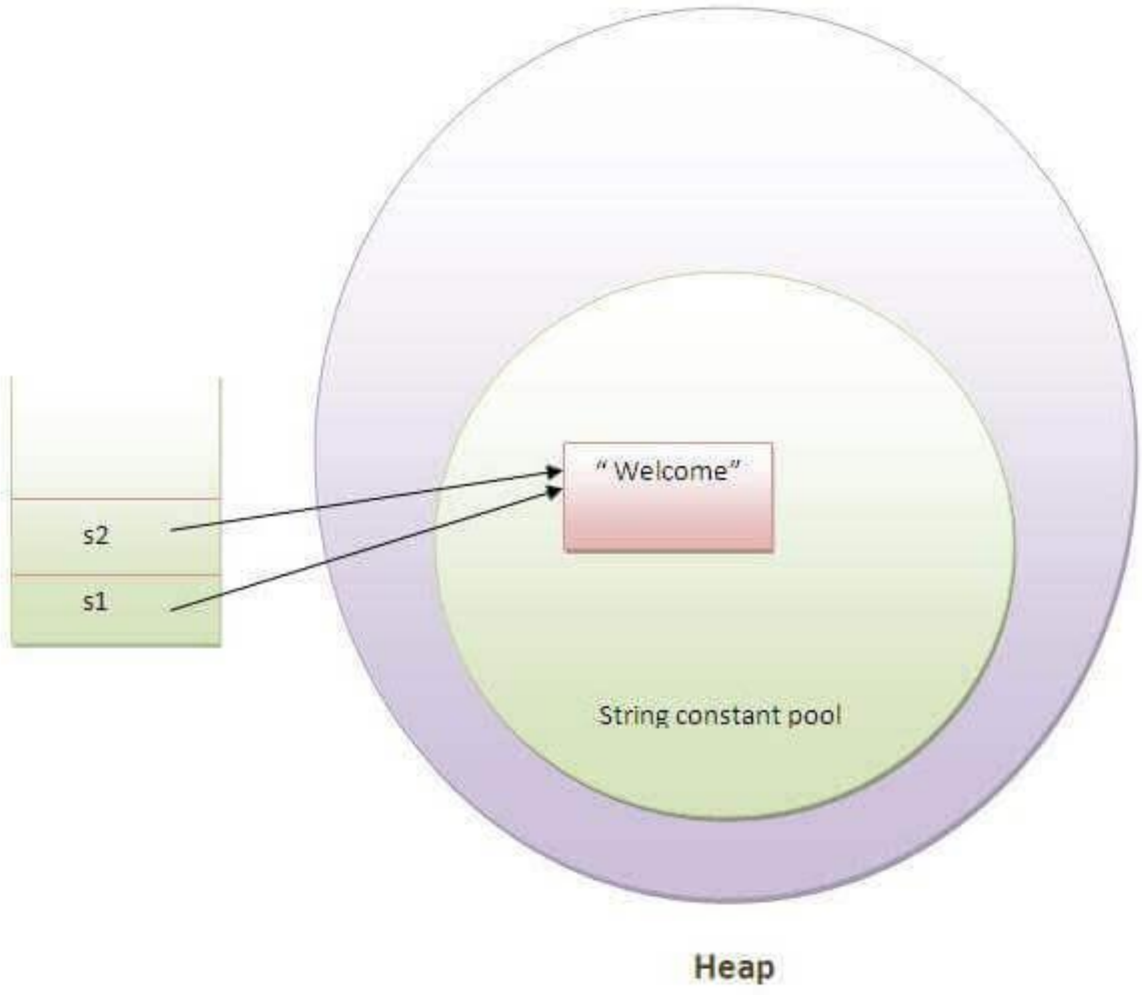


Java: String Handling Interview Questions

There is given a list of string handling interview questions with short and pointed answers. If you know any string handling interview question, kindly post it in the comment section.

149) What is String Pool?

String pool is the space reserved in the heap memory that can be used to store the strings. The main advantage of using the String pool is whenever we create a string literal; the JVM checks the "string constant pool" first. If the string already exists in the pool, a reference to the pooled instance is returned. If the string doesn't exist in the pool, a new string instance is created and placed in the pool. Therefore, it saves the memory by avoiding the duplicacy.



150) What is the meaning of immutable regarding String?

The simple meaning of immutable is unmodifiable or unchangeable. In Java, String is immutable, i.e., once string object has been created, its value can't be changed. Consider the following example for better understanding.

```
class Testimmutablestring{
 public static void main(String args[]){
   String s="Sachin";
   s.concat(" Tendulkar");//concat() method appends the string at the end
   System.out.println(s);//will print Sachin because strings are immutable objects
 }
}
```

Test it Now

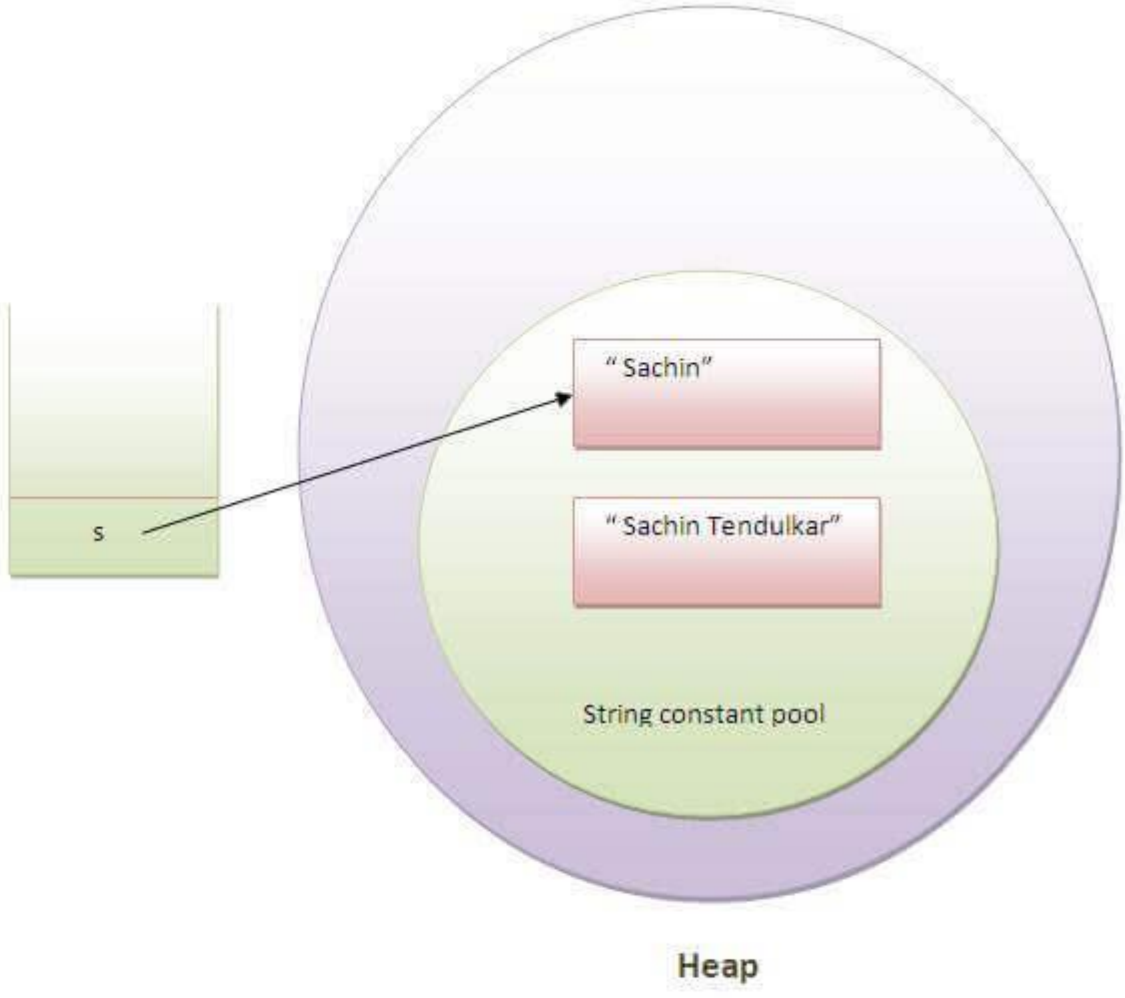
Output:

```
Sachin
```

More details.

151) Why are the objects immutable in java?

Because Java uses the concept of the string literal. Suppose there are five reference variables, all refer to one object "sachin". If one reference variable changes the value of the object, it will be affected by all the reference variables. That is why string objects are immutable in java.



152) How many ways can we create the string object?

1) String Literal

Java String literal is created by using double quotes. For Example:

```
String s="welcome";
```

Each time you create a string literal, the JVM checks the "string constant pool" first. If the string already exists in the pool, a reference to the pooled instance is returned. If the string doesn't exist in the pool, a new string instance is created and placed in the pool. String objects are stored in a special memory area known as the **string constant pool** For example:

```
String s1="Welcome";
String s2="Welcome";//It doesn't create a new instance
```

2) By new keyword

```
String s=new String("Welcome");//creates two objects and one reference variable
```

In such case, JVM will create a new string object in normal (non-pool) heap memory, and the literal "Welcome" will be placed in the constant string pool. The variable s will refer to the object in a heap (non-pool).

153) How many objects will be created in the following code?

```
String s1="Welcome";
String s2="Welcome";
String s3="Welcome";
```

Only one object will be created using the above code because strings in Java are immutable.

154) Why java uses the concept of the string literal?

To make Java more memory efficient (because no new objects are created if it exists already in the string constant pool).

155) How many objects will be created in the following code?

```java
String s = new String("Welcome");
```

Two objects, one in string constant pool and other in non-pool(heap).

---

156) What is the output of the following Java program?

```java
public class Test

  public static void main (String args[])
  {
      String a = new String("Sharma is a good player");
      String b = "Sharma is a good player";
      if(a == b)
      {
          System.out.println("a == b");
      }
      if(a.equals(b))
      {
          System.out.println("a equals b");
      }
  }
```

Output

```
a equals b
```

Explanation

The operator **==** also check whether the references of the two string objects are equal or not. Although both of the strings contain the same content, their references are not equal because both are created by different ways(Constructor and String literal) therefore, **a == b** is unequal. On the other hand, the equal() method always check for the content. Since their content is equal hence, **a equals b** is printed.

157) What is the output of the following Java program?

```
public class Test
{
    public static void main (String args[])
    {
        String s1 = "Sharma is a good player";
        String s2 = new String("Sharma is a good player");
        s2 = s2.intern();
        System.out.println(s1 ==s2);
    }
}
```

Output

```
true
```

Explanation

The intern method returns the String object reference from the string pool. In this case, s1 is created by using string literal whereas, s2 is created by using the String pool. However, s2 is changed to the reference of s1, and the operator == returns true.

158) What are the differences between String and StringBuffer?

The differences between the String and StringBuffer is given in the table below.

No.	String	StringBuffer
1)	The String class is immutable.	The StringBuffer class is mutable.

2)	The String is slow and consumes more memory when you concat too many strings because every time it creates a new instance.	The StringBuffer is fast and consumes less memory when you cancat strings.
3)	The String class overrides the equals() method of Object class. So you can compare the contents of two strings by equals() method.	The StringBuffer class doesn't override the equals() method of Object class.

159) What are the differences between StringBuffer and StringBuilder?

The differences between the StringBuffer and StringBuilder is given below.

No.	StringBuffer	StringBuilder
1)	StringBuffer is synchronized, i.e., thread safe. It means two threads can't call the methods of StringBuffer simultaneously.	StringBuilder is non-synchronized,i.e., not thread safe. It means two threads can call the methods of StringBuilder simultaneously.
2)	StringBuffer is less efficient than StringBuilder.	StringBuilder is more efficient than StringBuffer.

160) How can we create an immutable class in Java?

We can create an immutable class by defining a final class having all of its members as final. Consider the following example.

```
public final class Employee{
final String pancardNumber;

public Employee(String pancardNumber){
this.pancardNumber=pancardNumber;
}

```

```java
public String getPancardNumber(){
return pancardNumber;
}

}
```

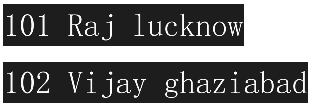
161) What is the purpose of toString() method in Java?

The toString() method returns the string representation of an object. If you print any object, java compiler internally invokes the toString() method on the object. So overriding the toString() method, returns the desired output, it can be the state of an object, etc. depending upon your implementation. By overriding the toString() method of the Object class, we can return the values of the object, so we don't need to write much code. Consider the following example.

```java
class Student{
 int rollno;
 String name;
 String city;

 Student(int rollno, String name, String city){
 this.rollno=rollno;
 this.name=name;
 this.city=city;
 }

 public String toString(){//overriding the toString() method
  return rollno+" "+name+" "+city;
 }
 public static void main(String args[]){
   Student s1=new Student(101,"Raj","lucknow");
   Student s2=new Student(102,"Vijay","ghaziabad");

```

```
    System.out.println(s1);//compiler writes here s1.toString()
    System.out.println(s2);//compiler writes here s2.toString()
  }
}
```

Output:



```
101 Raj lucknow
102 Vijay ghaziabad
```

162) Why CharArray() is preferred over String to store the password?

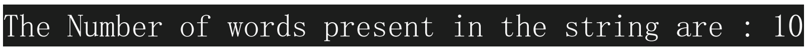
String stays in the string pool until the garbage is collected. If we store the password into a string, it stays in the memory for a longer period, and anyone having the memory-dump can extract the password as clear text. On the other hand, Using CharArray allows us to set it to blank whenever we are done with the password. It avoids the security threat with the string by enabling us to control the memory.

163) Write a Java program to count the number of words present in a string?

Program:

```
public class Test
{
  public static void main (String args[])
  {
    String s = "Sharma is a good player and he is so punctual";
    String words[] = s.split(" ");
    System.out.println("The Number of words present in the string are : "+words.length);
  }
}
```

Output

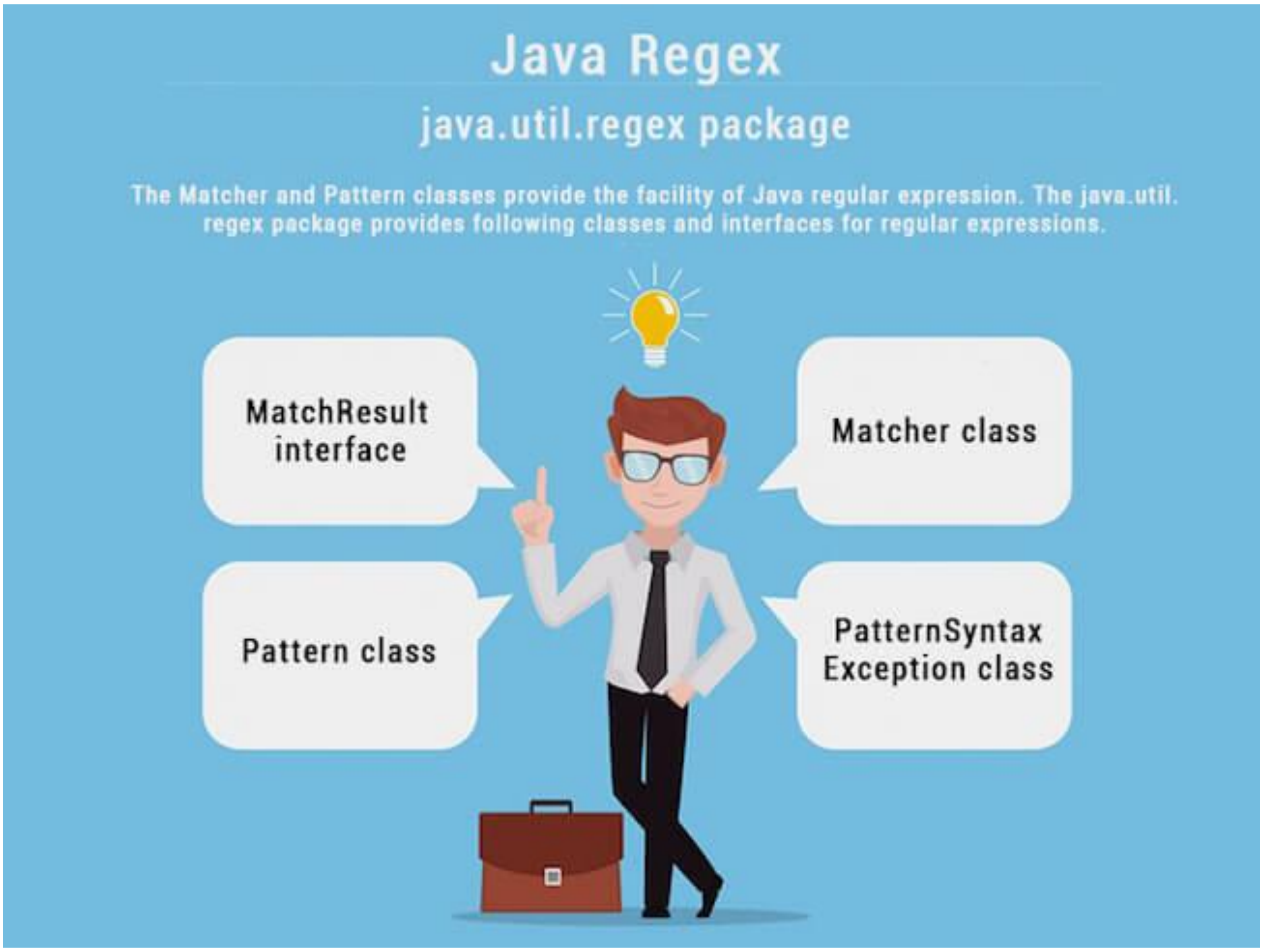


```
The Number of words present in the string are : 10
```

164) Name some classes present in java.util.regex package.

There are the following classes and interfaces present in java.util.regex package.

- MatchResult Interface
- Matcher class
- Pattern class
- PatternSyntaxException class



Java Regex

java.util.regex package

The Matcher and Pattern classes provide the facility of Java regular expression. The java.util.regex package provides following classes and interfaces for regular expressions.

165) How the metacharacters are different from the ordinary characters?

Metacharacters have the special meaning to the regular expression engine. The metacharacters are ^, $, ., *, +, etc. The regular expression engine does not consider them as the regular characters. To enable the regular expression engine treating the metacharacters as ordinary characters, we need to escape the metacharacters with the backslash.

166) Write a regular expression to validate a password. A password must start with an alphabet and followed by alphanumeric characters; Its length must be in between 8 to 20.

The regular expression for the above criteria will be: **^[a-zA-Z][a-zA-Z0-9]{8,19}** where ^ represents the start of the regex, [a-zA-Z] represents that the first character must be an alphabet, [a-zA-Z0-9] represents the alphanumeric character, {8,19} represents that the length of the password must be in between 8 and 20.

167) What is the output of the following Java program?

```java
import java.util.regex.*;
class RegexExample2{
public static void main(String args[]){
System.out.println(Pattern.matches(".s", "as")); //line 4
System.out.println(Pattern.matches(".s", "mk")); //line 5
System.out.println(Pattern.matches(".s", "mst")); //line 6
System.out.println(Pattern.matches(".s", "amms")); //line 7
System.out.println(Pattern.matches("..s", "mas")); //line 8
}}
```

Output

```
true
false
```

```
false
false
true
```

Explanation

line 4 prints true since the second character of string is s, line 5 prints false since the second character is not s, line 6 prints false since there are more than 3 characters in the string, line 7 prints false since there are more than 2 characters in the string, and it contains more than 2 characters as well, line 8 prints true since the third character of the string is s.

Core Java: Nested classes and Interfaces Interview Questions

168) What are the advantages of Java inner classes?

There are two types of advantages of Java inner classes.

- Nested classes represent a special type of relationship that is it can access all the members (data members and methods) of the outer class including private.
- Nested classes are used to develop a more readable and maintainable code because it logically groups classes and interfaces in one place only.
- **Code Optimization:** It requires less code to write.

169) What is a nested class?

The nested class can be defined as the class which is defined inside another class or interface. We use the nested class to logically group classes and interfaces in one place so that it can be more readable and maintainable. A nested class can access all the data members of the outer class including private data members and methods. The syntax of the nested class is defined below.

```
class Java_Outer_class{
 //code
 class Java_Nested_class{
  //code
 }
}

```

There are two types of nested classes, static nested class, and non-static nested class. The non-static nested class can also be called as inner-class

170) What are the disadvantages of using inner classes?

There are the following main disadvantages of using inner classes.

- Inner classes increase the total number of classes used by the developer and therefore increases the workload of JVM since it has to perform some routine operations for those extra classes which result in slower performance.
- IDEs provide less support to the inner classes as compare to the top level classes and therefore it annoys the developers while working with inner classes.

171) What are the types of inner classes (non-static nested class) used in Java?

There are mainly three types of inner classes used in Java.

Type	Description
Member Inner Class	A class created within class and outside method.
Anonymous Inner Class	A class created for implementing an interface or extending class. Its na decided by the java compiler.
Local Inner Class	A class created within the method.

172) Is there any difference between nested classes and inner classes?

Yes, inner classes are non-static nested classes. In other words, we can say that inner classes are the part of nested classes.

More details.

173) Can we access the non-final local variable, inside the local inner class?

No, the local variable must be constant if you want to access it in the local inner class.

174) How many class files are created on compiling the OuterClass in the following program?

```
public class Person {
String name, age, address;
class Employee{
  float salary=10000;
}
class BusinessMen{
  final String gstin="£4433drt3$";
}
public static void main (String args[])
{
  Person p = new Person();
}
}
```

3 class-files will be created named as Person.class, Person$BusinessMen.class, and Person$Employee.class.

175) What are anonymous inner classes?

Anonymous inner classes are the classes that are automatically declared and instantiated within an expression. We cannot apply different access modifiers to them. Anonymous class cannot be static, and cannot define any static fields, method, or class. In other words, we can say that it a class without the name and can have only one object that is created by its definition. Consider the following example.

```
abstract class Person{
  abstract void eat();
}
class TestAnonymousInner{
 public static void main(String args[]){
  Person p=new Person(){
  void eat(){System.out.println("nice fruits");}
  };
  p.eat();
 }
}
```

Test it Now

Output:

```
nice fruits
```

Consider the following example for the working of the anonymous class using interface.

```
interface Eatable{
 void eat();
}
class TestAnnonymousInner1{
 public static void main(String args[]){
 Eatable e=new Eatable(){
  public void eat(){System.out.println("nice fruits");}
 };
 e.eat();
 }
}
```

Test it Now

Output:

```
nice fruits
```

176) What is the nested interface?

An Interface that is declared inside the interface or class is known as the nested interface. It is static by default. The nested interfaces are used to group related interfaces so that they can be easy to maintain. The external interface or class must refer to the nested interface. It can't be accessed directly. The nested interface must be public if it is declared inside the interface but it can have any access modifier if declared within the class. The syntax of the nested interface is given as follows.

```
interface interface_name{
 ...
 interface nested_interface_name{
  ...
 }
}

```

More details.

177) Can a class have an interface?

Yes, an interface can be defined within the class. It is called a nested interface.

More details.

178) Can an Interface have a class?

Yes, they are static implicitly.

More details.

Garbage Collection Interview Questions

179) 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.

180) What is gc()?

The gc() method is used to invoke the garbage collector for cleanup processing. This method is found in System and Runtime classes. This function explicitly makes the Java Virtual Machine free up the space occupied by the unused objects so that it can be utilized or reused. Consider the following example for the better understanding of how the gc() method invoke the garbage collector.

```
public class TestGarbage1{
 public void finalize(){System.out.println("object is garbage collected");}

 public static void main(String args[]){
  TestGarbage1 s1=new TestGarbage1();
  TestGarbage1 s2=new TestGarbage1();
  s1=null;
  s2=null;
  System.gc();
```

```
    }
}
```

Test it Now

```
object is garbage collected
object is garbage collected
```

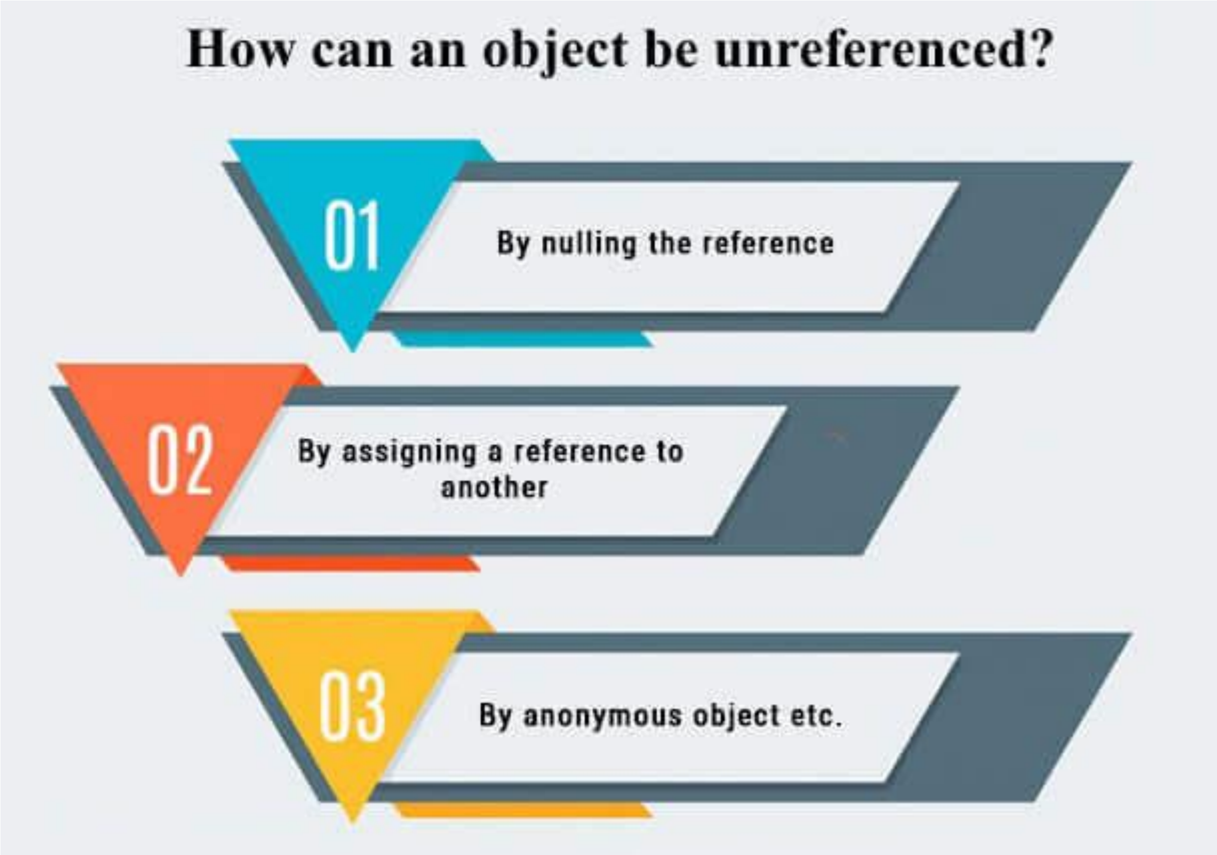
181) How is garbage collection controlled?

Garbage collection is managed by JVM. It is performed when there is not enough space in the memory and memory is running low. We can externally call the System.gc() for the garbage collection. However, it depends upon the JVM whether to perform it or not.

182) How can an object be unreferenced?

There are many ways:

- By nulling the reference
- By assigning a reference to another
- By anonymous object etc.


How can an object be unreferenced?
01
By nulling the reference
02
By assigning a reference to another
03
By anonymous object etc.

1) By nulling a reference:

```
1.          Employee e=new Employee();
2.          e=null;
```

2) By assigning a reference to another:

```
1.          Employee e1=new Employee();
2.          Employee e2=new Employee();
3.          e1=e2;//now the first object referred by e1 is available for garbage collection
```

3) By anonymous object:

```
1.          new Employee();
```

183) 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();
    }
}

```

184) Can an unreferenced object be referenced again?

Yes,

185) What kind of thread is the Garbage collector thread?

Daemon thread.

186) What is the difference between final, finally and finalize?

No.	final	finally	finalize
1)	Final is used to apply restrictions on class, method, and variable. The final class can't be inherited, final method can't be overridden, and final variable value can't be changed.	Finally is used to place important code, it will be executed whether an exception is handled or not.	Finalize is use perform clean processing just an object is ga collected.
2)	Final is a keyword.	Finally is a block.	Finalize is a metho

187) What is the purpose of the Runtime class?

Java Runtime class is used to interact with a java runtime environment. Java Runtime class provides methods to execute a process, invoke GC, get total and free memory, etc. There is only one instance of java.lang.Runtime class is available for one java application. The Runtime.getRuntime() method returns the singleton instance of Runtime class.

188) How will you invoke any external process in Java?

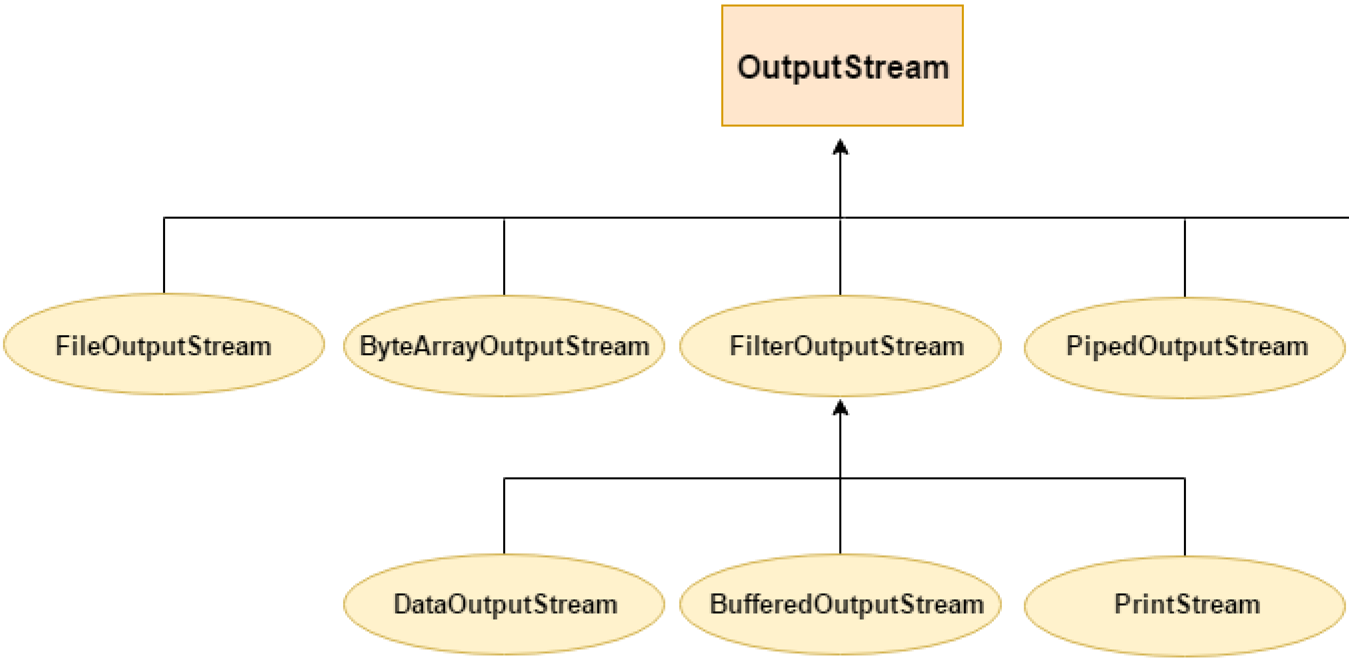
By Runtime.getRuntime().exec(?) method. Consider the following example.

```
public class Runtime1{
 public static void main(String args[])throws Exception{
  Runtime.getRuntime().exec("notepad");//will open a new notepad
 }
5.          }
```

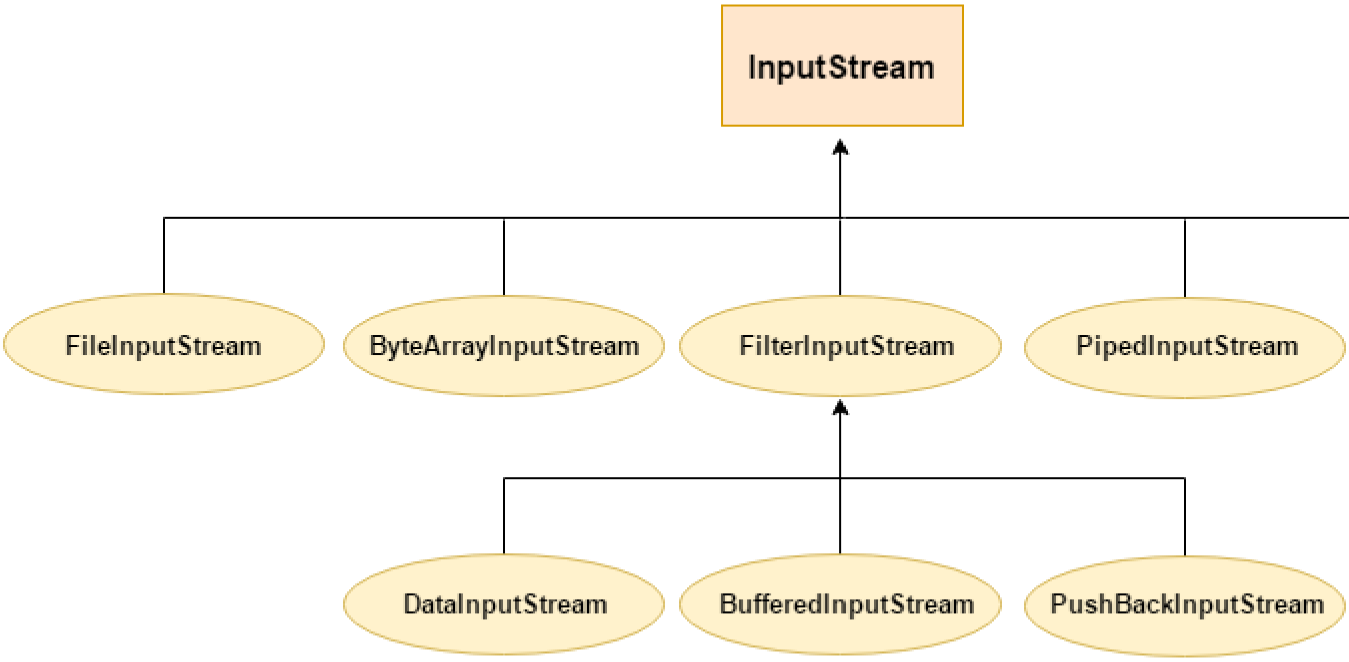
I/O Interview Questions

189) Give the hierarchy of InputStream and OutputStream classes.

OutputStream Hierarchy



InputStream Hierarchy



190) What do you understand by an IO stream?

The stream is a sequence of data that flows from source to destination. It is composed of bytes. In Java, three streams are created for us automatically.

- System.out: standard output stream
- System.in: standard input stream
- System.err: standard error stream

191) What is the difference between the Reader/Writer class hierarchy and the InputStream/OutputStream class hierarchy?

The Reader/Writer class hierarchy is character-oriented, and the InputStream/OutputStream class hierarchy is byte-oriented. The ByteStream classes are used to perform input-output of 8-bit bytes whereas the CharacterStream classes are used to perform the input/output for the 16-bit Unicode system. There are many classes in the ByteStream class hierarchy, but the most frequently used classes are FileInputStream and FileOutputStream. The most frequently used classes CharacterStream class hierarchy is FileReader and FileWriter.

192) What are the super most classes for all the streams?

All the stream classes can be divided into two types of classes that are ByteStream classes and CharacterStream Classes. The ByteStream classes are further divided into InputStream classes and OutputStream classes. CharacterStream classes are also divided into Reader classes and Writer classes. The SuperMost classes for all the InputStream classes is java.io.InputStream and for all the output stream classes is java.io.OutPutStream. Similarly, for all the reader classes, the super-most class is java.io.Reader, and for all the writer classes, it is java.io.Writer.

193) What are the FileInputStream and FileOutputStream?

Java FileOutputStream is an output stream used for writing data to a file. If you have some primitive values to write into a file, use FileOutputStream class. You can write byte-oriented as well as character-oriented data through the FileOutputStream class. However, for character-oriented data, it is preferred to use FileWriter than FileOutputStream. Consider the following example of writing a byte into a file.

```
import java.io.FileOutputStream;
public class FileOutputStreamExample {
    public static void main(String args[]){
           try{
             FileOutputStream fout=new FileOutputStream("D:\\testout.txt");
             fout.write(65);
             fout.close();
             System.out.println("success...");
            }catch(Exception e){System.out.println(e);}
      }
}
```

Java FileInputStream class obtains input bytes from a file. It is used for reading byte-oriented data (streams of raw bytes) such as image data, audio, video, etc. You can also read character-stream data. However, for reading streams of characters, it is recommended to use FileReader class. Consider the following example for reading bytes from a file.

```java
import java.io.FileInputStream;
public class DataStreamExample {
    public static void main(String args[]){
        try{
          FileInputStream fin=new FileInputStream("D:\\testout.txt");
          int i=fin.read();
          System.out.print((char)i);

          fin.close();
        }catch(Exception e){System.out.println(e);}
       }
      }

```

194) What is the purpose of using BufferedInputStream and BufferedOutputStream classes?

Java BufferedOutputStream class is used for buffering an output stream. It internally uses a buffer to store data. It adds more efficiency than to write data directly into a stream. So, it makes the performance fast. Whereas, Java BufferedInputStream class is used to read information from the stream. It internally uses the buffer mechanism to make the performance fast.

195) How to set the Permissions to a file in Java?

In Java, FilePermission class is used to alter the permissions set on a file. Java FilePermission class contains the permission related to a directory or file. All the permissions are related to the path. The path can be of two types:

- D:\\IO\\-: It indicates that the permission is associated with all subdirectories and files recursively.

- D:\\IO\\*: It indicates that the permission is associated with all directory and files within this directory excluding subdirectories.

Let's see the simple example in which permission of a directory path is granted with read permission and a file of this directory is granted for write permission.

```
package com.javatpoint;
import java.io.*;
import java.security.PermissionCollection;
public class FilePermissionExample{
     public static void main(String[] args) throws IOException {
      String srg = "D:\\IO Package\\java.txt";
      FilePermission file1 = new FilePermission("D:\\IO Package\\-", "read");
      PermissionCollection permission = file1.newPermissionCollection();
      permission.add(file1);
           FilePermission file2 = new FilePermission(srg, "write");
           permission.add(file2);
         if(permission.implies(new FilePermission(srg, "read,write"))) {
           System.out.println("Read, Write permission is granted for the path "+srg );
             }else {
            System.out.println("No Read, Write permission is granted for the path "+srg);            }
     }
}
```

Output

```
Read, Write permission is granted for the path D:\IO Package\java.txt
```

196) What are FilterStreams?

FilterStream classes are used to add additional functionalities to the other stream classes. FilterStream classes act like an interface which read the data from a stream, filters it, and pass the filtered data to the caller. The FilterStream classes provide extra functionalities like adding line numbers to the destination file, etc.

197) What is an I/O filter?

An I/O filter is an object that reads from one stream and writes to another, usually altering the data in some way as it is passed from one stream to another. Many Filter classes that allow a user to make a chain using multiple input streams. It generates a combined effect on several filters.

198) In Java, How many ways you can take input from the console?

In Java, there are three ways by using which, we can take input from the console.

- **Using BufferedReader class:** we can take input from the console by wrapping System.in into an InputStreamReader and passing it into the BufferedReader. It provides an efficient reading as the input gets buffered. Consider the following example.

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
public class Person
{
    public static void main(String[] args) throws IOException
    {
      System.out.println("Enter the name of the person");
        BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));
        String name = reader.readLine();
        System.out.println(name);
    }
}
```

- **Using Scanner class:** The Java Scanner class breaks the input into tokens using a delimiter that is whitespace by default. It provides many methods to read and parse various primitive values. Java Scanner class is widely used to parse text for string and primitive types using a regular expression. Java Scanner class extends Object class and implements Iterator and Closeable interfaces. Consider the following example.

```java
import java.util.*;
public class ScannerClassExample2 {
    public static void main(String args[]){
        String str = "Hello/This is JavaTpoint/My name is Abhishek.";
        //Create scanner with the specified String Object
        Scanner scanner = new Scanner(str);
        System.out.println("Boolean Result: "+scanner.hasNextBoolean());
        //Change the delimiter of this scanner
        scanner.useDelimiter("/");
        //Printing the tokenized Strings
        System.out.println("---Tokenizes String---");
      while(scanner.hasNext()){
          System.out.println(scanner.next());
      }
        //Display the new delimiter
        System.out.println("Delimiter used: " +scanner.delimiter());

        scanner.close();
        }
}

```

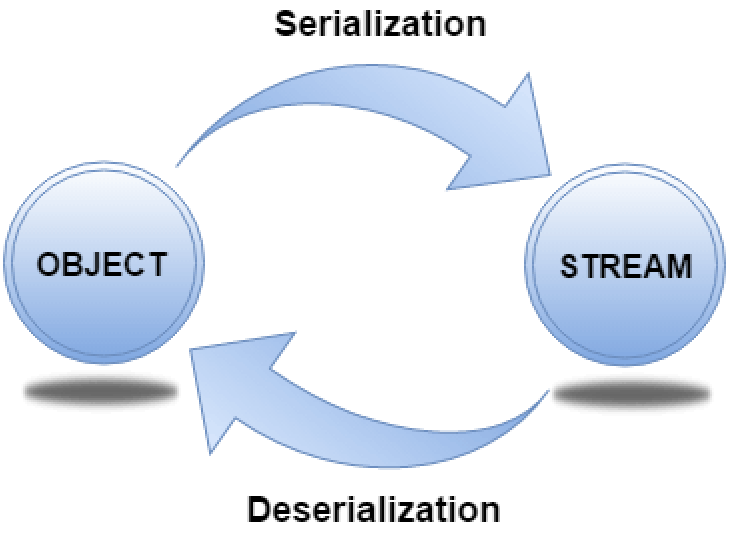
- **Using Console class:** The Java Console class is used to get input from the console. It provides methods to read texts and passwords. If you read the password using the Console class, it will not be displayed to the user. The java.io.Console class is attached to the system console internally. The Console class is introduced since 1.5. Consider the following example.

```
import java.io.Console;
class ReadStringTest{
public static void main(String args[]){
Console c=System.console();
System.out.println("Enter your name: ");
String n=c.readLine();
System.out.println("Welcome "+n);
}
}
```

Serialization Interview Questions

199) What is serialization?

Serialization in Java is a mechanism of writing the state of an object into a byte stream. It is used primarily in Hibernate, RMI, JPA, EJB and JMS technologies. It is mainly used to travel object's state on the network (which is known as marshaling). Serializable interface is used to perform serialization. It is helpful when you require to save the state of a program to storage such as the file. At a later point of time, the content of this file can be restored using deserialization. It is also required to implement RMI(Remote Method Invocation). With the help of RMI, it is possible to invoke the method of a Java object on one machine to another machine.



200) How can you make a class serializable in Java?

A class can become serializable by implementing the Serializable interface.

201) How can you avoid serialization in child class if the base class is implementing the Serializable interface?

It is very tricky to prevent serialization of child class if the base class is intended to implement the Serializable interface. However, we cannot do it directly, but the serialization can be avoided by implementing the writeObject() or readObject() methods in the subclass and throw NotSerializableException from these methods. Consider the following example.

```
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.NotSerializableException;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.io.Serializable;
```

```
class Person implements Serializable
{
    String name = " ";
    public Person(String name)
    {
        this.name = name;
    }
}
class Employee extends Person
{
    float salary;
    public Employee(String name, float salary)
    {
        super(name);
        this.salary = salary;
    }
    private void writeObject(ObjectOutputStream out) throws IOException
    {
        throw new NotSerializableException();
    }
    private void readObject(ObjectInputStream in) throws IOException

    {
        throw new NotSerializableException();
    }

}
public class Test
{
    public static void main(String[] args)
            throws Exception
    {
        Employee emp = new Employee("Sharma", 10000);

        System.out.println("name = " + emp.name);
        System.out.println("salary = " + emp.salary);

        FileOutputStream fos = new FileOutputStream("abc.ser");
        ObjectOutputStream oos = new ObjectOutputStream(fos);

        oos.writeObject(emp);

        oos.close();
        fos.close();

        System.out.println("Object has been serialized");

        FileInputStream f = new FileInputStream("ab.txt");
        ObjectInputStream o = new ObjectInputStream(f);

        Employee emp1 = (Employee)o.readObject();

        o.close();
        f.close();

        System.out.println("Object has been deserialized");

        System.out.println("name = " + emp1.name);
        System.out.println("salary = " + emp1.salary);
    }
}
```

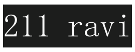
202) Can a Serialized object be transferred via network?

Yes, we can transfer a serialized object via network because the serialized object is stored in the memory in the form of bytes and can be transmitted over the network. We can also write the serialized object to the disk or the database.

203) What is Deserialization?

Deserialization is the process of reconstructing the object from the serialized state. It is the reverse operation of serialization. An ObjectInputStream deserializes objects and primitive data written using an ObjectOutputStream.

```
import java.io.*;
class Depersist{
 public static void main(String args[])throws Exception{

  ObjectInputStream in=new ObjectInputStream(new FileInputStream("f.txt"));
  Student s=(Student)in.readObject();
  System.out.println(s.id+" "+s.name);

  in.close();
 }
}
```



```
211 ravi
```

204) What is the transient keyword?

If you define any data member as transient, it will not be serialized. By determining transient keyword, the value of variable need not persist when it is restored. More details.

205) What is Externalizable?

The Externalizable interface is used to write the state of an object into a byte stream in a compressed format. It is not a marker interface.

206) What is the difference between Serializable and Externalizable interface?

No.	Serializable	Externalizable
1)	The Serializable interface does not have any method, i.e., it is a marker interface.	The Externalizable interface contains is not a marker interface, It contains two methods, i.e., writeExternal() and readExternal().
2)	It is used to "mark" Java classes so that objects of these classes may get the certain capability.	The Externalizable interface provides control of the serialization logic to the programmer.
3)	It is easy to implement but has the higher performance cost.	It is used to perform the serialization and often result in better performance.
4)	No class constructor is called in serialization.	We must call a public default constructor while using this interface.

213) What is the purpose of using java.lang.Class class?

The java.lang.Class class performs mainly two tasks:

- Provides methods to get the metadata of a class at runtime.

- Provides methods to examine and change the runtime behavior of a class.

217) Can you access the private method from outside the class?

Yes, by changing the runtime behavior of a class if the class is not secured.

Miscellaneous Interview Questions

218)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

```
public class Test1
{
 public static void main(String[] args) {
 Integer i = new Integer(201);
 Integer j = new Integer(201);
 if(i == j)
 {
  System.out.println("hello");
 }
 else
 {
  System.out.println("bye");
 }
 }
}
```

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.

221) What is object cloning?

The object cloning is a way to create an exact copy of an object. The clone() method of the Object class is used to clone an object. The java.lang.Cloneable interface must be implemented by the class whose object clone we want to create. If we don't implement Cloneable interface, clone() method generates CloneNotSupportedException. The clone() method is defined in the Object class. The syntax of the clone() method is as follows:

protected Object clone() throws CloneNotSupportedException

222) What are the advantages and disadvantages of object cloning?

Advantage of Object Cloning

- You don't need to write lengthy and repetitive codes. Just use an abstract class with a 4- or 5-line long clone() method.
- It is the easiest and most efficient way of copying objects, especially if we are applying it to an already developed or an old project. Just define a parent class, implement Cloneable in it, provide the definition of the clone() method and the task will be done.
- Clone() is the fastest way to copy the array.

Disadvantage of Object Cloning

- To use the Object.clone() method, we have to change many syntaxes to our code, like implementing a Cloneable interface, defining the clone() method and handling CloneNotSupportedException, and finally, calling Object.clone(), etc.

- We have to implement the Cloneable interface while it does not have any methods in it. We have to use it to tell the JVM that we can perform a clone() on our object.
- Object.clone() is protected, so we have to provide our own clone() and indirectly call Object.clone() from it.
- Object.clone() does not invoke any constructor, so we do not have any control over object construction.
- If you want to write a clone method in a child class, then all of its superclasses should define the clone() method in them or inherit it from another parent class. Otherwise, the super.clone() chain will fail.
- Object.clone() supports only shallow copying, but we will need to override it if we need deep cloning.

223) What is a native method?

A native method is a method that is implemented in a language other than Java. Natives methods are sometimes also referred to as foreign methods.

224) What is the purpose of the strictfp keyword?

Java strictfp keyword ensures that you will get the same result on every platform if you perform operations in the floating-point variable. The precision may differ from platform to platform that is why java programming language has provided the strictfp keyword so that you get the same result on every platform. So, now you have better control over the floating-point arithmetic.

225) What is the purpose of the System class?

The purpose of the System class is to provide access to system resources such as standard input and output. It cannot be instantiated. Facilities provided by System class are given below.

- Standard input
- Error output streams
- Standard output
- utility method to copy the portion of an array

- utilities to load files and libraries

There are the three fields of Java System class, i.e., static printstream err, static inputstream in, and standard output stream.

226) What comes to mind when someone mentions a shallow copy in Java?

Object cloning.

227) What is a singleton class?

Singleton class is the class which can not be instantiated more than once. To make a class singleton, we either make its constructor private or use the static getInstance method. Consider the following example.

```
class Singleton{
    private static Singleton single_instance = null;
    int i;
     private Singleton ()
     {
         i=90;
     }
     public static Singleton getInstance()
     {
         if(single_instance == null)
         {
             single_instance = new Singleton();
         }
         return single_instance;
     }
}
public class Main
```

```
{
    public static void main (String args[])
    {
        Singleton first = Singleton.getInstance();
        System.out.println("First instance integer value:"+first.i);
        first.i=first.i+90;
        Singleton second = Singleton.getInstance();
        System.out.println("Second instance integer value:"+second.i);
    }
}

```

228) Write a Java program that prints all the values given at command-line.

Program

```
class A{
public static void main(String args[]){
for(int i=0;i<args.length;i++)
System.out.println(args[i]);
}
}
```

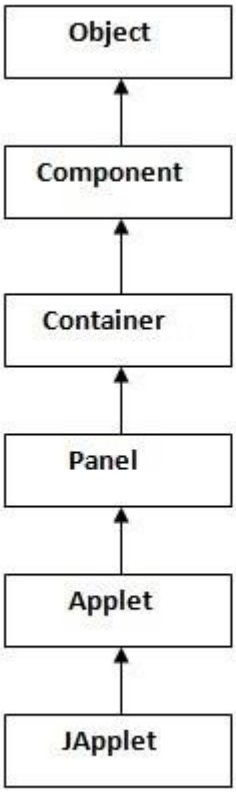
```
compile by > javac A.java
run by > java A sonoo jaiswal 1 3 abc
```

Output

```
sonoo
jaiswal
1
3
abc
```

235) What is an applet?

An applet is a small java program that runs inside the browser and generates dynamic content. It is embedded in the webpage and runs on the client side. It is secured and takes less response time. It can be executed by browsers running under many platforms, including Linux, Windows, Mac Os, etc. However, the plugins are required at the client browser to execute the applet. The following image shows the architecture of Applet.



When an applet is created, the following methods are invoked in order.

- init()
- start()
- paint()

When an applet is destroyed, the following functions are invoked in order.

- stop()
- destroy()

236) Can you write a Java class that could be used both as an applet as well as an application?

Yes. Add a main() method to the applet.

Internationalization Interview Questions

Java Bean Interview Questions

239) What is a JavaBean?

JavaBean is a reusable software component written in the Java programming language, designed to be manipulated visually by a software development environment, like JBuilder or VisualAge for Java. t. A JavaBean encapsulates many objects into one object so that we can access this object from multiple places. Moreover, it provides the easy maintenance. Consider the following example to create a JavaBean class.

```
//Employee.java
package mypack;
public class Employee implements java.io.Serializable{
private int id;
private String name;
public Employee(){}
public void setId(int id){this.id=id;}
public int getId(){return id;}
public void setName(String name){this.name=name;}
public String getName(){return name;}
}
```

240) What is the purpose of using the Java bean?

According to Java white paper, it is a reusable software component. A bean encapsulates many objects into one object so that we can access this object from multiple places. Moreover, it provides the easy maintenance.

241) What do you understand by the bean persistent property?

The persistence property of Java bean comes into the act when the properties, fields, and state information are saved to or retrieve from the storage.

Core Java: Data Structure interview questions

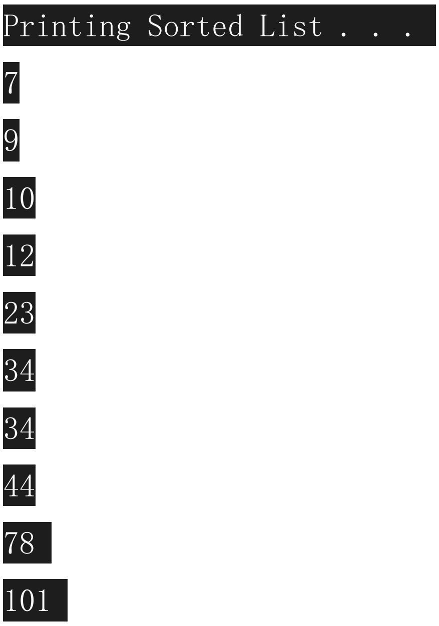
248) How to perform Bubble Sort in Java?

Consider the following program to perform Bubble sort in Java.

```
public class BubbleSort {
 public static void main(String[] args) {
 int[] a = {10, 9, 7, 101, 23, 44, 12, 78, 34, 23};
 for(int i=0;i<10;i++)
 {
   for (int j=0;j<10;j++)
   {
     if(a[i]<a[j])
     {
       int temp = a[i];
       a[i]=a[j];
       a[j] = temp;
     }
   }
 }
 System.out.println("Printing Sorted List ...");
```

```
  for(int i=0;i<10;i++)
  {
   System.out.println(a[i]);
  }
 }
 }
```

Output:



```
Printing Sorted List . . .
7
9
10
12
23
34
34
44
78
101
```

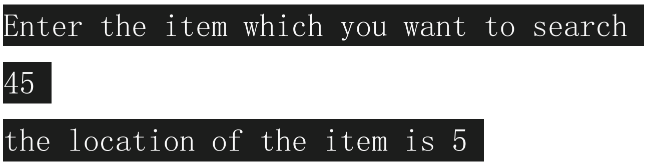
249) How to perform Binary Search in Java?

Consider the following program to perform the binary search in Java.

```
import java.util.*;
public class BinarySearch {
public static void main(String[] args) {
  int[] arr = {16, 19, 20, 23, 45, 56, 78, 90, 96, 100};
  int item, location = -1;
  System.out.println("Enter the item which you want to search");
  Scanner sc = new Scanner(System.in);
  item = sc.nextInt();
  location = binarySearch(arr,0,9,item);
```

```
 if(location != -1)
 System.out.println("the location of the item is "+location);
 else
  System.out.println("Item not found");
 }
public static int binarySearch(int[] a, int beg, int end, int item)
{
 int mid;
 if(end >= beg)
 {
  mid = (beg + end)/2;
  if(a[mid] == item)
  {
   return mid+1;
  }
  else if(a[mid] < item)
  {
   return binarySearch(a,mid+1,end,item);
  }
  else
  {
   return binarySearch(a,beg,mid-1,item);
  }
 }
 return -1;
}
}
```

Output:



```
Enter the item which you want to search
45
the location of the item is 5
```

250) How to perform Selection Sort in Java?

Consider the following program to perform selection sort in Java.

```
public class SelectionSort {
public static void main(String[] args) {
  int[] a = {10, 9, 7, 101, 23, 44, 12, 78, 34, 23};
  int i,j,k,pos,temp;
  for(i=0;i<10;i++)
  {
    pos = smallest(a,10,i);
    temp = a[i];
    a[i]=a[pos];
    a[pos] = temp;
  }
  System.out.println("\nprinting sorted elements...\n");
  for(i=0;i<10;i++)
  {
    System.out.println(a[i]);
  }
}
public static int smallest(int a[], int n, int i)
{
  int small,pos,j;
  small = a[i];
  pos = i;
  for(j=i+1;j<10;j++)
  {
    if(a[j]<small)
    {
      small = a[j];
      pos=j;
```

```
            }
          }
          return pos;
        }
        }
```

Output:

```
printing sorted elements...
7
9
10
12
23
23
34
44
78
101
```

251) How to perform Linear Search in Java?

Consider the following program to perform Linear search in Java.

```
import java.util.Scanner;

public class Leniear_Search {
public static void main(String[] args) {
  int[] arr = {10, 23, 15, 8, 4, 3, 25, 30, 34, 2, 19};
  int item,flag=0;
  Scanner sc = new Scanner(System.in);
  System.out.println("Enter Item ?");
  item = sc.nextInt();
  for(int i = 0; i<10; i++)
```

```java
    {
        if(arr[i]==item)
        {
            flag = i+1;
            break;
        }
        else
            flag = 0;
    }
    if(flag != 0)
    {
        System.out.println("Item found at location" + flag);
    }
    else
        System.out.println("Item not found");

}
}
```

Output:

```
Enter Item ?
23
Item found at location 2
Enter Item ?
22
Item not found
```

252) How to perform merge sort in Java?

Consider the following program to perform merge sort in Java.

```java
public class MyMergeSort
{
```

```
void merge(int arr[], int beg, int mid, int end)
{

int l = mid - beg + 1;
int r = end - mid;

intLeftArray[] = new int [l];
intRightArray[] = new int [r];

for (int i=0; i<l; ++i)
LeftArray[i] = arr[beg + i];

for (int j=0; j<r; ++j)
RightArray[j] = arr[mid + 1+ j];


int i = 0, j = 0;
int k = beg;
while (i<l&&j<r)
{
if (LeftArray[i] <= RightArray[j])
{
arr[k] = LeftArray[i];
i++;
}
else
{
arr[k] = RightArray[j];
j++;
}
k++;
}
```

```
while (i<l)
{
arr[k] = LeftArray[i];
i++;
k++;
}

while (j<r)
{
arr[k] = RightArray[j];
j++;
k++;
}
}

void sort(int arr[], int beg, int end)
{
if (beg<end)
{
int mid = (beg+end)/2;
sort(arr, beg, mid);
sort(arr , mid+1, end);
merge(arr, beg, mid, end);
}
}
public static void main(String args[])
{
intarr[] = {90,23,101,45,65,23,67,89,34,23};
MyMergeSort ob = new MyMergeSort();
ob.sort(arr, 0, arr.length-1);

System.out.println("\nSorted array");
```

```
for(int i =0; i<arr.length;i++)
{
  System.out.println(arr[i]+"");
}
}
}
```

Output:

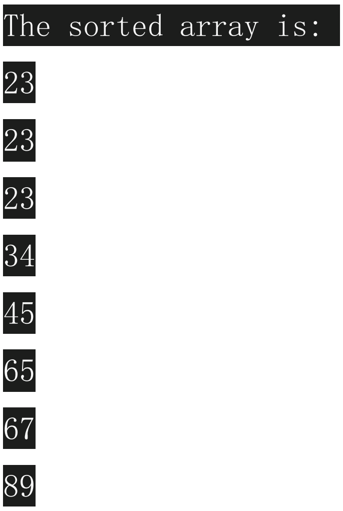
```
Sorted array
23
23
23
34
45
65
67
89
90
101
```

253) How to perform quicksort in Java?

Consider the following program to perform quicksort in Java.

```
public class QuickSort {
public static void main(String[] args) {
    int i;
    int[] arr={90,23,101,45,65,23,67,89,34,23};
    quickSort(arr, 0, 9);
    System.out.println("\n The sorted array is: \n");
    for(i=0;i<10;i++)
    System.out.println(arr[i]);
  }
  public static int partition(int a[], int beg, int end)
  {

    int left, right, temp, loc, flag;
    loc = left = beg;
    right = end;
    flag = 0;
    while(flag != 1)
    {
      while((a[loc] <= a[right]) && (loc!=right))
      right--;
      if(loc==right)
      flag =1;
      elseif(a[loc]>a[right])
      {
        temp = a[loc];
        a[loc] = a[right];
        a[right] = temp;
        loc = right;
      }
      if(flag!=1)
      {
        while((a[loc] >= a[left]) && (loc!=left))
        left++;
        if(loc==left)
        flag =1;
        elseif(a[loc] <a[left])
        {
          temp = a[loc];
          a[loc] = a[left];
          a[left] = temp;
          loc = left;
        }
      }
    }
    returnloc;
  }
  static void quickSort(int a[], int beg, int end)
  {

    int loc;
    if(beg<end)
    {
      loc = partition(a, beg, end);
      quickSort(a, beg, loc-1);
      quickSort(a, loc+1, end);
    }
  }
}
```

Output:



```
The sorted array is:
23
23
23
34
45
65
67
89
```

90

101

254) Write a program in Java to create a doubly linked list containing n nodes.

Consider the following program to create a doubly linked list containing n nodes.

```java
public class CountList {

    //Represent a node of the doubly linked list

    class Node{
        int data;
        Node previous;
        Node next;

        public Node(int data) {
            this.data = data;
        }
    }

    //Represent the head and tail of the doubly linked list
    Node head, tail = null;

    //addNode() will add a node to the list
    public void addNode(int data) {
        //Create a new node
        Node newNode = new Node(data);

        //If list is empty
        if(head == null) {
            //Both head and tail will point to newNode
            head = tail = newNode;
            //head's previous will point to null
            head.previous = null;
            //tail's next will point to null, as it is the last node of the list
            tail.next = null;
        }
        else {
            //newNode will be added after tail such that tail's next will point to newNode
            tail.next = newNode;
            //newNode's previous will point to tail
            newNode.previous = tail;
            //newNode will become new tail
            tail = newNode;
            //As it is last node, tail's next will point to null
            tail.next = null;
        }
    }

    //countNodes() will count the nodes present in the list
    public int countNodes() {
        int counter = 0;
        //Node current will point to head
        Node current = head;

        while(current != null) {
            //Increment the counter by 1 for each node
            counter++;
            current = current.next;
        }
        return counter;
```

```java
    }

    //display() will print out the elements of the list
    public void display() {
        //Node current will point to head
        Node current = head;
        if(head == null) {
            System.out.println("List is empty");
            return;
        }
        System.out.println("Nodes of doubly linked list: ");
        while(current != null) {
            //Prints each node by incrementing the pointer.

            System.out.print(current.data + " ");
            current = current.next;
        }
    }

    public static void main(String[] args) {

        CountList dList = new CountList();
        //Add nodes to the list
        dList.addNode(1);
        dList.addNode(2);
        dList.addNode(3);
        dList.addNode(4);
        dList.addNode(5);

        //Displays the nodes present in the list
        dList.display();

```

```
            //Counts the nodes present in the given list
            System.out.println("\nCount of nodes present in the list: " + dList.countNodes());
        }
    }
```

Output:

```
Nodes of doubly linked list:
1 2 3 4 5
Count of nodes present in the list: 5
```

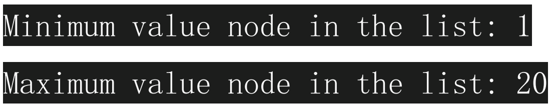
255) Write a program in Java to find the maximum and minimum value node from a circular linked list.

Consider the following program.

```
public class MinMax {
    //Represents the node of list.
    public class Node{
        int data;
        Node next;
        public Node(int data) {
            this.data = data;
        }
    }

    //Declaring head and tail pointer as null.
    public Node head = null;
    public Node tail = null;

    //This function will add the new node at the end of the list.
    public void add(int data){
        //Create new node
        Node newNode = new Node(data);
        //Checks if the list is empty.
        if(head == null) {
             //If list is empty, both head and tail would point to new node.
            head = newNode;
            tail = newNode;
            newNode.next = head;
        }
        else {
            //tail will point to new node.
            tail.next = newNode;
            //New node will become new tail.
            tail = newNode;
            //Since, it is circular linked list tail will points to head.
            tail.next = head;
        }
    }

    //Finds out the minimum value node in the list
    public void minNode() {
        Node current = head;
        //Initializing min to initial node data
        int min = head.data;
        if(head == null) {
            System.out.println("List is empty");
        }
        else {
             do{
                //If current node's data is smaller than min
                //Then replace value of min with current node's data
                if(min > current.data) {
                    min = current.data;
                }
                current= current.next;
            }while(current != head);

            System.out.println("Minimum value node in the list: "+ min);
        }
    }

    //Finds out the maximum value node in the list
    public void maxNode() {
        Node current = head;
        //Initializing max to initial node data
        int max = head.data;
        if(head == null) {
            System.out.println("List is empty");
        }
        else {
             do{
                //If current node's data is greater than max
                //Then replace value of max with current node's data
                if(max < current.data) {
                    max = current.data;
                }
                current= current.next;
               }while(current != head);

            System.out.println("Maximum value node in the list: "+ max);
        }
    }

    public static void main(String[] args) {
        MinMax cl = new MinMax();
```

```
        //Adds data to the list
        cl.add(5);
        cl.add(20);
        cl.add(10);
        cl.add(1);
        //Prints the minimum value node in the list
        cl.minNode();
        //Prints the maximum value node in the list
        cl.maxNode();
    }
}
```

Output:



```
Minimum value node in the list: 1
Maximum value node in the list: 20
```

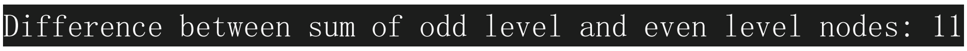
256) Write a program in Java to calculate the difference between the sum of the odd level and even level nodes of a Binary Tree.

Consider the following program.

```
import java.util.LinkedList;
import java.util.Queue;

public class DiffOddEven {

    //Represent a node of binary tree
    public static class Node{
        int data;
        Node left;
        Node right;

        public Node(int data){
            //Assign data to the new node, set left and right children to null

            this.data = data;
            this.left = null;
            this.right = null;
            }
        }

    //Represent the root of binary tree
    public Node root;

    public DiffOddEven(){
        root = null;
    }

    //difference() will calculate the difference between sum of odd and even levels of binary tree
    public int difference() {
          int oddLevel = 0, evenLevel = 0, diffOddEven = 0;

          //Variable nodesInLevel keep tracks of number of nodes in each level
          int nodesInLevel = 0;

          //Variable currentLevel keep track of level in binary tree
          int currentLevel = 0;

          //Queue will be used to keep track of nodes of tree level-wise
          Queue<Node> queue = new LinkedList<Node>();

        //Check if root is null
        if(root == null) {
            System.out.println("Tree is empty");
            return 0;
        }
        else {
            //Add root node to queue as it represents the first level
            queue.add(root);
            currentLevel++;

            while(queue.size() != 0) {

                //Variable nodesInLevel will hold the size of queue i.e. number of elements in queue
                nodesInLevel = queue.size();

                while(nodesInLevel > 0) {
                    Node current = queue.remove();

                  //Checks if currentLevel is even or not.
                    if(currentLevel % 2 == 0)
                        //If level is even, add nodes's to variable evenLevel
                        evenLevel += current.data;
                    else
                        //If level is odd, add nodes's to variable oddLevel
                        oddLevel += current.data;

                    //Adds left child to queue
                    if(current.left != null)
                        queue.add(current.left);
                    //Adds right child to queue
                    if(current.right != null)
```

```
                        queue.add(current.right);
                    nodesInLevel--;
                }
                currentLevel++;
            }
            //Calculates difference between oddLevel and evenLevel
            diffOddEven = Math.abs(oddLevel - evenLevel);
        }
        return diffOddEven;
    }

    public static void main (String[] args) {

        DiffOddEven bt = new DiffOddEven();
        //Add nodes to the binary tree
        bt.root = new Node(1);
        bt.root.left = new Node(2);
        bt.root.right = new Node(3);
        bt.root.left.left = new Node(4);
        bt.root.right.left = new Node(5);
        bt.root.right.right = new Node(6);

        //Display the difference between sum of odd level and even level nodes
        System.out.println("Difference between sum of odd level and even level nodes: " + bt.difference());
    }
}
```

Output:



```
Difference between sum of odd level and even level nodes: 11
```

Multithreading Interview Questions

1) What is multithreading?

Multithreading is a process of executing multiple threads simultaneously. Multithreading is used to obtain the multitasking. It consumes less memory and gives the fast and efficient performance. Its main advantages are:

- Threads share the same address space.
- The thread is lightweight.
- The cost of communication between the processes is low.

2) What is the thread?

A thread is a lightweight subprocess. It is a separate path of execution because each thread runs in a different stack frame. A process may contain multiple threads. Threads share the process resources, but still, they execute independently.

5) What is the purpose of wait() method in Java?

The wait() method is provided by the Object class in Java. This method is used for inter-thread communication in Java. The java.lang.Object.wait() is used to pause the current thread, and wait until another thread does not call the notify() or notifyAll() method. Its syntax is given below.

public final void wait()

6) Why must wait() method be called from the synchronized block?

We must call the wait method otherwise it will throw **java.lang.IllegalMonitorStateException** exception. Moreover, we need wait() method for inter-thread communication with notify() and notifyAll(). Therefore It must be present in the synchronized block for the proper and correct communication.

7) What are the advantages of multithreading?

Multithreading programming has the following advantages:

- Multithreading allows an application/program to be always reactive for input, even already running with some background tasks
- Multithreading allows the faster execution of tasks, as threads execute independently.
- Multithreading provides better utilization of cache memory as threads share the common memory resources.
- Multithreading reduces the number of the required server as one server can execute multiple threads at a time.

8) What are the states in the lifecycle of a Thread?

A thread can have one of the following states during its lifetime:

1. **New:** In this state, a Thread class object is created using a new operator, but the thread is not alive. Thread doesn't start until we call the start() method.
2. **Runnable:** In this state, the thread is ready to run after calling the start() method. However, the thread is not yet selected by the thread scheduler.
3. **Running:** In this state, the thread scheduler picks the thread from the ready state, and the thread is running.
4. **Waiting/Blocked:** In this state, a thread is not running but still alive, or it is waiting for the other thread to finish.
5. **Dead/Terminated:** A thread is in terminated or dead state when the run() method exits.

14) What is the difference between wait() and sleep() method?

wait()	sleep()
1) The wait() method is defined in Object class.	The sleep() method is defined in Thread class.
2) The wait() method releases the lock.	The sleep() method doesn't release the lock.

15) Is it possible to start a thread twice?

No, we cannot restart the thread, as once a thread started and executed, it goes to the Dead state. Therefore, if we try to start a thread twice, it will give a runtimeException "java.lang.IllegalThreadStateException". Consider the following example.

```java
public class Multithread1 extends Thread
{
   public void run()
    {
      try {
          System.out.println("thread is executing now........");
      } catch(Exception e) {
      }
    }
    public static void main (String[] args) {
        Multithread1 m1= new Multithread1();
        m1.start();
        m1.start();
    }
}
```

Output

```
thread is executing now........
Exception in thread "main" java.lang.IllegalThreadStateException
	at java.lang.Thread.start(Thread.java:708)
	at Multithread1.main(Multithread1.java:13)
```

16) Can we call the run() method instead of start()?

Yes, calling run() method directly is valid, but it will not work as a thread instead it will work as a normal object. There will not be context-switching between the threads. When we call the start() method, it internally calls the run() method, which creates a new stack for a thread while directly calling the run() will not create a new stack.

21) What is the synchronization?

Synchronization is the capability to control the access of multiple threads to any shared resource. It is used:

1. To prevent thread interference.
2. To prevent consistency problem.

When the multiple threads try to do the same task, there is a possibility of an erroneous result, hence to remove this issue, Java uses the process of synchronization which allows only one thread to be executed at a time. Synchronization can be achieved in three ways:

- by the synchronized method
- by synchronized block
- by static synchronization

Syntax for synchronized block

```
synchronized(object reference expression)
```

```
{
    //code block
}

```

22) What is the purpose of the Synchronized block?

The Synchronized block can be used to perform synchronization on any specific resource of the method. Only one thread at a time can execute on a particular resource, and all other threads which attempt to enter the synchronized block are blocked.

- Synchronized block is used to lock an object for any shared resource.
- The scope of the synchronized block is limited to the block on which, it is applied. Its scope is smaller than a method.

23)Can Java object be locked down for exclusive use by a given thread?

Yes. You can lock an object by putting it in a "synchronized" block. The locked object is inaccessible to any thread other than the one that explicitly claimed it.

24) What is static synchronization?

If you make any static method as synchronized, the lock will be on the class not on the object. If we use the synchronized keyword before a method so it will lock the object (one thread can access an object at a time) but if we use static synchronized so it will lock a class (one thread can access a class at a time).

25)What is the difference between notify() and notifyAll()?

The notify() is used to unblock one waiting thread whereas notifyAll() method is used to unblock all the threads in waiting state.

26)What is the deadlock?

Deadlock is a situation in which every thread is waiting for a resource which is held by some other waiting thread. In this situation, Neither of the thread executes nor it gets the chance to be executed. Instead, there exists a universal waiting state among all the threads. Deadlock is a very complicated situation which can break our code at runtime.

27) How to detect a deadlock condition? How can it be avoided?

We can detect the deadlock condition by running the code on cmd and collecting the Thread Dump, and if any deadlock is present in the code, then a message will appear on cmd.

Ways to avoid the deadlock condition in Java:

- **Avoid Nested lock:** Nested lock is the common reason for deadlock as deadlock occurs when we provide locks to various threads so we should give one lock to only one thread at some particular time.
- **Avoid unnecessary locks:** we must avoid the locks which are not required.
- **Using thread join:** Thread join helps to wait for a thread until another thread doesn't finish its execution so we can avoid deadlock by maximum use of join method.

28) What is Thread Scheduler in java?

In Java, when we create the threads, they are supervised with the help of a Thread Scheduler, which is the part of JVM. Thread scheduler is only responsible for deciding which thread should be executed. Thread scheduler uses two mechanisms for scheduling the threads: Preemptive and Time Slicing.

Java thread scheduler also works for deciding the following for a thread:

- It selects the priority of the thread.
- It determines the waiting time for a thread
- It checks the Nature of thread

29) Does each thread have its stack in multithreaded programming?

Yes, in multithreaded programming every thread maintains its own or separate stack area in memory due to which every thread is independent of each other.

30) How is the safety of a thread achieved?

If a method or class object can be used by multiple threads at a time without any race condition, then the class is thread-safe. Thread safety is used to make a program safe to use in multithreaded programming. It can be achieved by the following ways:

- Synchronization
- Using Volatile keyword
- Using a lock based mechanism
- Use of atomic wrapper classes

31) What is race-condition?

A Race condition is a problem which occurs in the multithreaded programming when various threads execute simultaneously accessing a shared resource at the same time. The proper use of synchronization can avoid the Race condition.

32) What is the volatile keyword in java?

Volatile keyword is used in multithreaded programming to achieve the thread safety, as a change in one volatile variable is visible to all other threads so one variable can be used by one thread at a time.

33) What do you understand by thread pool?

- Java Thread pool represents a group of worker threads, which are waiting for the task to be allocated.
- Threads in the thread pool are supervised by the service provider which pulls one thread from the pool and assign a job to it.
- After completion of the given task, thread again came to the thread pool.
- The size of the thread pool depends on the total number of threads kept at reserve for execution.

The advantages of the thread pool are :

- Using a thread pool, performance can be enhanced.
- Using a thread pool, better system stability can occur.

Java Collections Interview Questions

In Java, collection interview questions are most asked by the interviewers. Here is the list of the most asked collections interview questions with answers.

1) What is the Collection framework in Java?

Collection Framework is a combination of classes and interface, which is used to store and manipulate the data in the form of objects. It provides various classes such as ArrayList, Vector, Stack, and HashSet, etc. and interfaces such as List, Queue, Set, etc. for this purpose.

2) What are the main differences between array and collection?

Array and Collection are somewhat similar regarding storing the references of objects and manipulating the data, but they differ in many ways. The main differences between the array and Collection are defined below:

- Arrays are always of fixed size, i.e., a user can not increase or decrease the length of the array according to their requirement or at runtime, but In Collection, size can be changed dynamically as per need.
- Arrays can only store homogeneous or similar type objects, but in Collection, heterogeneous objects can be stored.
- Arrays cannot provide the ?ready-made? methods for user requirements as sorting, searching, etc. but Collection includes readymade methods to use.

3) Explain various interfaces used in Collection framework?

Collection framework implements various interfaces, Collection interface and Map interface (java.util.Map) are the mainly used interfaces of Java Collection Framework. List of interfaces of Collection Framework is given below

1. Collection interface: Collection (java.util.Collection) is the primary interface, and every collection must implement this interface.

Syntax:

```
public interface Collection<E>extends Iterable
```

Where <E> represents that this interface is of Generic type

2. List interface: List interface extends the Collection interface, and it is an ordered collection of objects. It contains duplicate elements. It also allows random access of elements.

Syntax:

```
public interface List<E> extends Collection<E>
```

3. Set interface: Set (java.util.Set) interface is a collection which cannot contain duplicate elements. It can only include inherited methods of Collection interface

Syntax:

```
public interface Set<E> extends Collection<E>
```

Queue interface: Queue (java.util.Queue) interface defines queue data structure, which stores the elements in the form FIFO (first in first out).

Syntax:

```
public interface Queue<E> extends Collection<E>
```

4. Dequeue interface: it is a double-ended-queue. It allows the insertion and removal of elements from both ends. It implants the properties of both Stack and queue so it can perform LIFO (Last in first out) stack and FIFO (first in first out) queue, operations.

Syntax:

```
public interface Dequeue<E> extends Queue<E>
```

5. Map interface: A Map (java.util.Map) represents a key, value pair storage of elements. Map interface does not implement the Collection interface. It can only contain a unique key but can have duplicate elements. There are two interfaces which implement Map in java that are Map interface and Sorted Map.

4) What is the difference between ArrayList and Vector?

No.	ArrayList	Vector
1)	ArrayList is not synchronized.	Vector is synchronized.
2)	ArrayList is not a legacy class.	Vector is a legacy class.
3)	ArrayList increases its size by 50% of the array size.	Vector increases its size by doubling the size.
4)	ArrayList is not ?thread-safe? as it is not synchronized.	Vector list is ?thread-safe? as it?s every met synchronized.

No.	ArrayList	LinkedList
1)	ArrayList uses a dynamic array.	LinkedList uses a doubly linked list.
2)	ArrayList is not efficient for manipulation because too much is required.	LinkedList is efficient for manipulation.
3)	ArrayList is better to store and fetch data.	LinkedList is better to manipulate data.

4)	ArrayList provides random access.	LinkedList does not provide random access.
5)	ArrayList takes less memory overhead as it stores only object	LinkedList takes more memory overhead, as it stores the object as well as the address of that object.

5) What is the difference between ArrayList and LinkedList?

6) What is the difference between Iterator and ListIterator?

Iterator traverses the elements in the forward direction only whereas ListIterator traverses the elements into forward and backward direction.

No.	Iterator	ListIterator
1)	The Iterator traverses the elements in the forward direction only.	ListIterator traverses the elements in backward and forward directions both.
2)	The Iterator can be used in List, Set, and Queue.	ListIterator can be used in List only.
3)	The Iterator can only perform remove operation while traversing the collection.	ListIterator can perform ?add,? ?remove,? and ?set? operation while traversing the collection.

No.	Iterator	Enumeration
1)	The Iterator can traverse legacy and non-legacy elements.	Enumeration can traverse only legacy elements.
2)	The Iterator is fail-fast.	Enumeration is not fail-fast.
3)	The Iterator is slower than Enumeration.	Enumeration is faster than Iterator.
4)	The Iterator can perform remove operation while traversing the collection.	The Enumeration can perform only traverse operation on the collection.

7) What is the difference between Iterator and Enumeration?

8) What is the difference between List and Set?

The List and Set both extend the collection interface. However, there are some differences between the both which are listed below.

- The List can contain duplicate elements whereas Set includes unique items.
- The List is an ordered collection which maintains the insertion order whereas Set is an unordered collection which does not preserve the insertion order.
- The List interface contains a single legacy class which is Vector class whereas Set interface does not have any legacy class.
- The List interface can allow n number of null values whereas Set interface only allows a single null value.

9) What is the difference between HashSet and TreeSet?

The HashSet and TreeSet, both classes, implement Set interface. The differences between the both are listed below.

- HashSet maintains no order whereas TreeSet maintains ascending order.
- HashSet impended by hash table whereas TreeSet implemented by a Tree structure.
- HashSet performs faster than TreeSet.
- HashSet is backed by HashMap whereas TreeSet is backed by TreeMap.

10) What is the difference between Set and Map?

The differences between the Set and Map are given below.

- Set contains values only whereas Map contains key and values both.
- Set contains unique values whereas Map can contain unique Keys with duplicate values.
- Set holds a single number of null value whereas Map can include a single null key with n number of null values.

11) What is the difference between HashSet and HashMap?

The differences between the HashSet and HashMap are listed below.

- HashSet contains only values whereas HashMap includes the entry (key, value). HashSet can be iterated, but HashMap needs to convert into Set to be iterated.
- HashSet implements Set interface whereas HashMap implements the Map interface
- HashSet cannot have any duplicate value whereas HashMap can contain duplicate values with unique keys.
- HashSet contains the only single number of null value whereas HashMap can hold a single null key with n number of null values.

12) What is the difference between HashMap and TreeMap?

The differences between the HashMap and TreeMap are given below.

- HashMap maintains no order, but TreeMap maintains ascending order.
- HashMap is implemented by hash table whereas TreeMap is implemented by a Tree structure.
- HashMap can be sorted by Key or value whereas TreeMap can be sorted by Key.
- HashMap may contain a null key with multiple null values whereas TreeMap cannot hold a null key but can have multiple null values.

13) What is the difference between HashMap and Hashtable?

No.	HashMap	Hashtable
1)	HashMap is not synchronized.	Hashtable is synchronized.
2)	HashMap can contain one null key and multiple null values.	Hashtable cannot contain any null key or null value.
3)	HashMap is not ?thread-safe,? so it is useful for non-threaded applications.	Hashtable is thread-safe, and it can be shared between various threads.
4)	4) HashMap inherits the AbstractMap class	Hashtable inherits the Dictionary class.

14) What is the difference between Collection and Collections?

The differences between the Collection and Collections are given below.

- The Collection is an interface whereas Collections is a class.
- The Collection interface provides the standard functionality of data structure to List, Set, and Queue. However, Collections class is to sort and synchronize the collection elements.
- The Collection interface provides the methods that can be used for data structure whereas Collections class provides the static methods which can be used for various operation on a collection.

15) What is the difference between Comparable and Comparator?

No.	Comparable	Comparator

1)	Comparable provides only one sort of sequence.	The Comparator provides multiple sorts of sequences.
2)	It provides one method named compareTo().	It provides one method named compare().
3)	It is found in java.lang package.	It is located in java.util package.
4)	If we implement the Comparable interface, The actual class is modified.	The actual class is not changed.

16) What do you understand by BlockingQueue?

BlockingQueue is an interface which extends the Queue interface. It provides concurrency in the operations like retrieval, insertion, deletion. While retrieval of any element, it waits for the queue to be non-empty. While storing the elements, it waits for the available space. BlockingQueue cannot contain null elements, and implementation of BlockingQueue is thread-safe.

Syntax:

```
public interface BlockingQueue<E> extends Queue <E>
```

21) What is the advantage of the generic collection?

There are three main advantages of using the generic collection.

- If we use the generic class, we don't need typecasting.
- It is type-safe and checked at compile time.
- Generic confirms the stability of the code by making it bug detectable at compile time.

26) What is the difference between Array and ArrayList?

SN	Array	ArrayList

1	The Array is of fixed size, means we cannot resize the array as per need.	ArrayList is not of the fixed size we can change the size dynamically.
2	Arrays are of the static type.	ArrayList is of dynamic size.
3	Arrays can store primitive data types as well as objects.	ArrayList cannot store the primitive data types it can only store the objects.

28) How to convert ArrayList to Array and Array to ArrayList?

We can convert an Array to ArrayList by using the asList() method of Arrays class. asList() method is the static method of Arrays class and accepts the List object. Consider the following syntax:

```
Arrays.asList(item)
```

We can convert an ArrayList to Array using toArray() method of the ArrayList class. Consider the following syntax to convert the ArrayList to the List object.

```
List_object.toArray(new String[List_object.size()])
```

31) How to reverse ArrayList?

To reverse an ArrayList, we can use reverse() method of Collections class. Consider the following example.

```
import java.util.ArrayList;
import java.util.Collection;
import java.util.Collections;
import java.util.Iterator;
import java.util.List;
public class ReverseArrayList {
public static void main(String[] args) {
        List list = new ArrayList<>();
        list.add(10);
        list.add(50);
        list.add(30);
        Iterator i = list.iterator();
        System.out.println("printing the list....");
        while(i.hasNext())
        {
            System.out.println(i.next());
        }
        Iterator i2 = list.iterator();
        Collections.reverse(list);
        System.out.println("printing list in reverse order....");
        while(i2.hasNext())
        {
            System.out.println(i2.next());
        }
       }
    }
```

Output

```
printing the list....
10
50
30
printing list in reverse order....
30
50
10
```

32) How to sort ArrayList in descending order?

To sort the ArrayList in descending order, we can use the reverseOrder method of Collections class. Consider the following example.

```
import java.util.ArrayList;
import java.util.Collection;
import java.util.Collections;
import java.util.Comparator;
import java.util.Iterator;
import java.util.List;

public class ReverseArrayList {
public static void main(String[] args) {
    List list = new ArrayList<>();
    list.add(10);
    list.add(50);
    list.add(30);
    list.add(60);
    list.add(20);
    list.add(90);

    Iterator i = list.iterator();
    System.out.println("printing the list....");
    while(i.hasNext())
    {
        System.out.println(i.next());
    }

   Comparator cmp = Collections.reverseOrder();
   Collections.sort(list,cmp);
    System.out.println("printing list in descending order....");
    Iterator i2 = list.iterator();
    while(i2.hasNext())
    {
```

```
                System.out.println(i2.next());
            }

        }
        }
```

Output

```
printing the list....
10
50
30
60
20
90
printing list in descending order....
90
60
50
30
20
10
```

34) When to use ArrayList and LinkedList?

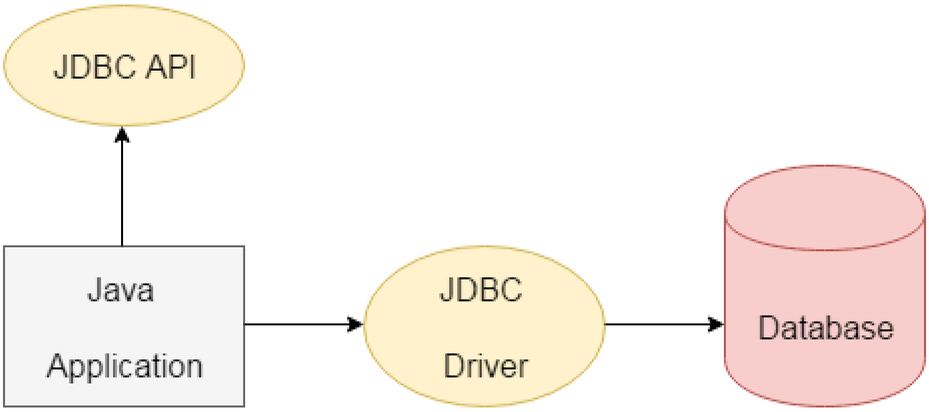
LinkedLists are better to use for the update operations whereas ArrayLists are better to use for the search operations.

JDBC Interview Questions

A list of top frequently asked JDBC interview questions and answers is given below.

1) What is JDBC?

JDBC is a Java API that is used to connect and execute the query to the database. JDBC API uses JDBC drivers to connect to the database. JDBC API can be used to access tabular data stored into any relational database.



2) What is JDBC Driver?

JDBC Driver is a software component that enables Java application to interact with the database. There are 4 types of JDBC drivers:

1. **JDBC-ODBC bridge driver:** The JDBC-ODBC bridge driver uses the ODBC driver to connect to the database. The JDBC-ODBC bridge driver converts JDBC method calls into the ODBC function calls. This is now discouraged because of the thin driver. It is easy to use and can be easily connected to any database.
2. **Native-API driver (partially java driver):** The Native API driver uses the client-side libraries of the database. The driver converts JDBC method calls into native calls of the database API. It is not written entirely in Java. Its performance is better than JDBC-ODBC bridge driver. However, the native driver must be installed on each client machine.
3. **Network Protocol driver (fully java driver):** The Network Protocol driver uses middleware (application server) that converts JDBC calls directly or indirectly into the vendor-specific database protocol. It is

entirely written in Java. There is no requirement of the client-side library because of the application server that can perform many tasks like auditing, load balancing, logging, etc.

4. **Thin driver (fully java driver):** The thin driver converts JDBC calls directly into the vendor-specific database protocol. That is why it is known as the thin driver. It is entirely written in Java language. Its performance is better than all other drivers however these drivers depend upon the database.

---

3) What are the steps to connect to the database in java?

The following steps are used in database connectivity.

- **Registering the driver class:**

  The forName() method of the Class class is used to register the driver class. This method is used to load the driver class dynamically. Consider the following example to register OracleDriver class.

```
1.    Class.forName("oracle.jdbc.driver.OracleDriver");
```

- **Creating connection:**

  The getConnection() method of DriverManager class is used to establish the connection with the database. The syntax of the getConnection() method is given below.

```
1.    1) public static Connection getConnection(String url)throws SQ
      LException
2.    2) public static Connection getConnection(String url,String nam
      e,String password)
3.    throws SQLException
```

Consider the following example to establish the connection with the Oracle database.

```
Connection con=DriverManager.getConnection(
"jdbc:oracle:thin:@localhost:1521:xe","system","password");
```

- **Creating the statement:**

The createStatement() method of Connection interface is used to create the Statement. The object of the Statement is responsible for executing queries with the database.

```
public Statement createStatement()throws SQLException
```

consider the following example to create the statement object

```
Statement stmt=con.createStatement();
```

- **Executing the queries:**

The executeQuery() method of Statement interface is used to execute queries to the database. This method returns the object of ResultSet that can be used to get all the records of a table.

Syntax of executeQuery() method is given below.

```
public ResultSet executeQuery(String sql)throws SQLException
```

Example to execute the query

```
ResultSet rs=stmt.executeQuery("select * from emp");
while(rs.next()){
System.out.println(rs.getInt(1)+" "+rs.getString(2));
}
```

However, to perform the insert and update operations in the database, executeUpdate() method is used which returns the boolean value to indicate the successful completion of the operation.

- **Closing connection:**

  By closing connection, object statement and ResultSet will be closed automatically. The close() method of Connection interface is used to close the connection.

  Syntax of close() method is given below.

```
public void close()throws SQLException
```

Consider the following example to close the connection.

```
con.close();

```

4) What are the JDBC API components?

The java.sql package contains following interfaces and classes for JDBC API.

Interfaces:

- **Connection:** The Connection object is created by using getConnection() method of DriverManager class. DriverManager is the factory for connection.

- **Statement:** The Statement object is created by using createStatement() method of Connection class. The Connection interface is the factory for Statement.

- **PreparedStatement:** The PrepareStatement object is created by using prepareStatement() method of Connection class. It is used to execute the parameterized query.

- **ResultSet:** The object of ResultSet maintains a cursor pointing to a row of a table. Initially, cursor points before the first row. The executeQuery() method of Statement interface returns the ResultSet object.

- **ResultSetMetaData:** The object of ResultSetMetaData interface cotains the information about the data (table) such as numer of columns, column name, column type, etc. The getMetaData() method of ResultSet returns the object of ResultSetMetaData.

- **DatabaseMetaData:** DatabaseMetaData interface provides methods to get metadata of a database such as the database product name, database product version, driver name, name of the total number of tables, the name of the total number of views, etc. The getMetaData() method of Connection interface returns the object of DatabaseMetaData.

- **CallableStatement:** CallableStatement interface is used to call the stored procedures and functions. We can have business logic on the database through the use of stored procedures and functions that will make the performance better because these are precompiled. The prepareCall() method of Connection interface returns the instance of CallableStatement.

Classes:

- **DriverManager:** The DriverManager class acts as an interface between the user and drivers. It keeps track of the drivers that are available and handles establishing a connection between a database and the appropriate driver. It contains several methods to keep the interaction between the user and drivers.

- **Blob:** Blob stands for the binary large object. It represents a collection of binary data stored as a single entity in the database management system.

- **Clob:** Clob stands for Character large object. It is a data type that is used by various database management systems to store character files. It is similar to Blob except for the difference that BLOB represent binary data such as images, audio and video files, etc. whereas Clob represents character stream data such as character files, etc.

- **SQLException** It is an Exception class which provides information on database access errors.

5) What are the JDBC statements?

In JDBC, Statements are used to send SQL commands to the database and receive data from the database. There are various methods provided by JDBC statements such as execute(), executeUpdate(), executeQuery, etc. which helps you to interact with the database.

There is three type of JDBC statements given in the following table.

Statements	Explanation
Statement	Statement is the factory for resultset. It is used for general purpose access database. It executes a static SQL query at runtime.
PreparedStatement	The PreparedStatement is used when we need to provide input parame the query at runtime.
CallableStatement	CallableStatement is used when we need to access the database procedures. It can also accept runtime parameters.

6) What is the return type of Class.forName() method?

The Class.forName() method returns the object of java.lang.Class object.

7) What are the differences between Statement and PreparedStatement interface?

Statement	PreparedStatement
The Statement interface provides methods to execute queries with the database. The statement interface is a factory of ResultSet; i.e., it provides the factory method to get the object of ResultSet.	The PreparedStatement interface subinterface of Statement. It is us execute the parameterized query.
In the case of Statement, the query is compiled each time we run the program.	In the case of PreparedStatemen query is compiled only once.
The Statement is mainly used in the case when we need to run the static query at runtime.	PreparedStatement is used when we to provide input parameters to the at runtime.

8) How can we set null value in JDBC PreparedStatement?

By using setNull() method of PreparedStatement interface, we can set the null value to an index. The syntax of the method is given below.

```
void setNull(int parameterIndex, int sqlType) throws SQLException

```

9) What are the benefits of PreparedStatement over Statement?

The benefits of using PreparedStatement over Statement interface is given below.

- The PreparedStatement performs faster as compare to Statement because the Statement needs to be compiled everytime we run the code whereas the PreparedStatement compiled once and then execute only on runtime.
- PreparedStatement can execute Parameterized query whereas Statement can only run static queries.
- The query used in PreparedStatement is appeared to be similar every time. Therefore, the database can reuse the previous access plan whereas, Statement inline the parameters into the String, therefore, the query doesn't appear to be same everytime which prevents cache reusage.

10) What are the differences between execute, executeQuery, and executeUpdate?

execute	executeQuery	executeUpdate
The execute method can be used for any SQL statements(Select and Update both).	The executeQuery method can be used only with the select statement.	The executeUpdate metho be used to update/delete operations in the database.
The execute method returns a boolean type value where true indicates that the ResultSet s returned which can later be extracted and false indicates that the integer or void value is returned.	The executeQuery() method returns a ResultSet object which contains the data retrieved by the select statement.	The executeUpdate() m returns an integer representing the numbe records affected where 0 inc that query returns nothing.

11) What are the different types of ResultSet?

ResultSet is categorized by the direction of the reading head and sensitivity or insensitivity of the result provided by it. There are three general types of ResultSet.

Type	Description
ResultSet.TYPE_Forward_ONLY	The cursor can move in the forward direction only.
ResultSet.TYPE_SCROLL_INSENSITIVE	The cursor can move in both the direction (forwar backward). The ResultSet is not sensitive to the changes by the others to the database.
ResultSet.TYPE_SCROLL_SENSITIVE	The cursor can move in both the direction. The Resu sensitive to the changes made by the others to the datab

12) What are the differences between ResultSet and RowSet?

ResultSet	RowSet
ResultSet cannot be serialized as it maintains the connection with the database.	RowSet is disconnected from the database and can be serialize
ResultSet object is not a JavaBean object	ResultSet Object is a JavaBean object.
ResultSet is returned by the executeQuery() method of Statement Interface.	Rowset Interface extends ResultSet Interface and returned by the RowSetProvider.newFactory().createJdbcRowSet() method.
ResultSet object is non-scrollable and non-updatable by default.	RowSet object is scrollable and updatable by default.

14) What is the role of the JDBC DriverManager class?

The DriverManager class acts as an interface between user and drivers. It keeps track of the drivers that are available and handles establishing a connection between a database and the appropriate driver. The

DriverManager class maintains a list of Driver classes that have registered themselves by calling the method DriverManager.registerDriver().

More details.

15) What are the functions of the JDBC Connection interface?

The **Connection interface** maintains a session with the database. It can be used for transaction management. It provides factory methods that return the instance of Statement, PreparedStatement, CallableStatement, and DatabaseMetaData.

More details.

16) What does the JDBC ResultSet interface?

The ResultSet object represents a row of a table. It can be used to change the cursor pointer and get the information from the database. By default, ResultSet object can move in the forward direction only and is not updatable. However, we can make this object to move the forward and backward direction by passing either TYPE_SCROLL_INSENSITIVE or TYPE_SCROLL_SENSITIVE in createStatement(int, int) method.

More details.

17) What does the JDBC ResultSetMetaData interface?

The ResultSetMetaData interface returns the information of table such as the total number of columns, column name, column type, etc.

More details.

18) What does the JDBC DatabaseMetaData interface?

The DatabaseMetaData interface returns the information of the database such as username, driver name, driver version, number of tables, number of views, etc. Consider the following example.

```java
import java.sql.*;
class Dbmd{
public static void main(String args[]){
try{
Class.forName("oracle.jdbc.driver.OracleDriver");

Connection con=DriverManager.getConnection(
"jdbc:oracle:thin:@localhost:1521:xe","system","oracle");
DatabaseMetaData dbmd=con.getMetaData();

System.out.println("Driver Name: "+dbmd.getDriverName());
System.out.println("Driver Version: "+dbmd.getDriverVersion());
System.out.println("UserName: "+dbmd.getUserName());
System.out.println("Database Product Name: "+dbmd.getDatabaseProductName());
System.out.println("Database Product Version: "+dbmd.getDatabaseProductVersion());

con.close();
}catch(Exception e){ System.out.println(e);}
}
}
```

Output

```
Driver Name: Oracle JDBC Driver
Driver Version: 10.2.0.1.0XE
Database Product Name: Oracle
Database Product Version: Oracle Database 10g Express Edition Release
10.2.0.1.0 -Production
```

More details.

19) Which interface is responsible for transaction management in JDBC?

The **Connection interface** provides methods for transaction management such as commit(), rollback() etc.

More details.

20) What is batch processing and how to perform batch processing in JDBC?

By using the batch processing technique in JDBC, we can execute multiple queries. It makes the performance fast. The java.sql.Statement and java.sql.PreparedStatement interfaces provide methods for batch processing. The batch processing in JDBC requires the following steps.

- Load the driver class
- Create Connection
- Create Statement
- Add query in the batch
- Execute the Batch
- Close Connection

Consider the following example to perform batch processing using the Statement interface.

```
import java.sql.*;
class FetchRecords{
public static void main(String args[])throws Exception{
Class.forName("oracle.jdbc.driver.OracleDriver");
Connection con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","system","oracle");
con.setAutoCommit(false);

Statement stmt=con.createStatement();
stmt.addBatch("insert into user420 values(190,'abhi',40000)");
stmt.addBatch("insert into user420 values(191,'umesh',50000)");

stmt.executeBatch();//executing the batch

con.commit();
con.close();
}}
```

More details.

21) What are CLOB and BLOB data types in JDBC?

BLOB: Blob can be defined as the variable-length, binary large object which is used to hold the group of Binary data such as voice, images, and mixed media. It can hold up to 2GB data on MySQL database and 128 GB on Oracle database. BLOB is supported by many databases such as MySQL, Oracle, and DB2 to store the binary data (images, video, audio, and mixed media).

CLOB: Clob can be defined as the variable-length, character-large object which is used to hold the character-based data such as files in many databases. It can hold up to 2 GB on MySQL database, and 128 GB on Oracle Database. A CLOB is considered as a character string.

22) What are the different types of lockings in JDBC?

A lock is a certain type of software mechanism by using which, we can restrict other users from using the data resource. There are four type of locks given in JDBC that are described below.

- **Row and Key Locks:** These type of locks are used when we update the rows.
- **Page Locks:** These type of locks are applied to a page. They are used in the case, where a transaction remains in the process and is being updated, deleting, or inserting some data in a row of the table. The database server locks the entire page that contains the row. The page lock can be applied once by the database server.
- **Table locks:** Table locks are applied to the table. It can be applied in two ways, i.e., shared and exclusive. Shared lock lets the other

transactions to read the table but not update it. However, The exclusive lock prevents others from reading and writing the table.

- **Database locks:** The Database lock is used to prevent the read and update access from other transactions when the database is open.

23) How can we store and retrieve images from the database?

By using the PreparedStatement interface, we can store and retrieve images. Create a table which contains two columns namely NAME and PHOTO.

```
CREATE TABLE  "IMGTABLE"
   (    "NAME" VARCHAR2(4000),
    "PHOTO" BLOB
   )
```

Consider the following example to store the image in the database.

```
import java.sql.*;
import java.io.*;
public class InsertImage {
public static void main(String[] args) {
try{
Class.forName("oracle.jdbc.driver.OracleDriver");
Connection con=DriverManager.getConnection(
"jdbc:oracle:thin:@localhost:1521:xe","system","oracle");

PreparedStatement ps=con.prepareStatement("insert into imgtable values(?,?)");
ps.setString(1,"sonoo");

FileInputStream fin=new FileInputStream("d:\\g.jpg");
ps.setBinaryStream(2,fin,fin.available());
int i=ps.executeUpdate();
System.out.println(i+" records affected");

con.close();
}catch (Exception e) {e.printStackTrace();}
}
}
```

Consider the following example to retrieve the image from the table.

```
import java.sql.*;
import java.io.*;
public class RetrieveImage {
public static void main(String[] args) {
try{
Class.forName("oracle.jdbc.driver.OracleDriver");
Connection con=DriverManager.getConnection(
"jdbc:oracle:thin:@localhost:1521:xe","system","oracle");

PreparedStatement ps=con.prepareStatement("select * from imgtable");
ResultSet rs=ps.executeQuery();
if(rs.next()){//now on 1st row

Blob b=rs.getBlob(2);//2 means 2nd column data
byte barr[]=b.getBytes(1,(int)b.length());//1 means first image

FileOutputStream fout=new FileOutputStream("d:\\sonoo.jpg");
fout.write(barr);

fout.close();
}//end of if
System.out.println("ok");

con.close();
}catch (Exception e) {e.printStackTrace();  }
}
}
```

Servlet interview questions

There is a list of 30 servlet interview questions for beginners and professionals. If you know any servlet interview question that has not been included here, kindly post your question in the Ask Question section.

1) How many objects of a servlet is created?

Only one object at the time of first request by servlet or web container.

2) What is the life-cycle of a servlet?

1. Servlet is loaded
2. servlet is instantiated
3. servlet is initialized
4. service the request
5. servlet is destroyed

3) What are the life-cycle methods for a servlet?

Method	Description
public void init(ServletConfig config)	It is invoked only once when first r comes for the servlet. It is used to in the servlet.
public void service(ServletRequest request,ServletResponse)throws ServletException,IOException	It is invoked at each request.The se method is used to service the request
public void destroy()	It is invoked only once when ser unloaded.

more details...

4) Who is responsible to create the object of servlet?

The web container or servlet container.

5) When servlet object is created?

At the time of first request.

6) What is difference between Get and Post method?

Get	Post
1) Limited amount of data can be sent because data is sent in header.	Large amount of data can be sent becaus is sent in body.
2) Not Secured because data is exposed in URL bar.	Secured because data is not exposed i bar.
3) Can be bookmarked	Cannot be bookmarked
4) Idempotent	Non-Idempotent
5) It is more efficient and used than Post	It is less efficient and used

7) What is difference between PrintWriter and ServletOutputStream?

PrintWriter is a character-stream class where as ServletOutputStream is a byte-stream class. The PrintWriter class can be used to write only character-based information whereas ServletOutputStream class can be used to write primitive values as well as character-based information.

8) What is difference between GenericServlet and HttpServlet?

The GenericServlet is protocol independent whereas HttpServlet is HTTP protocol specific. HttpServlet provides additional functionalities such as state management etc.

9) What is servlet collaboration?

When one servlet communicates to another servlet, it is known as servlet collaboration. There are many ways of servlet collaboration:

- RequestDispacher interface
- sendRedirect() method etc.

10) What is the purpose of RequestDispatcher Interface?

The RequestDispacher interface provides the facility of dispatching the request to another resource it may be html, servlet or jsp. This interceptor can also be used to include the content of antoher resource.

11) Can you call a jsp from the servlet?

Yes, one of the way is RequestDispatcher interface for example:

```
RequestDispatcher rd=request.getRequestDispatcher("/login.jsp");
rd.forward(request,response);
```

12) Difference between forward() method and sendRedirect() method ?

forward() method	sendRedirect() method
1) forward() sends the same request to another resource.	1) sendRedirect() method sends new request always bec uses the URL bar of the browser.
2) forward() method works at server side.	2) sendRedirect() method works at client side.
3) forward() method works within the server only.	3) sendRedirect() method works within and outside the s

13) What is difference between ServletConfig and ServletContext?

The container creates object of ServletConfig for each servlet whereas object of ServletContext is created for each web application.

14) What is Session Tracking?

Session simply means a particular interval of time.

Session Tracking is a way to maintain state of an user.Http protocol is a stateless protocol.Each time user requests to the server, server treats the request as the new request.So we need to maintain the state of an user to recognize to particular user.

15) What are Cookies?

A cookie is a small piece of information that is persisted between the multiple client requests. A cookie has a name, a single value, and optional attributes such as a comment, path and domain qualifiers, a maximum age, and a version number.

16) What is difference between Cookies and HttpSession?

Cookie works at client side whereas HttpSession works at server side.

17) What is filter?

A filter is an object that is invoked either at the preprocessing or postprocessing of a request. It is pluggable.

more details...

18) How can we perform any action at the time of deploying the project?

By the help of ServletContextListener interface.

19) What is the disadvantage of cookies?

It will not work if cookie is disabled from the browser.

more details...

20) How can we upload the file to the server using servlet?

One of the way is by MultipartRequest class provided by third party.

more details...

21) What is load-on-startup in servlet?

The load-on-startup element of servlet in web.xml is used to load the servlet at the time of deploying the project or server start. So it saves time for the response of first request.

22) What if we pass negative value in load-on-startup?

It will not affect the container, now servlet will be loaded at first request.

more details...

23) What is war file?

A war (web archive) file specifies the web elements. A servlet or jsp project can be converted into a war file. Moving one servlet project from one place to another will be fast as it is combined into a single file.

more details...

24) How to create war file?

The war file can be created using jar tool found in jdk/bin directory. If you are using Eclipse or Netbeans IDE, you can export your project as a war file.

To create war file from console, you can write following code.

```
1.          jar -cvf abc.war *
```

Now all the files of current directory will be converted into abc.war file.

more details...

25) What are the annotations used in Servlet 3?

There are mainly 3 annotations used for the servlet.

1. @WebServlet : for servlet class.
2. @WebListener : for listener class.
3. @WebFilter : for filter class.

26) Which event is fired at the time of project deployment and undeployment?

ServletContextEvent.

more details...

27) Which event is fired at the time of session creation and destroy?

HttpSessionEvent.

more details...

28) Which event is fired at the time of setting, getting or removing attribute from application scope?

ServletContextAttributeEvent.

29) What is the use of welcome-file-list?

It is used to specify the welcome file for the project.

more details...

30) What is the use of attribute in servlets?

Attribute is a map object that can be used to set, get or remove in request, session or application scope. It is mainly used to share information between one servlet to another.

JSP Interview Questions

There is a list of top 40 frequently asked JSP interview questions and answers for freshers and professionals. If you know any JSP interview question that has not been included here, post your question in the Ask Question section.

1) What is JSP?

Java Server Pages technology (JSP) is a server-side programming language used to create a dynamic web page in the form of HyperText Markup Language (HTML). It is an extension to the servlet technology.

A JSP page is internally converted into the servlet. JSP has access to the entire family of the Java API including JDBC API to access enterprise database. Hence, Java language syntax has been used in the java server pages (JSP). The JSP pages are more accessible to maintain than Servlet because we can separate designing and development. It provides some additional features such as Expression Language, Custom Tags, etc.

More details.

2) What are the life-cycle methods for a JSP?

Method	Description
public void jspInit()	It is invoked only once, same method of the servlet.
public void _jspService(ServletRequest request,ServletResponse)throws ServletException,IOException	It is invoked at each request, sa service() method of the servlet.
public void jspDestroy()	It is invoked only once, sa destroy() method of the servlet.

3) List out some advantages of using JSP.

- Better performance.

- The compilation of JSP is done before it is processed by the server which eradicates the need for loading of interpreter and code script each time.
- JSP has access to all-powerful enterprises.
  Easy to maintain: JSP can be easily managed because we can easily separate our business logic with presentation logic. In Servlet technology, we mix our business logic with the presentation logic.
- JSP can also be used in combination with servlets.

4) Give the syntax for JSP comments.

The syntax for JSP comments is:

```
1.      <%-- Comment --%>
```

5) What is the difference between hide comment and output comment?

The JSP comment is called hide comment whereas HTML comment is called output comment. If a user views the source of the page, the JSP comment will not be shown whereas HTML comment will be displayed.

6) What are the JSP implicit objects?

JSP provides nine implicit objects by default. They are as follows:

Object	Type
1) out	JspWriter
2) request	HttpServletRequest
3) response	HttpServletResponse
4) config	ServletConfig
5) session	HttpSession
6) application	ServletContext
7) pageContext	PageContext
8) page	Object
9) exception	Throwable

More details.

7) What is the difference between include directive and include action?

include directive	include action
1) The include directive includes the content at page translation time.	1) The include action includes the content at r time.
2) The include directive includes the original content of the page, so page size increases at runtime	2) The include action doesn't include the o content rather invokes the include() meth Vendor provided class.
3) It's better for static pages.	3) 3) It's better for dynamic pages.

8) Is JSP technology extensible?

Yes. JSP technology is extensible through the development of custom actions, or tags, which are encapsulated in tag libraries.

9) How can I implement a thread-safe JSP page? What are the advantages and Disadvantages of using it?

You can make your JSPs thread-safe by having them implement the SingleThreadModel interface. This is done by adding the directive <%@ page isThreadSafe="false" %> within your JSP page.

10) How can I prevent the output of my JSP or Servlet pages from being cached by the browser?

(OR) How to disable caching on the back button of the browser?

```
<%
response.setHeader("Cache-Control","no-store");
response.setHeader("Pragma","no-cache");
response.setHeader ("Expires", "0"); //prevents caching at the proxy server
%>
```

11) How can we handle the exceptions in JSP?

There are two ways to perform exception handling, one is by the errorPage element of page directive, and second is by the error-page element of the web.xml file.

More details.

12) What are the two ways to include the result of another page. ?

There are two ways to include the result of another page:

- By include directive
- By include action

13) How can we forward the request from JSP page to the servlet?

Yes of course! With the help of "forward action" tag, but we need to give the URL-pattern of the servlet.

forward action tag

14) Can we use the exception implicit object in any JSP page?

No. The exception implicit object can only be used in the error page which defines it with the isErrorPage attribute of page directive.

More details.

15) How is JSP used in the MVC model?

JSP is usually used for presentation in the MVC pattern (Model View Controller), i.e., it plays the role of the view. The controller deals with calling the model and the business classes which in turn get the data, and this data is then presented to the JSP for rendering on to the client.

forward action tag

16) What are context initialization parameters?

Context initialization parameters are specified by the <context-param> in the web.xml file, and these are initialization parameter for the whole application and not specific to any servlet or JSP.

More details.

17) What are the different scope values for the <jsp:useBean> tag?

There are 4 values:

1. page
2. request
3. session
4. application

18) What do JSP literals consist of?

- Boolean
- Integer
- Floating point
- String
- Null

19) What is the purpose of <jsp:useBean>?

The jsp:useBean action searches for the existence of the object with specified name. If not found, it creates one.

20) What is the purpose of <jsp:setProperty >?

This action sets the properties of a bean.

21) What is the purpose of <jsp:getProperty >?

This action retrieves the properties of a bean.

22) List out the various scope values of JSP action.

The possible scope values are:

- page
- request
- session

- application

23) What is the use of 'out' implicit object?

The object is used to give a response to contents.

24) Give the use of session object.

The object is used between the client requests for the tracking of client sessions.

25) Give the use of exception object.

The object is used for the generation of a response to the errors thrown.

26) What is the difference between ServletContext and PageContext?-

ServletContext gives the information about the container whereas PageContext gives the information about the Request.

27) What is the difference in using request.getRequestDispatcher() and context.getRequestDispatcher()?

request.getRequestDispatcher(path) is used to create it we need to give the relative path of the resource whereas context.getRequestDispatcher(path)to create it we need to give the absolute path of the resource.

28) What is EL in JSP?

The Expression Language(EL) is used in JSP to simplify the accessibility of objects. It provides many objects that can be used directly like param, requestScope, sessionScope, applicationScope, request, session, etc.

29) What are the primary differences between the JSP custom tags and java beans?

- Custom tags can manipulate JSP content whereas beans cannot.
- Complex operations can be reduced to a significantly simpler form with custom tags than with beans.
- Custom tags require quite a bit more work to set up than do beans.
- Custom tags are available only in JSP 1.1 and later, but beans can be used in all JSP 1.x versions.

30) Can an interface be implemented in the JSP file?

No.

31) What is JSTL?

JSP Standard Tag Library is a library of predefined tags that ease the development of JSP.

More details.

32) How many tags are provided in JSTL?

There is 5 type of JSTL tags.

1. core tags
2. sql tags
3. xml tags
4. internationalization tags
5. functions tags

More details.

33) Which directive is used in JSP custom tag?

The JSP taglib directive.

34) What are the three tags used in JSP bean development?

1. jsp:useBean
2. jsp:setProperty
3. jsp:getProperty

More details.

35) How to disable session in JSP?

```
1.          <%@ page session="false" %>
```

36) List the various action tags used in JSP.

Following are the list of various action tags used in JSP:

- jsp:forward: This action tag forwards the request and response to another resource.
- jsp:include: This action tag is used to include another resource.
- jsp:useBean: This action tag is used to create and locates bean object.
- jsp:setProperty: This action tag is used to set the value of the property of the bean.
- jsp:getProperty: This action tag is used to print the value of the property of the bean.
- jsp:plugin: This action tag is used to embed another component such as the applet.

- jsp:param: This action tag is used to set the parameter value. It is used in forward and includes mostly.
- jsp:fallback: This action tag can be used to print the message if the plugin is working.

37) Explain the steps for creating custom tags in JSP?

For creating any custom tag, we need to follow the following steps:

1. **Create the Tag handler class**
   To generate the Tag Handler, we are inheriting the TagSupport class and overriding its method doStartTag().To write data for the JSP, we need to use the JspWriter class.
   The PageContext class provides getOut() method that returns the instance of JspWriter class. TagSupport class provides an instance of pageContext by default.
2. **Create the TLD file**
   Tag Library Descriptor (TLD) file contains information of tag and Tag Hander classes. It must hold inside the WEB-INF directory.
3. **Create the JSP file**
   Let's use the tag in our JSP file. Here, we are specifying the path of tld file directly. However, it is recommended to use the URI name instead of full a path of tld file. We will learn about URI later. It uses taglib directive to use the tags defined in the tld file.

38) How can the applets be displayed in the JSP? Explain with an example.

The jsp:plugin action tag is used to embed an applet in the JSP file. The jsp:plugin action tag downloads plugin at client side to execute an applet or bean.

Example of displaying applet in JSP

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Mouse Drag</title>
  </head>
  <body bgcolor="khaki">
<h1>Mouse Drag Example</h1>

 <jsp:plugin align="middle" height="500" width="500"
    type="applet"  code="MouseDrag.class" name="clock" codebase="."/>

  </body>
</html>
```

39) What is Expression language in JSP?

The Expression Language (EL) simplifies the accessibility of data stored in the Java Bean component, and other objects like request, session, application.

There are many implicit objects, operators and reserve words in EL.

It is the newly added feature in JSP technology version 2.0.

40) Explain various implicit objects in expression language.

Implicit Objects	Usage
pageScope	it maps the given attribute name with the value set in the page scope
requestScope	it maps the given attribute name with the value set in the request scope
sessionScope	it maps the given attribute name with the value set in the session scope
applicationScope	it maps the given attribute name with the value set in the application scope

param	it maps the request parameter to the single value
paramValues	it maps the request parameter to an array of values
header	it maps the request header name to the single value
headerValues	it maps the request header name to an array of values
cookie	it maps the given cookie name to the cookie value
initParam	it maps the initialization parameter
pageContext	it provides access to many objects request, session, etc.

Spring Interview Questions

Spring interview questions and answers are frequently asked because it is now widely used framework to develop enterprise application in java. There are given a list of top 40 frequently asked spring interview questions.

1) What is Spring?

It is a lightweight, loosely coupled and integrated framework for developing enterprise applications in java.

2) What are the advantages of spring framework?

1. Predefined Templates

2. Loose Coupling
3. Easy to test
4. Lightweight
5. Fast Development
6. Powerful Abstraction
7. Declarative support

More details...

3) What are the modules of spring framework?

1. Test
2. Spring Core Container
3. AOP, Aspects and Instrumentation
4. Data Access/Integration
5. Web

More details...

4) What is IOC and DI?

IOC (Inversion of Control) and DI (Dependency Injection) is a design pattern to provide loose coupling. It removes the dependency from the program.

Let's write a code without following IOC and DI.

```
public class Employee{
Address address;
Employee(){
address=new Address();//creating instance
}
}
```

Now, there is dependency between Employee and Address because Employee is forced to use the same address instance.

Let's write the IOC or DI code.

```
public class Employee{
Address address;
Employee(Address address){
this.address=address;//not creating instance
}
}
```

Now, there is no dependency between Employee and Address because Employee is not forced to use the same address instance. It can use any address instance.

5) What is the role of IOC container in spring?

IOC container is responsible to:

- create the instance
- configure the instance, and
- assemble the dependencies

More details...

6) What are the types of IOC container in spring?

There are two types of IOC containers in spring framework.

1. BeanFactory
2. ApplicationContext

More details...

7) What is the difference between BeanFactory and ApplicationContext?

BeanFactory is the **basic container** whereas ApplicationContext is the **advanced container**. ApplicationContext extends the BeanFactory interface. ApplicationContext provides more facilities than BeanFactory such as integration with spring AOP, message resource handling for i18n etc.

8) What is the difference between constructor injection and setter injection?

No.	Constructor Injection	Setter Injection
1)	No Partial Injection	Partial Injection
2)	Desn't override the setter property	Overrides the constructor property if bot defined.
3)	Creates new instance if any modification occurs	Doesn't create new instance if you chang property value
4)	Better for too many properties	Better for few properties.

9) What is autowiring in spring? What are the autowiring modes?

Autowiring enables the programmer to inject the bean automatically. We don't need to write explicit injection logic.

Let's see the code to inject bean using dependency injection.

```
<bean id="emp" class="com.javatpoint.Employee" autowire="byName" />
```

The autowiring modes are given below:

No.	Mode	Description
1)	no	this is the default mode, it means autowiring is not enabled.
2)	byName	injects the bean based on the property name. It uses setter method.
3)	byType	injects the bean based on the property type. It uses setter method.
4)	constructor	It injects the bean using constructor

The "autodetect" mode is deprecated since spring 3.

10) What are the different bean scopes in spring?

There are 5 bean scopes in spring framework.

No.	Scope	Description
1)	singleton	The bean instance will be only once and same instance will be returned IOC container. It is the default scope.
2)	prototype	The bean instance will be created each time when requested.
3)	request	The bean instance will be created per HTTP request.
4)	session	The bean instance will be created per HTTP session.
5)	globalsession	The bean instance will be created per HTTP global session. It can be u portlet context only.

11) In which scenario, you will use singleton and prototype scope?

Singleton scope should be used with EJB **stateless session bean** and prototype scope with EJB **stateful session bean**.

12) What are the transaction management supports provided by spring?

Spring framework provides two type of transaction management supports:

1. **Programmatic Transaction Management**: should be used for few transaction operations.
2. **Declarative Transaction Management**: should be used for many transaction operations.

» Spring JDBC Interview Questions

13) What are the advantages of JdbcTemplate in spring?

Less code: By using the JdbcTemplate class, you don't need to create connection,statement,start transaction,commit transaction and close connection to execute different queries. You can execute the query directly.

More details...

14) What are classes for spring JDBC API?

1. JdbcTemplate
2. SimpleJdbcTemplate
3. NamedParameterJdbcTemplate
4. SimpleJdbcInsert
5. SimpleJdbcCall

More details...

15) How can you fetch records by spring JdbcTemplate?

You can fetch records from the database by the **query method of JdbcTemplate**. There are two interfaces to do this:

1. ResultSetExtractor
2. RowMapper

16) What is the advantage of NamedParameterJdbcTemplate?

NamedParameterJdbcTemplate class is used to pass value to the named parameter. A named parameter is better than ? (question mark of PreparedStatement).

It is **better to remember**.

17) What is the advantage of SimpleJdbcTemplate?

The **SimpleJdbcTemplate** supports the feature of var-args and autoboxing.

» Spring AOP Interview Questions

18) What is AOP?

AOP is an acronym for Aspect Oriented Programming. It is a methodology that divides the program logic into pieces or parts or concerns.

It increases the modularity and the key unit is Aspect.

More details...

19) What are the advantages of spring AOP?

AOP enables you to dynamically add or remove concern before or after the business logic. It is **pluggable** and **easy to maintain**.

More details...

20) What are the AOP terminology?

AOP terminologies or concepts are as follows:

- JoinPoint
- Advice
- Pointcut
- Aspect
- Introduction
- Target Object
- Interceptor
- AOP Proxy
- Weaving

More details...

21) What is JoinPoint?

JoinPoint is any point in your program such as field access, method execution, exception handling etc.

22) Does spring framework support all JoinPoints?

No, spring framework supports method execution joinpoint only.

23) What is Advice?

Advice represents action taken by aspect.

24) What are the types of advice in AOP?

There are 5 types of advices in spring AOP.

1. Before Advice
2. After Advice

3. After Returning Advice
4. Throws Advice
5. Around Advice

25) What is Pointcut?

Pointcut is expression language of Spring AOP.

26) What is Aspect?

Aspect is a class in spring AOP that contains advices and joinpoints.

27) What is Introduction?

Introduction represents introduction of new fields and methods for a type.

28) What is target object?

Target Object is a proxy object that is advised by one or more aspects.

29) What is interceptor?

Interceptor is a class like aspect that contains one advice only.

30) What is weaving?

Weaving is a process of linking aspect with other application.

31) Does spring perform weaving at compile time?

No, spring framework performs weaving at runtime.

32) What are the AOP implementation?

There are 3 AOP implementation.

1. Spring AOP
2. Apache AspectJ
3. JBoss AOP

» Spring MVC Interview Questions

33) What is the front controller class of Spring MVC?

The **DispatcherServlet** class works as the front controller in Spring MVC.

34) What does @Controller annotation?

The **@Controller** annotation marks the class as controller class. It is applied on the class.

35) What does @RequestMapping annotation?

The **@RequestMapping** annotation maps the request with the method. It is applied on the method.

36) What does the ViewResolver class?

The **View Resolver** class resolves the view component to be invoked for the request. It defines prefix and suffix properties to resolve the view component.

37) Which ViewResolver class is widely used?

The **org.springframework.web.servlet.view.InternalResourceViewResolver** class is widely used.

38) Does spring MVC provide validation support?

Yes.

Hibernate Interview Questions

Hibernate interview questions are asked to the students because it is a widely used ORM tool. The important list of top 20 hibernate interview questions and answers for freshers and professionals are given below.

1) What is hibernate?

Hibernate is an open-source and lightweight ORM tool that is used to store, manipulate, and retrieve data from the database.

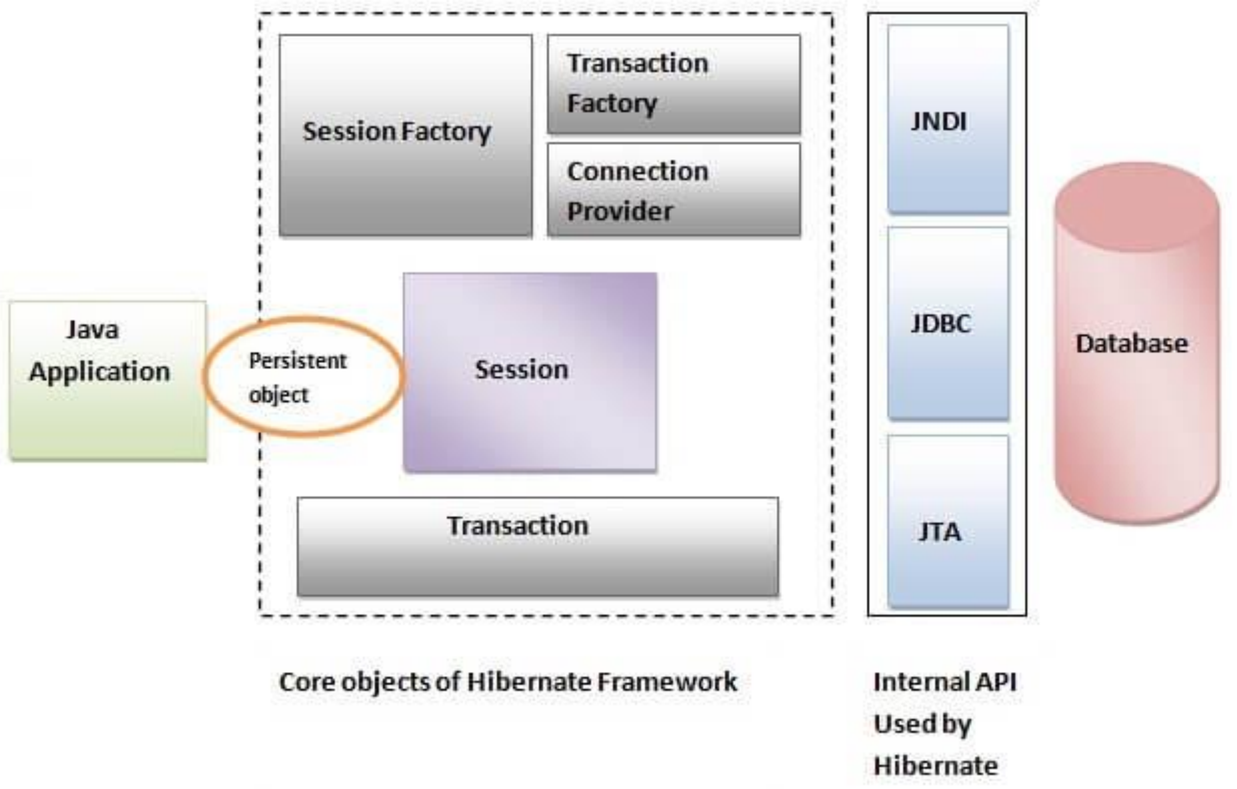
2) What is ORM?

ORM is an acronym for Object/Relational mapping. It is a programming strategy to map object with the data stored in the database. It simplifies data creation, data manipulation, and data access.

3) Explain hibernate architecture?

Hibernate architecture comprises of many interfaces such as Configuration, SessionFactory, Session, Transaction, etc.

more details...



4) What are the core interfaces of Hibernate?

The core interfaces of Hibernate framework are:

- Configuration
- SessionFactory
- Session

- Query
- Criteria
- Transaction

5) Mention some of the advantages of using ORM over JDBC.

ORM has the following advantages over JDBC:

- Application development is fast.
- Management of transaction.
- Generates key automatically.
- Details of SQL queries are hidden.

6) Define criteria in terms of Hibernate.

The objects of criteria are used for the creation and execution of the object-oriented criteria queries.

7) List some of the databases supported by Hibernate.

Some of the databases supported by Hibernate are:

- DB2
- MySQL
- Oracle
- Sybase SQL Server
- Informix Dynamic Server
- HSQL
- PostgreSQL
- FrontBase

8) List the key components of Hibernate.

Key components of Hibernate are:

- Configuration
- Session
- SessionFactory
- Criteria
- Query
- Transaction

9) Mention two components of Hibernate configuration object.

Database Connection

Class Mapping Setup

10) How is SQL query created in Hibernate?

The SQL query is created with the help of the following syntax:

Session.createSQLQuery

11) What does HQL stand for?

Hibernate Query Language

12) How is HQL query created?

The HQL query is created with the help of the following syntax:

Session.createQuery

13) How can we add criteria to a SQL query?

A criterion is added to a SQL query by using the Session.createCriteria.

14) Define persistent classes.

Classes whose objects are stored in a database table are called as persistent classes.

15) What is SessionFactory?

SessionFactory provides the instance of Session. It is a factory of Session. It holds the data of second level cache that is not enabled by default.

more details...

16) Is SessionFactory a thread-safe object?

Yes, SessionFactory is a thread-safe object, many threads cannot access it simultaneously.

17) What is Session?

It maintains a connection between the hibernate application and database.

It provides methods to store, update, delete or fetch data from the database such as persist(), update(), delete(), load(), get() etc.

It is a factory of Query, Criteria and Transaction i.e. it provides factory methods to return these instances.

more details...

18) Is Session a thread-safe object?

No, Session is not a thread-safe object, many threads can access it simultaneously. In other words, you can share it between threads.

19) What is the difference between session.save() and session.persist() method?

No.	save()	persist()
1)	returns the identifier (Serializable) of the instance.	Return nothing because its return type is void.
2)	Syn: public Serializable save(Object o)	Syn: public void persist(Object o)

20) What is the difference between get and load method?

No.	get()	load()
1)	Returns **null** if an object is not found.	Throws **ObjectNotFoundException** if an object is not found.
2)	get() method always **hit the database**.	load() method **doesn't hit** the database.
3)	It returns the real object, not the proxy.	It returns **proxy object.**
4)	It should be used if **you are not sure** about the existence of instance.	It should be used if **you are sure** that instance exists.

The differences between get() and load() methods are given below.

21) What is the difference between update and merge method?

No.	The update() method	merge() method
1)	Update means to edit something.	Merge means to combine something.
2)	update() should be used if the session doesn't contain an already persistent state with the same id. It means an update should be used inside the session only. After	merge() should be used if you don't know the state of the session, means you want to make the modification at

	closing the session, it will throw the error.	any time.

The differences between update() and merge() methods are given below.

Let's try to understand the difference by the example given below:

```
...
SessionFactory factory = cfg.buildSessionFactory();
Session session1 = factory.openSession();

Employee e1 = (Employee) session1.get(Employee.class, Integer.valueOf(101));//passing id of employee
session1.close();

e1.setSalary(70000);

Session session2 = factory.openSession();
Employee e2 = (Employee) session1.get(Employee.class, Integer.valueOf(101));//passing same id

Transaction tx=session2.beginTransaction();
session2.merge(e1);

tx.commit();
session2.close();
```

After closing session1, e1 is in detached state. It will not be in the session1 cache. So if you call update() method, it will throw an error.

Then, we opened another session and loaded the same Employee instance. If we call merge in session2, changes of e1 will be merged in e2.

22) What are the states of the object in hibernate?

There are 3 states of the object (instance) in hibernate.

1. **Transient**: The object is in a transient state if it is just created but has no primary key (identifier) and not associated with a session.
2. **Persistent**: The object is in a persistent state if a session is open, and you just saved the instance in the database or retrieved the instance from the database.
3. **Detached**: The object is in a detached state if a session is closed. After detached state, the object comes to persistent state if you call lock() or update() method.

23) What are the inheritance mapping strategies?

There are 3 ways of inheritance mapping in hibernate.

1. Table per hierarchy
2. Table per concrete class
3. Table per subclass

more details...

24) How to make an immutable class in hibernate?

If you mark a class as mutable="false", the class will be treated as an immutable class. By default, it is mutable="true".

25) What is automatic dirty checking in hibernate?

The automatic dirty checking feature of Hibernate, calls update statement automatically on the objects that are modified in a transaction.

Let's understand it by the example given below:

```
...
SessionFactory factory = cfg.buildSessionFactory();
Session session1 = factory.openSession();
Transaction tx=session2.beginTransaction();
```

```

        Employee e1 = (Employee) session1.get(Employee.class, Integer.valueOf(101));

        e1.setSalary(70000);

        tx.commit();
        session1.close();
```

Here, after getting employee instance e1 and we are changing the state of e1.

After changing the state, we are committing the transaction. In such a case, the state will be updated automatically. This is known as dirty checking in hibernate.

26) How many types of association mapping are possible in hibernate?

There can be 4 types of association mapping in hibernate.

1. One to One
2. One to Many
3. Many to One
4. Many to Many

27) Is it possible to perform collection mapping with One-to-One and Many-to-One?

No, collection mapping can only be performed with One-to-Many and Many-to-Many.

28) What is lazy loading in hibernate?

Lazy loading in hibernate improves the performance. It loads the child objects on demand.

Since Hibernate 3, lazy loading is enabled by default, and you don't need to do lazy="true". It means not to load the child objects when the parent is loaded.

29) What is HQL (Hibernate Query Language)?

Hibernate Query Language is known as an object-oriented query language. It is like a structured query language (SQL).

The main advantage of HQL over SQL is:

1. You don't need to learn SQL
2. Database independent
3. Simple to write a query

30) What is the difference between first level cache and second level cache?

No.	First Level Cache	Second Level Cache
1)	First Level Cache is **associated with Session**.	Second Level Cache is asso with **SessionFactory**.
2)	It is **enabled** by default.	It is **not enabled** by default.

MySQL Interview Questions

A list of top frequently asked MySQL interview questions and answers are given below.

1) What is MySQL?

MySQL is a multithreaded, multi-user SQL database management system which has more than 11 million installations. It is the world's second most popular and widely-used open source database. It is interesting how MySQL name was given to this query language. The term My is coined by the name of the daughter of co-founder Michael Widenius's daughter, and SQL is the short form of Structured Query Language. Using MySQL is free of cost for the developer, but enterprises have to pay a license fee to Oracle.

Formerly MySQL was initially owned by a for-profit firm MySQL AB, then Sun Microsystems bought it, and then Oracle bought Sun Microsystems, so Oracle currently owns MySQL.

MySQL is an Oracle-supported Relational Database Management System (RDBMS) based on structured query language. MySQL supports a wide range of operating systems, most famous of those include Windows, Linux & UNIX. Although it is possible to develop a wide range of applications with MySQL, it is only used for web applications & online publishing. It is a fundamental part of an open-source enterprise known as Lamp.

What is the Lamp?

The Lamp is a platform used for web development. The Lamp uses Linux, Apache, MySQL, and PHP as an operating system, web server, database & object-oriented scripting language. And hence abbreviated as LAMP.

2) In which language MySQL has been written?

MySQL is written in C and C++, and its SQL parser is written in yacc.

3) What are the technical specifications of MySQL?

MySQL has the following technical specifications -

- Flexible structure
- High performance
- Manageable and easy to use

- Replication and high availability
- Security and storage management
- Drivers
- Graphical Tools
- MySQL Enterprise Monitor
- MySQL Enterprise Security
- JSON Support
- Replication & High-Availability
- Manageability and Ease of Use
- OLTP and Transactions
- Geo-Spatial Support

4) What is the difference between MySQL and SQL?

SQL is known as the standard query language. It is used to interact with the database like MySQL. MySQL is a database that stores various types of data and keeps it safe.

A PHP script is required to store and retrieve the values inside the database.

SQL is a computer language, whereas MySQL is a software or an application

SQL is used for the creation of database management systems whereas MySQL is used to enable data handling, storing, deleting and modifying data

5) 5. What is the difference between the database and the table?

There is a major difference between a database and a table. The differences are as follows:

- Tables are a way to represent the division of data in a database while the database is a collection of tables and data.

- Tables are used to group the data in relation to each other and create a dataset. This dataset will be used in the database. The data stored in the table in any form is a part of the database, but the reverse is not true.
- A database is a collection of organized data and features used to access them, whereas the table is a collection of rows and columns used to store the data.

6) Why do we use the MySQL database server?

First of all, the MYSQL server is free to use for developers and small enterprises.

MySQL server is open source.

MySQL's community is tremendous and supportive; hence any help regarding MySQL is resolved as soon as possible.

MySQL has very stable versions available, as MySQL has been in the market for a long time. All bugs arising in the previous builds have been continuously removed, and a very stable version is provided after every update.

The MySQL database server is very fast, reliable, and easy to use. You can easily use and modify the software. MySQL software can be downloaded free of cost from the internet.

7) What are the different tables present in MySQL?

There are many tables that remain present by default. But, MyISAM is the default database engine used in MySQL. There are five types of tables that are present:

- MyISAM
- Heap
- Merge
- INNO DB
- ISAM

8) How to install MySQL?

Installing MySQL on our system allows us to safely create, drop, and test web applications without affecting our live website's data. There are many ways to use MySQL on our system, but the best way is to install it manually. The manual installation allows us to learn more about the system and provides more control over the database. To see the installation steps of MySQL in Windows goes to the below link:

https://www.javatpoint.com/how-to-install-mysql

Manual installation of MySQL has several benefits:

- Backing up, reinstalling, or moving databases from one location to another can be achieved in a second.
- It provides more control to how and when MySQL server starts and closes.
- We can install MySQL anywhere, like in a portable USB drive.

9) How to check the MySQL version?

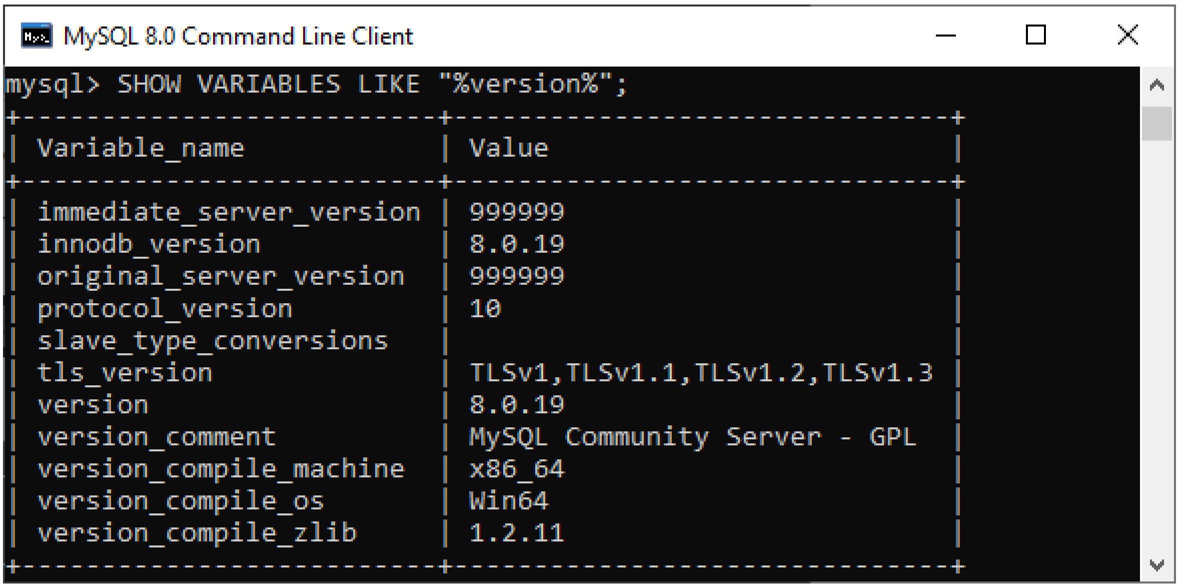
We can check the MySQL version on Linux using the below command:

```
1.    mysql -v
```

If we use the MySQL in windows, opening the MySQL command-line tool displayed the version information without using any flags. If we want to know more about the server information, use the below statement:

```
1.    SHOW VARIABLES LIKE "%version%";
```

It will return the output as below:



```
mysql> SHOW VARIABLES LIKE "%version%";
+---------------------------+------------------------------+
| Variable_name             | Value                        |
+---------------------------+------------------------------+
| immediate_server_version  | 999999                       |
| innodb_version            | 8.0.19                       |
| original_server_version   | 999999                       |
| protocol_version          | 10                           |
| slave_type_conversions    |                              |
| tls_version               | TLSv1,TLSv1.1,TLSv1.2,TLSv1.3 |
| version                   | 8.0.19                       |
| version_comment           | MySQL Community Server - GPL |
| version_compile_machine   | x86_64                       |
| version_compile_os        | Win64                        |
| version_compile_zlib      | 1.2.11                       |
+---------------------------+------------------------------+
```

In this output, we can see the additional version information about the installed MySQL software like innodb_version, protocol_version, version_ssl_library, etc.

10) How to add columns in MySQL?

A column is a series of cells in a table that stores one value for each row in a table. We can add columns in an existing table using the ALTER TABLE statement as follows:

1. **ALTER TABLE** table_name
2. **ADD COLUMN** column_name column_definition [**FIRST**|**AFTER** existing_column];

To read more information, click here.

11) How to delete a table in MySQL?

We can delete a table in MySQL using the Drop Table statement. This statement removes the complete data of a table, including structure and definition from the database permanently. Therefore, it is required to be careful while deleting a table. After using the statement, we cannot recover the table in MySQL. The statement is as follows:

1. **DROP TABLE** table_name;

To read more information, click here.

12) How to add foreign keys in MySQL?

The foreign key is used to link one or more tables together. It matches the primary key field of another table to link the two tables. It allows us to create a parent-child relationship with the tables. We can add a foreign key to a table in two ways:

- Using the CREATE TABLE Statement
- Using the ALTER TABLE Statement

Following is the syntax to define a foreign key using CREATE TABLE OR ALTER TABLE statement:

```
[CONSTRAINT constraint_name]
   FOREIGN KEY [foreign_key_name] (col_name, ...)
   REFERENCES parent_tbl_name (col_name,...)
```

To read more information, click here.

13) How to connect to the MySQL database?

MySQL allows us to connect with the database server in mainly two ways:

Using Command-line Tool

We can find the command-line client tool in the bin directory of the MySQL's installation folder. To invoke this program, we need to navigate the installation folder's bin directory and type the below command:

```
mysql
```

Next, we need to run the below command to connect to the MySQL Server:

```
shell>mysql -u root -p
```

Finally, type the password for the selected user account root and press Enter:

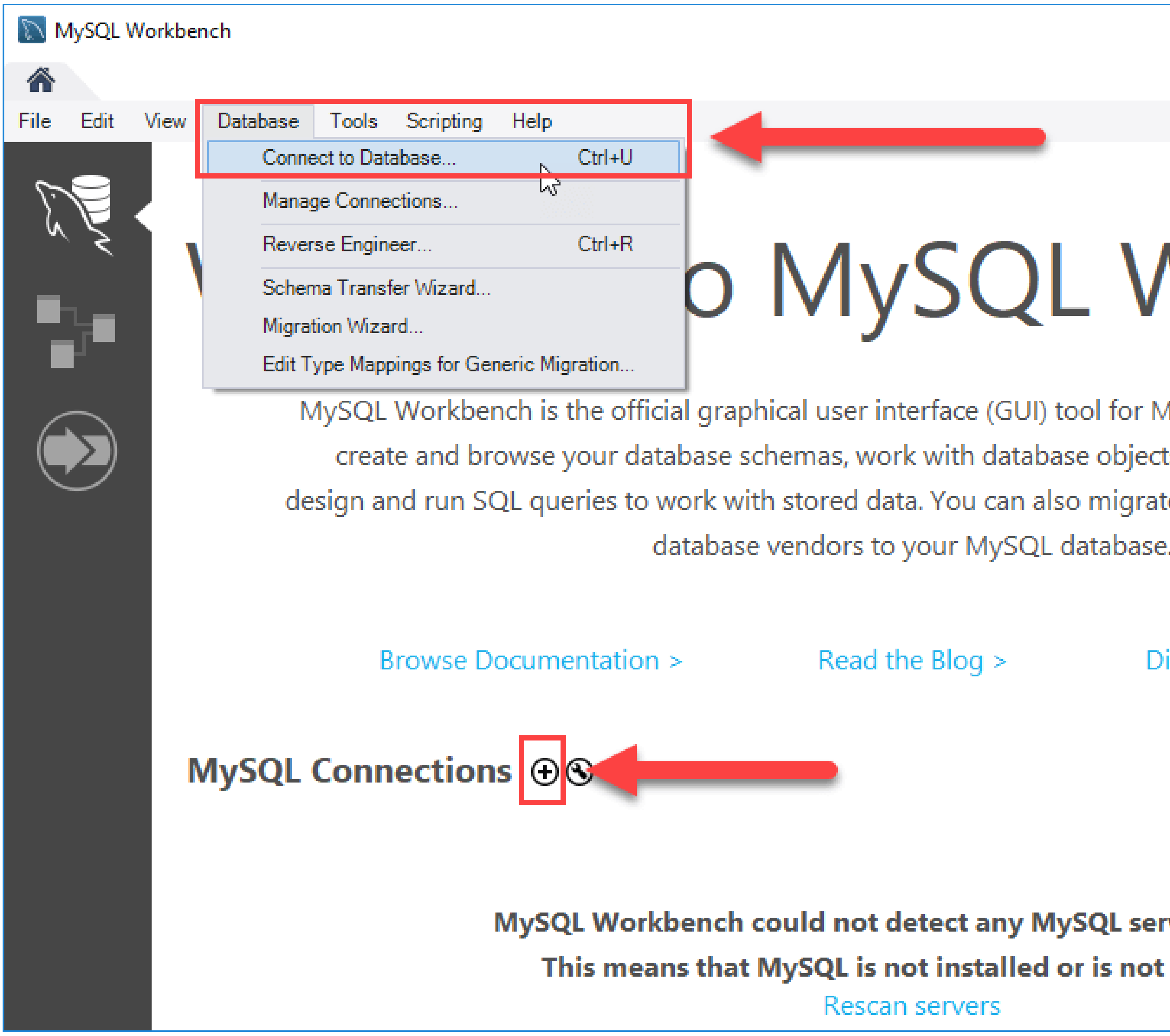
1. Enter **password**: ********

After successful connection, we can use the below command to use the:

1. USE database_name;

Using MySQL Workbench

We can make a connection with database using MySQL Workbench, simply clicking the plus (+) icon or navigating to the menu bar -> Database -> Connect to Database, the following screen appears. Now, you need to fill all the details to make a connection:



MySQL Workbench is the official graphical user interface (GUI) tool for M
create and browse your database schemas, work with database object
design and run SQL queries to work with stored data. You can also migrat
database vendors to your MySQL database.

MySQL Workbench could not detect any MySQL ser
This means that MySQL is not installed or is not

Once we finished this setup, it will open the MySQL Workbench screen. Now, we can double click on the newly created connection to connect with the database server.

To read more information, click here.

14) How to change the MySQL password?

We can change the MySQL root password using the below statement in the new notepad file and save it with an appropriate name:

```
1.          ALTER USER 'root'@'localhost' IDENTIFIED BY 'NewPassword';
```

Next, open a Command Prompt and navigate to the MySQL directory. Now, copy the following folder and paste it in our DOS command and press the Enter key.

```
1.          C:\Users\javatpoint> CD C:\Program Files\MySQL\MySQL Server 8.0\bin
```

Next, enter this statement to change the password:

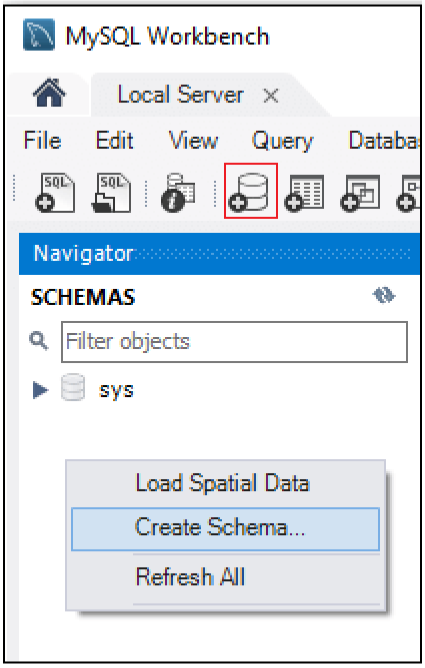
```
1.          mysqld --init-file=C:\\mysql-notepadfile.txt
```

Finally, we can log into the MySQL server as root using this new password. After launches the MySQL server, it is to delete the C:\myswl-init.txt file to ensure the password change.

To read more information, click here.

15) How to create a database in MySQL Workbench?

To create a new database in MySQL Workbench, we first need to launch the MySQL Workbench and log in using the username and password. Go to the Navigation tab and click on the Schema menu. Right-click under the Schema menu and select Create Schema or click the database icon (red rectangle), as shown in the following screen.

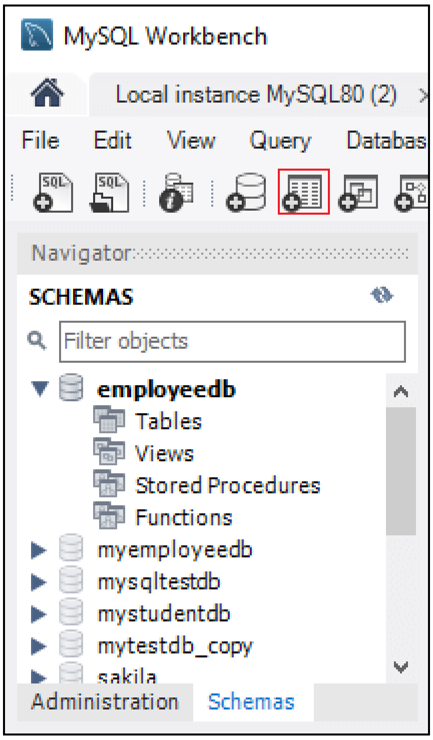


A new popup screen appears where we need to fill all the details. After entering the details, click on the Apply button and then the Finish button to complete the database creation.

To read more information, click here.

16) How to create a table in MySQL Workbench?

Launch the MySQL Workbench and go to the Navigation tab and click on the Schema menu where all the previously created databases are shown. Select any database and double click on it. It will show the sub-menus where we need to select the Tables option.



Select Tables sub-menu, right-click on it and select Create Table option. We can also click on create a new table icon (shown in red rectangle) to create a table. It will open the new popup screen where we need to fill all the details to create a table. Here, we will enter the table name and column details. After entering the details, click on the Apply button and then the Finish button to complete the table creation.

To read more information, click here.

17) How to change the table name in MySQL?

Sometimes our table name is non-meaningful. In that case, we need to change or rename the table name. MySQL provides the following syntax to rename one or more tables in the current database:

```
1.          mysql> RENAME old_table TO new_table;
```

If we want to change more than one table name, use the below syntax:

```
1.          RENAME TABLE old_tab1 TO new_tab1,
                    old_tab2 TO new_tab2, old_tab3 TO new_tab3;
```

To read more information, click here.

18) How to change the database name in MySQL?

Sometimes we need to change or rename the database name because of its non-meaningful name. To rename the database name, we need first to create a new database into the MySQL server. Next, MySQL provides the mysqldump shell command to create a dumped copy of the selected database and then import all the data into the newly created database. The following is the syntax of using mysqldump command:

```
mysqldump -u username -p "password" -R oldDbName > oldDbName.sql
```

Now, use the below command to import the data into the newly created database:

```
mysql -u username -p"password" newDbName < oldDbName.sql
```

19) How to import a database in MySQL?

Importing database in MySQL is a process of moving data from one place to another place. It is a very useful method for backing up essential data or transferring our data between different locations. For example, we have a contact book database, which is essential to keep it in a secure place. So we need to export it in a safe place, and whenever it lost from the original location, we can restore it using import options.

In MySQL, we can import a database in mainly two ways:

- Command Line Tool
- MySQL Workbench

To read more information for importing databases, click here.

20) How to change the column name in MySQL?

While creating a table, we have kept one of the column names incorrectly. To change or rename an existing column name in MySQL, we need to use the ALTER TABLE and CHANGE commands together. The following are the syntax used to rename a column in MySQL:

```
ALTER TABLE table_name
    CHANGE COLUMN old_column_name new_column_name column_definition [FIRST|AFTER existing_column];
```

Suppose the column's current name is S_ID, but we want to change this with a more appropriate title as Stud_ID. We will use the below statement to change its name:

```
ALTER TABLE Student CHANGE COLUMN S_ID Stud_ID varchar(10);
```

21) How to delete columns in MySQL?

We can remove, drop, or delete one or more columns in an existing table using the ALTER TABLE statement as follows:

```
ALTER TABLE table_name DROP COLUMN column_name1, column_name2....;
```

To read more information, click here.

22) How to insert data in MySQL?

We can insert data in a MySQL table using the INSERT STATEMENT. This statement allows us to insert single or multiple rows into a table. The following is the basic syntax to insert a record into a table:

```
INSERT INTO table_name ( field1, field2,...fieldN )
VALUES  ( value1, value2,...valueN );
```

If we want to insert more than one rows into a table, use the below syntax:

```
INSERT INTO table(field1, field2,...fieldN)
VALUES
   (value1, value 2, ...),
   (value1, value2, ...),
    ...
   (value1, value2, ...);
```

To read more information, click here.

23) How to delete a row in MySQL?

We can delete a row from the MySQL table using the DELETE STATEMENT within the database. The following is the generic syntax of DELETE statement in MySQL to remove one or more rows from a table:

```
DELETE FROM table_name WHERE Condition_specified;
```

It is noted that if we have not specified the WHERE clause with the syntax, this statement will remove all the records from the given table.

To read more information, click here.

24) How to join two tables in MySQL?

We can connect two or more tables in MySQL using the JOIN clause. MySQL allows various types of JOIN clauses. These clauses connect multiple tables and return only those records that match the same value and property in all tables. The following are the four easy ways to join two or more tables in MySQL:

- Inner Join
- Left Join
- Right Join
- Cross Join

To read more information, click here.

25) How to join three tables in MySQL?

Sometimes we need to fetch data from three or more tables. There are two types available to do these types of joins. Suppose we have three tables named Student, Marks, and Details.

Let's say Student has (stud_id, name) columns, Marks has (school_id, stud_id, scores) columns, and Details has (school_id, address, email) columns.

1. Using SQL Join Clause

This approach is similar to the way we join two tables. The following query returns result from three tables:

```
SELECT name, scores, address, email FROM Student s
INNER JOIN Marks m on s.stud_id = m.stud_id
INNER JOIN Details d on d.school_id = m.school_id;
```

2. Using Parent-Child Relationship

It is another approach to join more than two tables. In the above tables, we have to create a parent-child relationship. First, create column X as a primary key in one table and as a foreign key in another table. Therefore, stud_id is the primary key in the Student table and will be a foreign key in the Marks table. Next, school_id is the primary key in the Marks table and will be a foreign key in the Details table. The following query returns result from three tables:

```
SELECT name, scores, address, email
FROM Student s, Marks m, Details d
WHERE s.stud_id = m.stud_id AND m.school_id = d.school_id;
```

To read more information about the foreign key, click here.

26) How to update the table in MySQL?

We can update existing records in a table using the UPDATE statement that comes with the SET and WHERE clauses. The SET clause changes the values of the specified column. The WHERE clause is optional, which is used to specify the condition. This statement can also use to change values in one or more columns of a single row or multiple rows at a time. Following is a generic syntax of UPDATE command to modify data into the MySQL table:

```
UPDATE table_name
SET field1=new-value1, field2=new-value2, ...
[WHERE Clause]
```

To read more information, click here.

27) What is MySQL Workbench?

MySQL Workbench is a unified visual database designing or GUI tool used for working on MySQL databases. It is developed and maintained by Oracle that provides SQL development, data migration, and comprehensive administration tools for server configuration, user administration, backup, etc. We can use this Server Administration to create new physical data models, E-R diagrams, and SQL development. It is available for all major operating systems. MySQL provides supports for it from MySQL Server version v5.6 and higher.

It is mainly available in three editions, which are given below:

- Community Edition (Open Source, GPL)

- Standard Edition (Commercial)
- Enterprise Edition (Commercial)

To read more information, click here.

28) How to drop the primary key in MySQL?

MySQL primary key is a single or combination of the field used to identify each record in a table uniquely. A primary key column cannot be null or empty. We can remove or delete a primary key from the table using the ALTER TABLE statement. The following syntax is used to drop the primary key:

```
ALTER TABLE table_name  DROP PRIMARY KEY;
```

To read more information, click here.

29) How to create a Stored Procedure in MySQL?

A stored procedure is a group of SQL statements that we save in the database. The SQL queries, including INSERT, UPDATE, DELETE, etc. can be a part of the stored procedure. A procedure allows us to use the same code over and over again by executing a single statement. It stores in the database data dictionary.

We can create a stored procedure using the below syntax:

```
CREATE PROCEDURE procedure_name [ (parameter datatype [, parameter datatype]) ]
BEGIN
   Body_section of SQL statements
END;
```

This statement can return one or more value through parameters or may not return any result. The following example explains it more clearly:

```
DELIMITER $$
CREATE PROCEDURE get_student_info()
BEGIN
SELECT * FROM Student_table;
END$$
```

To read more information, click here.

30) How to execute a stored procedure in MySQL?

We can execute a stored procedure in MySQL by simply CALL query. This query takes the name of the stored procedure and any parameters we need to pass to it. The following is the basic syntax to execute a stored procedure:

```
CALL stored_procedure_name (argument_list);
```

Let's understand it with this example:

```
CALL Product_Pricing (@pricelow, @pricehigh);
```

Here, a stored procedure named Product_Pricing calculates and returns the lowest and highest product prices.

31) How to create a View in MySQL?

A view is a database object whose values are based on the base table. It is a **virtual table** created by a query by joining one or more tables. It is operated similarly to the base table but does not contain any data of its own. If any changes occur in the underlying table, the same changes reflected in the View also.

Following is the general syntax of creating a VIEW in MySQL:

```
CREATE [OR REPLACE] VIEW view_name AS
SELECT columns
FROM tables
[WHERE conditions];
```

To read more information, click here.

32) How to create a Trigger in MySQL?

A trigger is a procedural code in a database that automatically invokes whenever certain events on a particular table or view in the database occur. It can be executed when records are inserted into a table, or any columns are being updated. We can create a trigger in MySQL using the syntax as follows:

```
CREATE TRIGGER trigger_name
   [before | after]
  {insert | update | delete}
   ON table_name [FOR EACH ROW]
   BEGIN
      --variable declarations
      --trigger code
   END;

```

33) How to clear screen in MySQL?

If we use MySQL in Windows, it is not possible to clear the screen before version 8. At that time, the Windows operating system provides the only way to clear the screen by exiting the MySQL command-line tool and then again open MySQL.

After the release of MySQL version 8, we can use the below command to clear the command line screen:

```
1.          mysql> SYSTEM CLS;
```

34) How to create a new user in MySQL?

A USER in MySQL is a record in the USER-TABLE. It contains the login information, account privileges, and the host information for MySQL account to access and manage the databases. We can create a new user account in the database server using the MySQL Create User statement. It provides authentication, SSL/TLS, resource-limit, role, and password management properties for the new accounts.

The following is the basic syntax to create a new user in MySQL:

```
1.          CREATE USER [IF NOT EXISTS] account_name IDENTIFIED BY 'password
';
```

To read more information, click here.

35) How to check USERS in MySQL?

If we want to manage a database in MySQL, it is required to see the list of all user's accounts in a database server. The following command is used to check the list of all users available in the database server:

```
1.          mysql> SELECT USER FROM mysql.user;
```

To read more information, click here.

36) How to import a CSV file in MySQL?

MySQL allows us to import the CSV (comma separated values) file into a database or table. A CSV is a plain text file that contains the list of data and can be saved in a tabular format. MySQL provides the LOAD DATA INFILE statement to import a CSV file. This statement is used to read a text file and import it into a database table very quickly. The full syntax to import a CSV file is given below:

```
LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/filename.csv'
INTO TABLE tablename
FIELDS TERMINATED BY ','
OPTIONALLY ENCLOSED BY '"'
LINES TERMINATED BY '\r\n'
IGNORE 1 ROWS;
```

To read more information, click here.

37) How to insert Date in MySQL?

MySQL allows us to use the INSERT STATEMENT to add the date in MySQL table. MySQL provides several data types for storing dates such as DATE, TIMESTAMP, DATETIME, and YEAR. The default format of the date in MySQL is YYYY-MM-DD. Following is the basic syntax to insert date in MySQL table:

```
INSERT INTO table_name (column_name, column_date) VALUES ('DATE: Manual Date', '2008-7-04');
```

If we want to insert a date in the mm/dd/yyyy format, it is required to use the below statement:

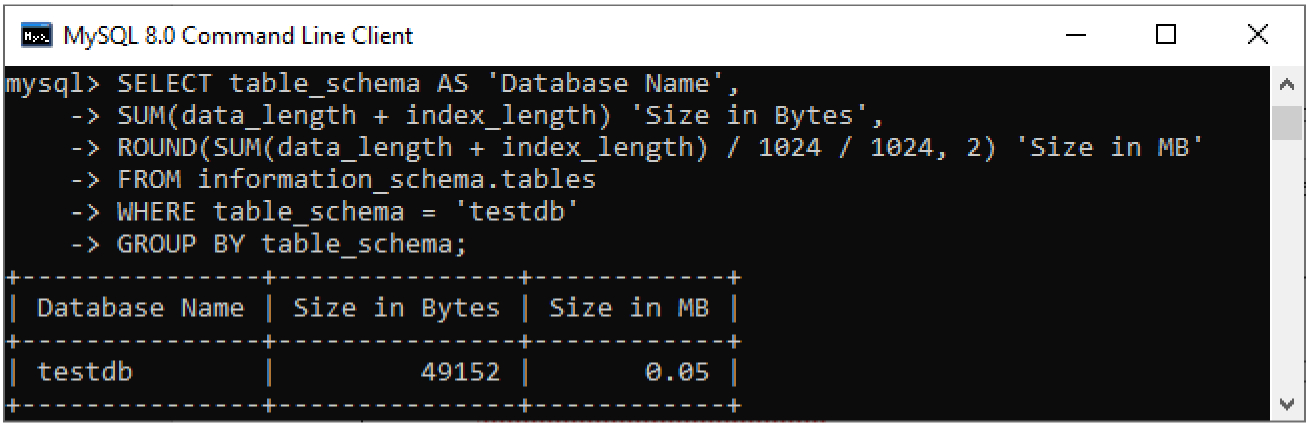
```
INSERT INTO table_name VALUES (STR_TO_DATE(date_value, format_specifier));
```

38) How to check database size in MySQL?

MySQL allows us to query the information_schema.tables table to get the information about the tables and databases. It will return the information about the data length, index length, collation, creation time, etc. We can check the size of the database on the server using the below syntax:

```
SELECT table_schema AS 'Database Name',
SUM(data_length + index_length) 'Size in Bytes',
ROUND(SUM(data_length + index_length) / 1024 / 1024, 2) 'Size in MB'

FROM information_schema.tables
WHERE table_schema = 'testdb'
GROUP BY table_schema;
```

It will return the output as follows:

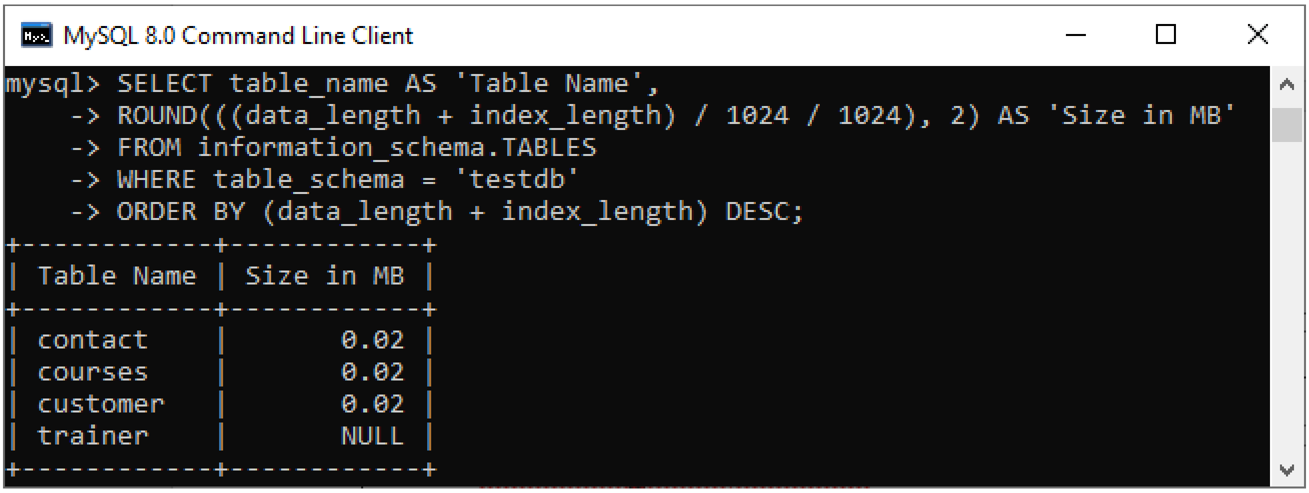


```
mysql> SELECT table_schema AS 'Database Name',
    -> SUM(data_length + index_length) 'Size in Bytes',
    -> ROUND(SUM(data_length + index_length) / 1024 / 1024, 2) 'Size in MB'
    -> FROM information_schema.tables
    -> WHERE table_schema = 'testdb'
    -> GROUP BY table_schema;
+---------------+---------------+------------+
| Database Name | Size in Bytes | Size in MB |
+---------------+---------------+------------+
| testdb        |         49152 |       0.05 |
+---------------+---------------+------------+
```

If we want to check the size of the table in a specific database, use the following statement:

```
SELECT table_name AS 'Table Name',
ROUND(((data_length + index_length) / 1024 / 1024), 2) AS 'Size in MB'

FROM information_schema.TABLES
WHERE table_schema = 'testdb'
ORDER BY (data_length + index_length) DESC;
```

It will return the output as follows:



```
mysql> SELECT table_name AS 'Table Name',
    -> ROUND(((data_length + index_length) / 1024 / 1024), 2) AS 'Size in MB'
    -> FROM information_schema.TABLES
    -> WHERE table_schema = 'testdb'
    -> ORDER BY (data_length + index_length) DESC;
+------------+------------+
| Table Name | Size in MB |
+------------+------------+
| contact    |       0.02 |
| courses    |       0.02 |
| customer   |       0.02 |
| trainer    |       NULL |
+------------+------------+
```

39) How does indexing works in MySQL?

Indexing is a process to find an unordered list into an ordered list. It helps in maximizing the query's efficiency while searching on tables in MySQL. The working of MySQL indexing is similar to the book index.

Suppose we have a book and want to get information about, say, searching. Without indexing, it is required to go through all pages one by one, until the specific topic was not found. On the other hand, an index contains a list of keywords to find the topic mentioned on pages. Then, we can flip to those pages directly without going through all pages.

40) Who owns MySQL?

MySQL is the most popular free and open-source database software which comes under the GNU General Public License. In the beginning, it was owned and sponsored by the Swedish company MySQL AB. Now, it is bought by Sun Microsystems (now Oracle Corporation), who is responsible for managing and developing the database.

To read more information, click here.

41) How to view the database in MySQL?

Working with the MySQL server, it is a common task to view or list the available databases. We can view all the databases on the MySQL server host using the following command:

```
mysql> SHOW DATABASES;
```

To read more information, click here.

42) How to set auto increment in MySQL?

Auto Increment is a constraint that automatically generates a unique number while inserting a new record into the table. Generally, it is used for the primary key field in a table. In MySQL, we can set the value for an AUTO_INCREMENT column using the ALTER TABLE statement as follows:

```
ALTER TABLE table_name AUTO_INCREMENT = value;
```

43) How to find the second highest salary in MySQL?

MySQL uses the LIMIT keyword, which can be used to limit the result set. It will allow us to get the first few rows, last few rows, or range of rows. It can also be used to find the second, third, or nth highest salary. It ensures that you have use order by clause to sort the result set first and then print the output that provides accurate results. The following query is used to get the second highest salary in MySQL:

```
SELECT salary
FROM (SELECT salary FROM employees ORDER BY salary DESC LIMIT 2) AS Emp ORDER BY salary LIMIT 1;
```

There are some other ways to find the second highest salary in MySQL, which are given below:

This statement uses subquery and IN clause to get the second highest salary:

```
SELECT MAX(salary)
FROM employees
WHERE salary NOT IN ( SELECT Max(salary) FROM employees);
```

This query uses subquery and < operator to return the second highest salary:

```
SELECT MAX(salary) From employees
WHERE salary < ( SELECT Max(salary) FROM employees);
```

44) What is the difference between TRUNCATE and DELETE in MySQL?

- TRUNCATE is a DDL command, and DELETE is a DML command.
- It is not possible to use Where command with TRUNCATE QLbut you can use it with DELETE command.
- TRUNCATE cannot be used with indexed views, whereas DELETE can be used with indexed views.
- The DELETE command is used to delete data from a table. It only deletes the rows of data from the table while truncate is a very dangerous command and should be used carefully because it deletes every row permanently from a table.

45) How many Triggers are possible in MySQL?

There are only six Triggers allowed to use in the MySQL database.

1. Before Insert
2. After Insert
3. Before Update
4. After Update
5. Before Delete
6. After Delete

46) What is the heap table?

Tables that are present in memory is known as HEAP tables. When you create a heap table in MySQL, you should need to specify the TYPE as HEAP. These tables are commonly known as memory tables. They are used for high-speed storage on a temporary basis. They do not allow BLOB or TEXT fields.

47) What is BLOB and TEXT in MySQL?

BLOB is an acronym that stands for a large binary object. It is used to hold a variable amount of data.

There are four types of the BLOB.

1. TINYBLOB
2. BLOB
3. MEDIUMBLOB
4. LONGBLOB

The differences among all these are the maximum length of values they can hold.

TEXT is a case-insensitive BLOB. TEXT values are non-binary strings (character string). They have a character set, and values are stored and compared based on the collation of the character set.

There are four types of TEXT.

1. TINYTEXT
2. TEXT
3. MEDIUMTEXT
4. LONGTEXT

48) What is a trigger in MySQL?

A trigger is a set of codes that executes in response to some events.

49) What is the difference between the heap table and the temporary table?

Heap tables:

Heap tables are found in memory that is used for high-speed storage temporarily. They do not allow BLOB or TEXT fields.

Heap tables do not support AUTO_INCREMENT.

Indexes should be NOT NULL.

Temporary tables:

The temporary tables are used to keep the transient data. Sometimes it is beneficial in cases to hold temporary data. The temporary table is deleted after the current client session terminates.

Main differences:

The heap tables are shared among clients, while temporary tables are not shared.

Heap tables are just another storage engine, while for temporary tables, you need a special privilege (create temporary table).

50) What is the difference between FLOAT and DOUBLE?

FLOAT stores floating-point numbers with accuracy up to 8 places and allocate 4 bytes. On the other hand, DOUBLE stores floating-point numbers with accuracy up to 18 places and allocates 8 bytes.

51) What are the advantages of MySQL in comparison to Oracle?

1. MySQL is a free, fast, reliable, open-source relational database while Oracle is expensive, although they have provided Oracle free edition to attract MySQL users.

2. MySQL uses only just under 1 MB of RAM on your laptop, while Oracle 9i installation uses 128 MB.
3. MySQL is great for database enabled websites while Oracle is made for enterprises.
4. MySQL is portable.

52) What are the disadvantages of MySQL?

1. MySQL is not so efficient for large scale databases.
2. It does not support COMMIT and STORED PROCEDURES functions version less than 5.0.
3. Transactions are not handled very efficiently.
4. The functionality of MySQL is highly dependent on other addons.
5. Development is not community-driven.

53) What is the difference between CHAR and VARCHAR?

1. CHAR and VARCHAR have differed in storage and retrieval.
2. CHAR column length is fixed, while VARCHAR length is variable.
3. The maximum no. of character CHAR data types can hold is 255 characters, while VARCHAR can hold up to 4000 characters.
4. CHAR is 50% faster than VARCHAR.
5. CHAR uses static memory allocation, while VARCHAR uses dynamic memory allocation.

54) What is the difference between MySQL_connect and MySQL_pconnect?

Mysql_connect:

1. It opens a new connection to the database.

2. Every time you need to open and close the database connection, depending on the request.
3. Opens page whenever it is loaded.

Mysql_pconnect:

1. In Mysql_pconnect, "p" stands for persistent connection, so it opens the persistent connection.
2. The database connection cannot be closed.
3. It is more useful if your site has more traffic because there is no need to open and close connection frequently and whenever the page is loaded.

55) What does "i_am_a_dummy flag" do in MySQL?

The "i_am_a_dummy flag" enables the MySQL engine to refuse any UPDATE or DELETE statement to execute if the WHERE clause is not present. Hence it can save the programmer from deleting the entire table my mistake if he does not use WHERE clause.

56) How to get the current date in MySQL?

To get current date, use the following syntax:

```
SELECT CURRENT_DATE();
```

57) What are the security alerts while using MySQL?

Install antivirus and configure the operating system's firewall.

Never use the MySQL Server as the UNIX root user.

Change the root username and password Restrict or disable remote access.

58) How to change a password for an existing user via mysqladmin?

Mysqladmin -u root -p password "newpassword".

59) What is the difference between UNIX timestamps and MySQL timestamps?

Actually, both Unix timestamp and MySQL timestamp are stored as 32-bit integers, but MySQL timestamp is represented in the readable format of YYYY-MM-DD HH:MM:SS format.

60) How to display the nth highest salary from a table in a MySQL query?

Let us take a table named the employee.

To find Nth highest salary is:

select distinct(salary)from employee order by salary desc limit n-1,1

if you want to find 3rd largest salary:

select distinct(salary)from employee order by salary desc limit 2,1

61) What is the MySQL default port number?

MySQL default port number is 3306.

62) What is REGEXP?

REGEXP is a pattern match using a regular expression. The regular expression is a powerful way of specifying a pattern for a sophisticated search.

Basically, it is a special text string for describing a search pattern. To understand it better, you can think of a situation of daily life when you search for .txt files to list all text files in the file manager. The regex equivalent for .txt will be .*\.txt.

63) How many columns can you create for an index?

You can a create maximum of 16 indexed columns for a standard table.

64) What is the difference between NOW() and CURRENT_DATE()?

NOW() command is used to show current year, month, date with hours, minutes, and seconds while CURRENT_DATE() shows the current year with month and date only.

65) What is the query to display the top 20 rows?

SELECT * FROM table_name LIMIT 0,20;

66) Write a query to display the current date and time?

If you want to display the current date and time, use -

SELECT NOW();

If you want to display the current date only, use:

SELECT CURRENT_DATE();

67) What is the save point in MySQL?

A defined point in any transaction is known as savepoint.

SAVEPOINT is a statement in MySQL, which is used to set a named transaction savepoint with the name of the identifier.

68) What is SQLyog?

SQLyog program is the most popular GUI tool for admin. It is the most popular MySQL manager and admin tool. It combines the features of MySQL administrator, phpMyadmin, and others. MySQL front ends and MySQL GUI tools.

69) How do you backup a database in MySQL?

It is easy to back up data with phpMyAdmin. Select the database you want to backup by clicking the database name in the left-hand navigation bar. Then click the export button and make sure that all tables are highlighted that you want to back up. Then specify the option you want under export and save the output.

70) What are the different column comparison operators in MySQL?

The =, <>, <=, <, >=, >, <<, >>, < = >, AND, OR or LIKE operator are the comparison operators in MySQL. These operators are generally used with SELECT statement.

71) Write a query to count the number of rows of a table in MySQL.

SELECT COUNT user_id FROM users;

72) Write a query to retrieve a hundred books starting from 20th.

SELECT book_title FROM books LIMIT 20, 100;

73) Write a query to select all teams that won either 1, 3, 5, or 7 games.

SELECT team_name FROM team WHERE team_won IN (1, 3, 5, 7);

74) What is the default port of MySQL Server?

The default port of MySQL Server is 3306.

75) How is the MyISAM table stored?

MyISAM table is stored on disk in three formats.

- '.frm' file : storing the table definition
- '.MYD' (MYData): data file
- '.MYI' (MYIndex): index file

76) What is the usage of ENUMs in MySQL?

ENUMs are string objects. By defining ENUMs, we allow the end-user to give correct input as in case the user provides an input that is not part of the ENUM defined data, then the query won't execute, and an error message will be displayed which says "The wrong Query". For instance, suppose we want to take the gender of the user as an input, so we specify ENUM('male', 'female', 'other'), and hence whenever the user tries to input any string any other than these three it results in an error.

ENUMs are used to limit the possible values that go in the table:

For example:

CREATE TABLE months (month ENUM 'January', 'February', 'March'); INSERT months VALUES ('April').

77) What are the advantages of MyISAM over InnoDB?

MyISAM follows a conservative approach to disk space management and stores each MyISAM table in a separate file, which can be further compressed if required. On the other hand, InnoDB stores the tables in the tablespace. Its further optimization is difficult.

78) What are the differences between MySQL_fetch_array(), MySQL_fetch_object(), MySQL_fetch_row()?

Mysql_fetch_object is used to retrieve the result from the database as objects, while mysql_fetch_array returns result as an array. This will allow access to the data by the field names.

For example:

Using mysql_fetch_object field can be accessed as $result->name.

Using mysql_fetch_array field can be accessed as $result->[name].

Using mysql_fetch_row($result) where $result is the result resource returned from a successful query executed using the mysql_query() function.

Example:

```
$result = mysql_query("SELECT * from students");
while($row = mysql_fetch_row($result))
{
    Some statement;
}
```

79) What is the difference between mysql_connect and mysql_pconnect?

Mysql_connect() is used to open a new connection to the database, while mysql_pconnect() is used to open a persistent connection to the database. It specifies that each time the page is loaded, mysql_pconnect() does not open the database.

80) What is the use of mysql_close()?

Mysql_close() cannot be used to close the persistent connection. However, it can be used to close a connection opened by mysql_connect().

81) What is MySQL data directory?

MySQL data directory is a place where MySQL stores its data. Each subdirectory under this data dictionary represents a MySQL database. By default, the information managed my MySQL = server mysqld is stored in the data directory.

82) How do you determine the location of MySQL data directory?

The default location of MySQL data directory in windows is C:\mysql\data or C:\Program Files\MySQL\MySQL Server 5.0 \data.

83) What is the usage of regular expressions in MySQL?

In MySQL, regular expressions are used in queries for searching a pattern in a string.

- * Matches 0 more instances of the string preceding it.
- + matches one more instances of the string preceding it.
- ? Matches 0 or 1 instances of the string preceding it.
- . Matches a single character.
- [abc] matches a or b or z
- | separates strings
- ^ anchors the match from the start.
- "." Can be used to match any single character. "|" can be used to match either of the two strings
- REGEXP can be used to match the input characters with the database.

Example:

The following statement retrieves all rows where column employee_name contains the text 1000 (example salary):

```
Select employee_name
From employee
Where employee_name REGEXP '1000'
Order by employee_name
```

84) What is the usage of the "i-am-a-dummy" flag in MySQL?

In MySQL, the "i-am-a-dummy" flag makes the MySQL engine to deny the UPDATE and DELETE commands unless the WHERE clause is present.

85) Which command is used to view the content of the table in MySQL?

The SELECT command is used to view the content of the table in MySQL.

Explain Access Control Lists.

An ACL is a list of permissions that are associated with an object. MySQL keeps the Access Control Lists cached in memory, and whenever the user tries to authenticate or execute a command, MySQL checks the permission required for the object, and if the permissions are available, then execution completes successfully.

86) What is InnoDB?

InnoDB is a storage database for SQL. The ACID-transactions are also provided in InnoDB and also includes support for the foreign key. Initially owned by InnobaseOY now belongs to Oracle Corporation after it acquired the latter since 2005.

87) What is ISAM?

It is a system for file management developed by IBM, which allows records to access sequentially or even randomly.

88) How can we run batch mode in MySQL?

To perform batch mode in MySQL, we use the following command:

```
mysql;
mysql mysql.out;
```

89) What are federated tables?

Federated tables are tables that point to the tables located on other databases on some other server.

90) What is the difference between primary key and candidate key?

To identify each row of a table, we will use a primary key. For a table, there exists only one primary key.

A candidate key is a column or a set of columns, which can be used to uniquely identify any record in the database without having to reference any other data.

91) What are the drivers in MySQL?

Following are the drivers available in MySQL:

- PHP Driver
- JDBC Driver
- ODBC Driver

- C WRAPPER
- PYTHON Driver
- PERL Driver
- RUBY Driver
- CAP11PHP Driver
- Ado.net5.mxz

92) What are DDL, DML, and DCL?

Majorly SQL commands can be divided into three categories, i.e., DDL, DML & DCL. Data Definition Language (DDL) deals with all the database schemas, and it defines how the data should reside in the database. Commands like CreateTABLE and ALTER TABLE are part of DDL.

Data Manipulative Language (DML) deals with operations and manipulations on the data. The commands in DML are Insert, Select, etc.

Data Control Languages (DCL) are related to the Grant and permissions. In short, the authorization to access any part of the database is defined by these.

Spring Boot Interview Questions

1) What is Spring Boot?

Spring Boot is a Spring module which provides RAD (Rapid Application Development) feature to Spring framework.

It is used to create stand alone spring based application that you can just run because it needs very little spring configuration.

For more information click here.

2) What are the advantages of Spring Boot?

- Create stand-alone Spring applications that can be started using java -jar.
- Embed Tomcat, Jetty or Undertow directly. You don't need to deploy WAR files.
- It provides opinionated 'starter' POMs to simplify your Maven configuration.
- It automatically configure Spring whenever possible.

3) What are the features of Spring Boot?

- Web Development
- SpringApplication
- Application events and listeners
- Admin features

For more information click here.

4) How to create Spring Boot application using Maven?

There are multiple approaches to create Spring Boot project. We can use any of the following approach to create application.

- Spring Maven Project
- Spring Starter Project Wizard
- Spring Initializr
- Spring Boot CLI

For more information click here.

5) How to create Spring Boot project using Spring Initializer?

It is a web tool which is provided by Spring on official site. You can create Spring Boot project by providing project details.

For more information click here.

6) How to create Spring Boot project using boot CLI?

It is a tool which you can download from the official site of Spring Framework. Here, we are explaining steps.

7) How to create simple Spring Boot application?

To create an application. We are using STS (Spring Tool Suite) IDE and it includes the various steps that are explaining in steps.

8) What are the Spring Boot Annotations?

The @RestController is a stereotype annotation. It adds @Controller and @ResponseBody annotations to the class. We need to import org.springframework.web.bind.annotation package in our file, in order to implement it.

For more information click here.

9) What is Spring Boot dependency management?

Spring Boot manages dependencies and configuration automatically. You don't need to specify version for any of that dependencies.

Spring Boot upgrades all dependencies automatically when you upgrade Spring Boot.

For more information click here.

10) What are the Spring Boot properties?

Spring Boot provides various properties which can be specified inside our project's **application.properties** file. These properties have default values and you can set that inside the properties file. Properties are used to set values like: server-port number, database connection configuration etc.

For more information click here.

11) What are the Spring Boot Starters?

Starters are a set of convenient dependency descriptors which we can include in our application.

Spring Boot provides built-in starters which makes development easier and rapid. For example, if we want to get started using Spring and JPA for database access, just include the **spring-boot-starter-data-jpa** dependency in your project.

For more information click here.

12) What is Spring Boot Actuator?

Spring Boot provides actuator to monitor and manage our application. Actuator is a tool which has HTTP endpoints. when application is pushed to production, you can choose to manage and monitor your application using HTTP endpoints.

For more information click here.

13) What is thymeleaf?

It is a server side Java template engine for web application. It's main goal is to bring elegant natural templates to your web application.

It can be integrate with Spring Framework and ideal for HTML5 Java web applications.

For more information click here.

14) How to use thymeleaf?

In order to use Thymeleaf we must add it into our pom.xml file like:

```
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-thymeleaf</artifactId>
</dependency>
```

15) How to connect Spring Boot to the database using JPA?

Spring Boot provides **spring-boot-starter-data-jpa** starter to connect Spring application with relational database efficiently. You can use it into project POM (Project Object Model) file.

For more information click here.

16) How to connect Spring Boot application to database using JDBC?

Spring Boot provides starter and libraries for connecting to our application with JDBC. Here, we are creating an application which connects with Mysql database. It includes the following steps to create and setup JDBC with Spring Boot.

For more information click here.

17) What is @RestController annotation in Spring Boot?

The @RestController is a stereotype annotation. It adds @Controller and @ResponseBody annotations to the class. We need to import org.springframework.web.bind.annotation package in our file, in order to implement it.

For more information click here.

18) What is @RequestMapping annotation in Spring Boot?

The **@RequestMapping** annotation is used to provide routing information. It tells to the Spring that any HTTP request should map to the corresponding method. We need to import org.springframework.web.annotation package in our file.

For more information click here.

19) How to create Spring Boot application using Spring Starter Project Wizard?

There is one more way to create Spring Boot project in STS (Spring Tool Suite). Creating project by using IDE is always a convenient way. Follow the following steps in order to create a Spring Boot Application by using this wizard.

For more information click here.

20) Spring Vs Spring Boot?

Spring is a web application framework based on Java. It provides tools and libraries to create a complete cutomized web application.

Wheras Spring Boot is a spring module which is used to create spring application project that can just run.

HR Interview Questions | Common Interview Questions

A list of frequently asked **HR interview questions** or **Common interview questions** or **Job interview questions** and answers are given below.

1) Tell me about yourself?

This is the most famous question for an interviewer and also most difficult to answer this question for the candidate. This question puts all the pressure on the candidate, and the interviewer relax.

You should alert enough to answer this question. You should start with an easy and confident tone and answer in a proper manner. It should not be scripted. Always remember, you are not giving the interview to a robot so your articulation, your pronunciation of each word should be clear and confident.

A good way:

Analyze your interviewer interests.

Express your most important accomplishments first.

Possible Answer 1

"Good morning/afternoon/evening" sir/mam.

First of all, thank you for giving me this opportunity to introduce myself.

My name is Ajeet Kumar.

As far as my education qualification is concerned, I have done MBA with finance stream from Srivenkateswara university in Emerald's P. G. College, Tirupathi, in the year of 2014.

I had completed B.tech from N.I.T Jaipur in 2012.

I had completed my schooling from G.I.C. Allahabad.

As far as concerned my family, I belong to a middle-class family. My father is a Businessman, and my Mother is a homemaker. My brother is preparing for civil services.

I am good in programming languages C, C++, and Java and very much interested in HTML, CSS, ASP. Net and SQL.

My strength is self-confidence, positive attitude, hard work.

My weakness is: I can easily believe every one.

My hobbies are: Watching news channels, Playing volleyball, Listening to music.

Possible Answer 2

"Good morning/afternoon/evening" sir/mam, it's my pleasure to introduce myself. I am Anshika Bansal. I belong to Meerut. I have done my B.Tech in CSE from Lovely Professional University.

> While coming to my family members, there are 4 members including me. My father is a doctor, and any mother is a teacher. My younger sister will appear her 12th CBSE board exam this year.
>
> Now coming to me, I am sweet smart, confident, and hardworking person. I am a cool hearted person, so usually see every difficulty with a positive side and keep myself always smiling which makes me stronger even more.
>
> I can carry out any task assigned to me without hesitation.
>
> My hobbies are dancing, Internet surfing, playing Chess, listening to music, watching the news channel. In my spare time, I like to read news on my phone and traveling to my hometown.
>
> Thank you for giving this opportunity to introduce myself.

Possible Answer 3

> "Good morning/afternoon/evening" sir/mam, it's my pleasure to introduce myself. I am Anshika Bansal. I belong to Meerut. I have done my B.Tech in CSE from Lovely Professional University.
>
> I am carrying 5 years of experience at top Wall Street Companies. In my recent company, I led the development of an award-winning new trading platform. I can survive in a fast-paced environment.
>
> Now I am looking for a chance to apply my technical expertize and my creative problem-solving skills at an innovative software company like yours.

---

2) Why are you applying for this job? (or)

Why this role attract you?

By this question, the interviewer wants to know that:

- If you fully understand what the job entails
- How well you might match their requirement

- What appeals to you most about this job

Before answering this question, take your own time an answer in the way that convinces the interviewer. Explain your qualities according to the above-stated points.

Possible Answer 1

> I have applied for this vacancy because it is an excellent match for my skills and experience. This role is exactly the sort of role I am currently targeting, and I am confident I will be able to make a major contribution.

Possible Answer 2

> Sir, it's a great privilege to work in a reputed company like yours. When I read about your requirement, I found that my skills are matching with them. Through this role, I can show my technical skills to contribute to the company growth.

3) Would you like to work overtime or odd hours?

You should become very honest to answer this question. Don't tell a lie or compromise to get the job only. If you don't have any problem, you can answer like this:

> I know that in the company being asked to work for an extended number of hours comes with a good reason, so I am ok with it. It an extra effort means I am doing something for the company, I'll be happy to do it.

4) What is more important to you: the money or the work?

This is a very tricky question. The work should always be more important than the money. This frame of mind is good for you(at least at the time of interview).

Possible Answer 1

> "Money is always important, but the work is most important for me."

Possible Answer 2

> "I would say that work is more important. If we work and achieve Company goals then obviously money would follow. I believe work to be prior."

Possible Answer 3

> "Work is more important for me. Working just for money may not be fulfilled if I don't feel satisfied with my job. My work makes me stay productive, and money would naturally come along well."

Possible Answer 4

> "I think money probably matters to me about as much as it does to anyone. It's vital and necessary for us to live and prosper but, at the same time, it's not my single most important driving force. I believe that money is rewarded for work."

5) What do you know about this organization?

You should fully aware of that organization where you are going for an interview, so check the history, present structure and working style of that organization. Check the company's website, Facebook, Twitter, Google+, LinkedIn pages to gather the information.

Possible Answer 1

> We all know that it is one of the fastest growing infrastructure company in India. The facilities provided to the employee is best. People feel proud to be the part of your company as the company provides full support to their employees in professional front. The working environment of this company is decent. It has crossed the branches in the world also. And I was in search of this type of company.

Possible Answer 2

> We all know that this company is India's no.1 company for development. I was delighted to see on your company website that your employees are talking about how great it is to work for your company. Now these days, so many people seem to hate the company where they work for one reason or another. It's great to see that your employees are proud to talk about how much they love their company and jobs.

6) Why did you leave your last job?

You should be very careful with this question. Avoid trashing other employers and making a statement like **"I need more money".** Instead of this, you can say that:

> Sir, it's a career move. I have learned a lot from my last job, but now I am looking for new challenges to broaden my horizons and to gain a new skill-set.

7) Why should we hire you?

Tell your qualifications and highlight that points which makes you unique.

Possible Answer 1

> "I believe that everyone starts with a beginning, I need a platform to prove my abilities and skills. I think your company is the right place to explore my abilities. I need to be a part of your growth. I will do my level best."

Possible Answer 2

> "As a fresher, I need a platform to prove my ability. If I will be a part of your company, I'll put my effort and strength to uplift your company. None is born with experience, and if you hire me, I will get professional experience through your company."

Possible Answer 3

> "Sir, as I am a fresher, I have theoretical knowledge, but I need a platform where I can implement my knowledge in the practical field. I am ensuring you that I will put all my efforts for the good progress of the organization. As a fresher, I have no preset mind regarding work culture in an organization, and this will help me to adapt the working culture of your company very easily. Being punctual and regular, I can finish the work giving to me on time and try my best to fulfill all the needs of the company."

Possible Answer 4

> "I have a good experience in that particular field (field of your specialization), and I think my talents will be a big

> contribution to the continuing pursuit of excellence of your company."

8) What are your salary expectations?

Don't ask your salary in exact numbers, instead of this show your commitment to the job itself.

Possible Answer 1

> I am more interested in the role than the pay, and I expect to be paid appropriate money for this role based on my experience. As you also know that the high cost of living here in Delhi.

Possible Answer 2

> As I am fresher, Salary is not an issue for me. Learning and gaining experience is my major priority. As your company is one of the most reputed company, I just accept the salary offered by you is the best in the industry.

Possible Answer 3

> As of now, I haven't thought much about it. I am more focused on learning the requirements for this position that I am applying for.

9) Assume you are hired, then how long would you expect to work for us?

Possible Answer 1

> "I will do my best for the growth of your company as long as I have the career growth, job satisfaction, respect and a healthy environment, then I don't need to change my company."

Possible Answer 2

> "I will work with the company as long as my presence benefits the company and I get ample opportunity to grow and develop both professionally and monetarily."

Possible Answer 3

> "Everyone looks for a bright future, healthy work environment, good salary, job satisfaction and I am pretty sure that your company gives such things, so I don't need to change the company."

Possible Answer 4

> "I will work with the company as long as my presence benefits both the company and mine in parallel. So your company gains good results, and I can be in a good position to improve my skills."

---

10) How would you rate yourself on a scale of 1 to 10?

Possible Answer 1

> I will rate myself 8 out of 10 because I would never like to think that there should be a room left for putting in more efforts. That thought will create an interest in learning the things. Thank you very much for giving me this wonderful opportunity.

Possible Answer 2

> I will answer this question based on some parameters. As far as hard work is concerned, I will rate myself as 8 because there should always be a scope to increase our skills which will create an interest in learning the things. When it comes to creativity, I would like to rate myself as 9. In the past, I have designed banners and brochures which were appreciated by the clients. To talk about patience, I will tag myself with 6 because I am an entry-level professional. Same as personal life, even professional life needs more experience for more patience. That is probably why in most companies, senior management looks more patient than entry level or even middle level. Overall, I would rate myself as 8 on a scale of 1 to 10.

11) What are your achievements in life?

This question may also be asked that what are your biggest achievements? Or what are you most proud of?

You should discuss only work-related achievements. Try to pick a recent achievement.

Possible Answer 1

> "My greatest achievement so far in my career would probably be winning the Manager of the Year award last year."

Possible Answer 2

> "My biggest accomplishment is overcoming my fear of failure. It gives me a complete sense of living and makes me more confident."

Possible Answer 3

> "I experienced my greatest achievement when I worked as a website manager for an entertainment outlet. My team was under pressure and the website was struggling at the time, and I was tasked with forming a strategy to increase traffic.

12) What is your objective in life?

Your answer should be realistic and practical.

Possible Answer 1

> "My short-term goal is to work in a reputed organization like yours where I can enhance my technical skills and knowledge.
>
> My long-term goal is to see the company at a topmost position where I am one of the people responsible for that."

Possible Answer 2

> "My goal is to become a successful person and make my family proud of me."

13) What are your strengths?

You should always remember that even if your strength is not business related, find a way to relate it to work. Tell your positive points related to the job.

Possible Answer 1

> My main strengths are the ability to use my initiative to take on challenges. I am always proactive at what I do, and that keeps my mind stimulated and focused.

Possible Answer 2

> My greatest strengths would be my intelligence and thoughtfulness . I believe that in every work environment you need to process every step and be detailed in your work.

Possible Answer 3

> My time management skills are excellent, and I'm organized, efficient, and take pride in excelling at my work.

Possible Answer 4

> I always understand the value of time, and I am always able to innovate. I listen to advice from others.

Possible Answer 5

> My greatest strength is my ability to focus on my work. I'm not easily distracted, and this means that my performance is very high.

Possible Answer 6

> My biggest strength is my Confidence . Apart from that, I am Hardworking, self-motivated with a positive attitude towards my career and my life. If I have a problem, then I think its an opportunity for me to express my ability.

14) What are your weaknesses?

Everyone has weaknesses so while answering this question don't spend so much time on this. This question is generally asked to know how honest you are with yourself. State one or two minor weaknesses and try to relate it works. (avoid saying "I work too hard" it is a very common answer). Don't pretend you don't have weaknesses and don't avoid answering the question.

Possible Answer 1

> You can say, "I used to be very disorganized, always forgetting assignments and birthdays. But I managed to work out a computerized system of to-do lists and reminders that keep me on top of everything."

Possible Answer 2

> "I am a straightforward person, and I cannot say no when someone asks me for help."

15) What are your hobbies?

It totally depends on you what you like and what hobbies do you have but always justify your answer.

For Example:

> My hobbies are dancing, Internet surfing, playing Chess, listening to music, watching the news channel. In my spare time, I like to read news on my phone and traveling to my hometown. Thank you for giving this opportunity to introduce myself.

16) Explain, how would you be an asset to this organization?

While answering this question, don't merely say that you are hard working, dedicated, punctual, etc. because these are not assets, these are the responsibility that management wants. Off course these are positive words and has a lot of meaning for you as well as for the company. Actually, these qualities make you an asset for the company after getting experience.

Possible Answer 1

> "To become an asset for an organization, we have to punctual, dedicated, quickly adapt of the environment and positive working attitude I have all of these qualities so I will prove an asset for this company."

Possible Answer 2

> "As I'm a fresher, I'll be passionate about my work, and if given a chance I'll be ready to expose myself into different sectors of work into the industry, and would be faithful to the company always."

Possible Answer 3

> "My skill in XYZ company is outstanding. I have earned a lot of certificates and awards from my past employers. As an employee, I can handle pressure with ease and can work with minimal supervision."

---

17) Would you lie for the company?

You should answer this question in a very diplomatic manner.

Possible Answer 1

> "It depends on the situation if my lie creates a positive impact on the company and It was useful for many people, then I will lie."

Possible Answer 2

> "Probably I would say no for a lie. But if condition persists that my lie will help thousands of persons and it is not giving any impact to other persons, then I will agree with the lie. My

> one lie results lot of success and happiness without hurting others rights then I expect to do this."

18) Are you applying for other jobs also?

This question has a very specific purpose. The interviewer wants to know that if the candidate admits that he has applied to other places or gives a canned answer. Or what the candidate think about their company.

You should never claim that you did not apply to other company. Despite this, you can say that -

> As you know, I would like to work as a software developer. Beyond your company, I have also applied to XYZ company and ABC company. This answer is more specific.

19) How do you get to know about our company?

The interviewer should not lie while answering such questions. Because an interviewer is a brilliant person and they can detect it if you are lying.

For Example

> I get to know about your company from several online websites.

20) What does success mean to you?

Possible Answer 1

> If I feel I am making a difference working with a team of people to make a more profitable company. It is a success for me.

Possible Answer 2

> If I put a smile on someone face and make him happy, it is a success for me.

21) Describe yourself in one word?

This question is asked to the candidate to judge how spontaneous and explaining he is. If he uses a word, he must be able to explain that word and also represent himself according to that word.

These are some positive words. You can use it but be sure that you are judging with the word.

> Original, genuine, logical, incredible, focused, curious, active, quick, balanced, achiever, etc.

22) What is the difference between confidence and overconfidence?

Confidence is based on facts and knowledge, and overconfidence is based on speculation. There is a small difference between confidence and overconfidence. Yes, I can do this work is self-confidence. But only I can do this work is overconfidence.

Or you can say that: confidence is an internal belief that I am a right person for this job and overconfidence is thought that I am only the right person for this job.

23) What is the difference between smart work and hard work?

The difference between smart work and hard work are as follows:

> Smart work and hard work are related to each other. Without being a hard worker, we can't be a smart worker. Smart work

> comes from the hard work. That means everyone has to specialize in his work to become a smart worker. So, all of us have to do hard work to achieve smart work.
>
> So hard work increases your accuracy, and smart work increases accuracy as well as efficiency.

24) Just imagine that you have enough money to retire right now. Would you?

If your answer is YES, you are surely get FAILED.

You can say that:

> "No sir, I don't think so. I am a professional, and I love my work, so there is no question to leave my work. Yes, it may be that I would take a break to spend quality time with my family."

25) Don't you think that you are overqualified for this position?

This is trick of the interviewer to trap you and judge how boasting you are?

So, be alert to answer this question and don't even hint to the interviewer that you are overqualified although you are.

This is a good answer:

> "I think I am qualified for this post because I am just a beginner and have lots more to learn. I think qualification is not only a matter of theoretical knowledge or results; it also depends on practical knowledge and learning. When I get practical knowledge, I will consider myself just a well-qualified not overqualified. Because learning never ends."

26) Do you have any blind spot?

It is a trick question. So don't specify your actual weaknesses. Instead of this try to show you are very team oriented

For example:

> "I would say everyone has blind spots and I would too that's why I believe in teamwork because when you are a team, you can point out the blind spots of other people, and they will also do the same for you."

Note: "don't admit failure as a blind spot. Failure is not a blind spot."

27) How do you handle stress, pressure, and anxiety?

It is a typical interview question asked to judge how you handle the job stress and pressure.

Possible Answer 1

> I believe in working in a planned way and finishing my work regularly. I always react to situations, rather than to stress. In this way, I handle the situation and don't take the stress.

Possible Answer 2

> I go to the gym regularly and do work out. In this way, I remove my stress. I believe that physical exercise is a great stress reducer.

28) What is the disappointment in your life?

This question is asked to judge, what type of situations make you uncomfortable and disappointed. While answering this question, don't just say your disappointment. Instead of this, you should include what you learn from that incident.

Possible Answer 1:

> "The greatest disappointment in my life so far as when my laptop had stolen, and I had lost all my important data. I remember that I had to work hard for the next 10 days to redo all my work to make sure to submit on time."

Possible Answer 2: If you did not face any disappointment in your life

> "Sir, I don't feel that I have faced such type of situation."

29) What makes you angry?

You should always answer this question in a manner that suits both your personality and management too.

> "Sir, I am not a short-tempered person, but I feel a bit of annoyance when someone disturbs me in my work without a genuine reason. Although I am an even-tempered person, when I get angry, I try to channel my negative feeling in my work."

30) What was the most difficult decision you have made in your past life?

This question is asked to judge your decision-making capabilities. The interviewer wants to know, how you take a decision in tough times.

Possible Answer 1

> After completing my graduation, the toughest decision is whether to go for higher studies or do a job. Then I chose the job because getting trained is better than educated and it was also the demand of that time.

Possible Answer 2

> My toughest decision was to take admission in B.tech. I belong to a middle-class family, and my father was not in favor of taking admission, but I convinced him, and today he is very happy.

Possible Answer 3

> Before some time when I had to choose between joining a group of employees protesting some issue, and staying away from the issue. I ended up being a mediator between our immediate supervisor and employees, and I am glad I made that decision because it all ended well and without further conflicts in the company.

---

31) How did you know about this position/ vacancy?

Just tell the source from where you got the information about this post. If the interviewer asks, what you know about this position or what appeals you most in this position?

Then you can add:

> I have carefully studied both the job description and the person specification, so I am fully aware of the duties and responsibilities of this role.

32) What gets you up in the morning?

It is my promise that gets up me in the morning. My promise is to learn something new and someone in need. It provides me the satisfaction that I am making a difference in someone life.

33) What is your favorite book?

This question is asked to judge your taste about reading books. The interviewer wants to know what types of book you like. Would you fit for the company culture?

Answer this question according to your sense, your knowledge about the book. Only named the books you have really read. You should choose something from a reputable author that your interviewer has probably heard of.

34) As you said, internet surfing is your hobby. Which site do you surf mostly?

This is your choice that which sites you surf most, but while answering this question always refers to sites which are relevant to your field of job. Don't take the name of social networking sites or other irrelevant sites.

35) What was the biggest mistake of your life?

At the time of answering this question, you should choose a story containing not too many mistakes. It should be real.

Talk about the mistake, but it is also important to convince the interviewer that you never make the same mistake again.

For example:

> I think the worst mistake I ever made at work was in my first ever job - five years ago now. A more senior member of the team seemed to take an instant dislike to me from the start, and one day she was particularly unpleasant to me in front of several colleagues.
>
> Later on, I was talking to one of those colleagues who was, I thought, attempting to console me. Angry and hurt, I foolishly vented my feelings and told her what I thought of the lady in question. I was naturally shocked to find out that she went on to tell everyone what I had said and this certainly didn't help my relationship with the team member who was causing me problems.
>
> Rather than let the situation carry on, I chose to have a quiet word with this lady to find out what her problem was with me and to see if we could put it behind us. It turned out it was nothing personal; she just resented the fact that a friend of hers had also been interviewed for my position and had been turned down. Once we had got matters out into the air, her behavior changed, and we got on quite well after that. However, I certainly learned a lot from experience. I learned that careful communication is vital in managing interpersonal relationships and that if I have a problem with someone, it's always best to talk it over with them rather than with someone else.

36) Do you have any reference?

This question has just two answers, YES or NO. If you know anyone from that company, say YES, otherwise NO.

37) How do you deal with an angry or irritated customer?

Possible Answer 1

> I would try to find out exactly what the problem was, and evaluate if there was something I could do to make it right.

Possible Answer 2

> I would ask the customer to explain his problem and carefully listen to him. After that, I do my best to solve his problem. If that problem is not regarding my work area, I spoke to someone who could help him immediately.

---

38) How do you deal with an angry or irritated customer?

Possible Answer 1

> I would try to find out exactly what the problem was, and evaluate if there was something I could do to make it right.

Possible Answer 2

> I would ask the customer to explain his problem and carefully listen to him. After that, I do my best to solve his problem. If that problem is not regarding my work area, I spoke to someone who could help him immediately.

---

39) What is your greatest fear?

Possible Answer 1

> Before some time, public speaking has been a challenge for me. I was very nervous and hesitate while giving any presentation, so I started taking public speaking seminars to improve it. Now, I still get nervous before pitches, but I have learned how to remain calm and get the job done right.

Possible Answer 2

> As you know that I have never worked in my life and this is my first job, my inexperience is my weakness. But I beg to differ. I am confident and a fast learner. I assure you that I will perform my job without carrying any pre-conceived notions regarding how I feel.

---

40) Explain the difference between group and team?

The difference between group and team are as follows:

> There is only one difference between the group and the team. That is unity. Any set of people who stand together without any purpose or goal can be called as Group. Whereas, when more than 2 people work towards a common goal, can be called as a Team. For example: If you assign work to a group, then the work will be divided between the members and each member will work out their part, without any coordination with the other members of the group. On the other hand, if you assign a project to a team, they will collectively take the responsibility and work together with the goal to achieve the desired result. In a team, the members will cooperate and coordinate with each other at all times.

---

41) What will you do if you don't get this position?

Possible Answer 1

> I have high hopes that I will be selected. In case if I will not select, I will continue to look for another job in the same field that will fit my schedule and goals.

Possible Answer 2

> If I don't get this job, I will use this experience to reflect my weakness and try my best to improve on them for the future opportunities along the way.

42) What is the success for you?

Success refers to the accomplishment of an aim or purpose.

43) Would you like to relocate or travel for the company?

Firstly, you need to understand the purpose of this question. The fresher candidate does not say YES at once. This will only show your desperation to this job. You can answer like this:

Possible Answer 1

> I will definitely consider traveling if the opportunity is appropriate, rewarding and feasible. I don't think that I will have any problem with the traveling involved.

Possible Answer 2

> Yes, as part of growing up. I used to travel a lot as my father is an ex-serviceman hence occasionally he used to be posted

> throughout the country. Though I would prefer my city as it is my birthplace and all my friends and relatives are there, but at the same time, I am ok with relocating.

44) What makes you happy?

I feel happy when I accomplished my task. I also feel happy when I achieved my goals. Holiday with my family and friends also makes me happy.

45) Is there anything which makes you different from other candidates?

It is good to start with a disclaimer that you are not aware of the strengths of the other candidate. Also, you sure that some impressive candidates apply for this position.

Possible Answer 1

> Although, I am not familiar with the others whom you are interviewing for this position. I am sure many talented people applying for the job. But because of my background and the problem-solving skills, I considered myself to be a strong candidate for this position.

Possible Answer 2

> I understand that success is not always guaranteed but there is still hope, and I never lose the faith, whether I succeed or not. I think this power makes me stand alone from all other candidates.

Possible Answer 3

> I have exceptional organizational skills. In my last job, I created a project which was termed as quite creative by the clients. I think my technical skills make me stand alone from all other candidates.

46) Describe the three things that are most important for you in a job?

Possible Answer 1

> According to me, Honesty, Loyalty, and determination to achieve my team's target are the three important things in a job.

Possible Answer 2

> According to me, Professionalism, growth and a healthy work-life balance are the three important things in a job.

47) What are your expectations from the company?

Though this answer is objective and can be different for different persons, but you should be positive in your thoughts and do not say many things about the company which gives the interviewer an illusion that you are exaggerating. In short, be realistic and precise.

For Example

> I have always wanted to work with an organization which provides a very comfortable and home like work environment. I would like to work in the company where I can get the opportunity to learn and enhance my skill to become a better professional in the future.

48) On a scale of one to ten, rate me as an interviewer?

Possible Answer 1

> Sir, I'm not in the position to rate you as an interviewer. Anyway, I'm not going to disappoint you. As an interviewer you've fulfilled your job, So, I can give you 10 out of 10. But I'll give you 9/10 as there should always be a scope to increase our skills which will create an interest in learning the things. Thank you very much for giving me this wonderful opportunity.

Possible Answer 2

> Thank you for giving me such an opportunity, but for sure I am not the person to rate you. As it is obvious that your position is highly reputed and you have been chosen to undergo this process that shows your excellence at this place, but still I have to answer this question so honestly, it would be 9/10 as no one is perfect and we always leave room for improvement.

49) Who is your role model? What have you incorporated into your life from him/her?

You should think of people who embody the qualities that you most admire.

For Example

> The role model of my life is my mother. Whenever I am down my mother helps me to push my limits, and she always keeps me on the track. She was scolding me whenever I do

> something wrong. She is everything for me, I still got inspired from her, how she manages every problem in every situation.

50) Do you have any questions for me?

It's your turn now. If you get such an opportunity, you may ask questions like that:

Possible Answer 1

> Thank you for giving me this opportunity. After my overall performance till now if I got selected what I need to improve and if I'm not selected how can I succeed further. Can you give any advice sir?

Possible Answer 2

> First of all thank you very much, for being so much polite & friendly to me throughout the session, that I can express myself so easily. Can you please tell me that what are the qualities you are expecting from fresher like us & I want to know, if am selected, then what should I improve before I join your company, if I am not selected, your opinion will help me to the upcoming interview.

OOPs Interview Questions

Object-oriented programming (OOPs) is a programming paradigm that is based on the concept of objects rather than just functions and procedures. It is the most popular methodology among developers.

Nowadays tech giants demanding and hiring who has expertise in **object-oriented** approaches and patterns and conducting interviews for the same. The advantage of hiring such candidates is that they can also learn other OOP languages easily as per organization requirements. Since, going through the section, you can increase your chance to get hire by companies if you have well prepared for **OOPs interview questions**.

In this section, we have collected some commonly asked **OOPs interview questions** for both fresher and experienced. It can help you to crack the interview to get your dream job.

1) What do you understand by OOP?

OOP stands for object-oriented programming. It is a programming paradigm that revolves around the object rather than function and procedure. In other words, it is an approach for developing applications that emphasize on objects. An object is a real word entity that contains data and code. It allows binding data and code together

2) Name any seven widely used OOP languages.

There are various OOP languages but the most widely used are:

- Python
- Java
- Go
- Dart
- C++
- C#
- Ruby

---

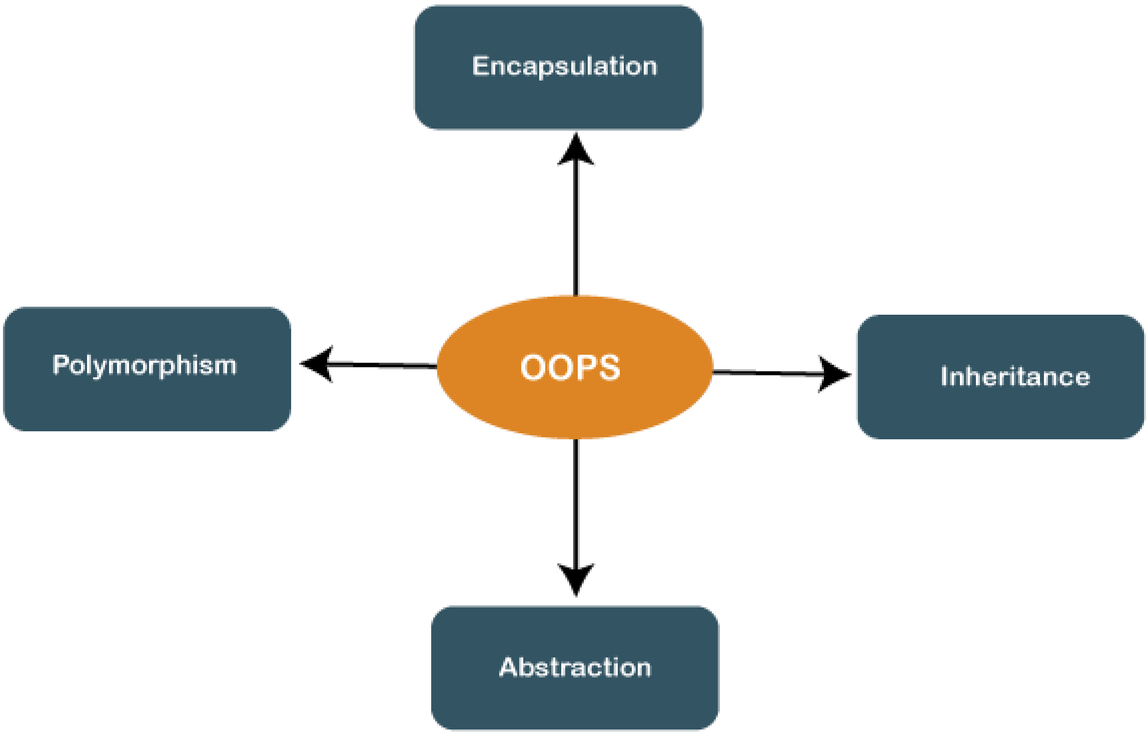
3) What is the purpose of using OOPs concepts?

The aim of OOP is to implement real-world entities like inheritance, hiding, polymorphism in programming. The main purpose of OOP is to bind together the data and the functions that operate on them so that no other part of the code can access this data except that function.

4) What are the four main features of OOPs?

The OOP has the following four features:

- Inheritance
- Encapsulation
- Polymorphism
- Data Abstraction



5) Why OOP is so popular?

OOPs, programming paradigm is considered as a better style of programming. Not only it helps in writing a complex piece of code easily, but it also allows users to handle and maintain them easily as well. Not only that, the main pillar of OOPs - Data Abstraction, Encapsulation, Inheritance, and Polymorphism, makes it easy for programmers to solve complex scenarios. As a result of these, OOPs is so popular.

6) What are the advantages and disadvantages of OOP?

Advantages of OOP

- It follows a bottom-up approach.
- It models the real word well.
- It allows us the reusability of code.
- Avoids unnecessary data exposure to the user by using the abstraction.
- OOP forces the designers to have a long and extensive design phase that results in better design and fewer flaws.
- Decompose a complex problem into smaller chunks.
- Programmer are able to reach their goals faster.
- Minimizes the complexity.
- Easy redesign and extension of code that does not affect the other functionality.

Disadvantages of OOP

- Proper planning is required.
- Program design is tricky.
- Programmer should be well skilled.
- Classes tend to be overly generalized.

7) What are the limitations of OOPs?

- Requires intensive testing processes.
- Solving problems takes more time as compared to Procedure Oriented Programming.
- The size of the programs created using this approach may become larger than the programs written using the procedure-oriented programming approach.
- Software developed using this approach requires a substantial amount of pre-work and planning.
- OOP code is difficult to understand if you do not have the corresponding class documentation.

- In certain scenarios, these programs can consume a large amount of memory.
- Not suitable for small problems.
- Takes more time to solve problems.

8) What are the differences between object-oriented programming and structural programming?

Object-oriented Programming	Structural Programming
It follows a bottom-up approach.	It follows a top-down approach.
It provides data hiding.	Data hiding is not allowed.
It is used to solve complex problems.	It is used to solve moderate problems.
It allows reusability of code that reduces redundancy of code.	Reusability of code is not allowed.
It is based on objects rather than functions and procedures.	It provides a logical structure to a program in which the program is divided into functions.
It provides more security as it has a data hiding feature.	It provides less security as it does not support the data hiding feature.
More abstraction more flexibility.	Less abstraction less flexibility.
It focuses on data.	It focuses on the process or logical structure.

9) What do you understand by pure object-oriented language? Why Java is not a pure object-oriented programming language?

The programming language is called pure object-oriented language that treats everything inside the program as an object. The primitive types are not supported by the pure OOPs language. There are some other features that must satisfy by a pure object-oriented language:

- Encapsulation
- Inheritance

- Polymorphism
- Abstraction
- All predefined types are objects
- All user-defined types are objects
- All operations performed on objects must be only through methods exposed to the objects.

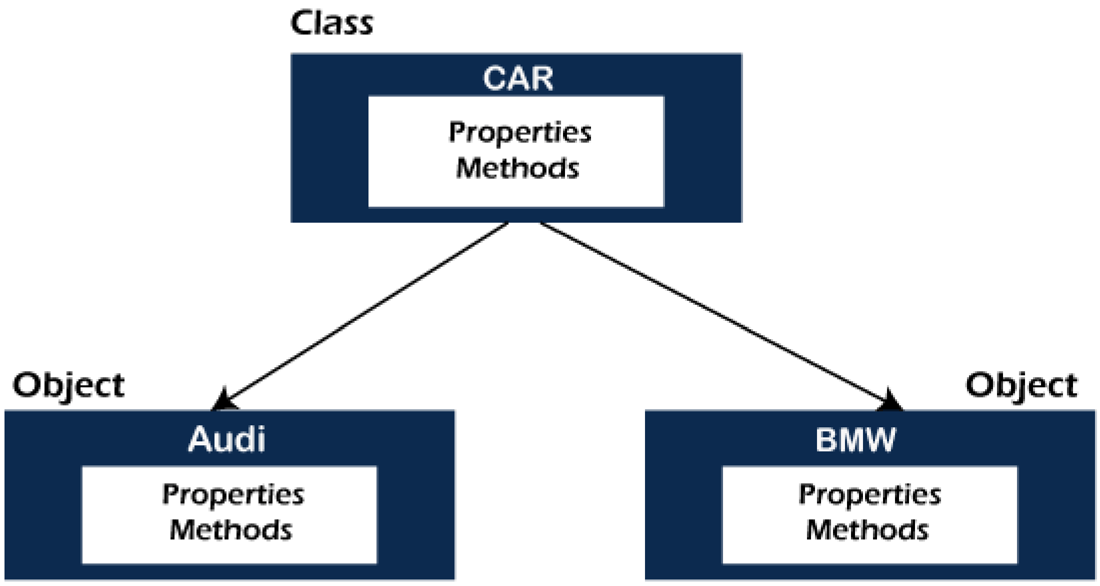
Java is not a pure object-oriented programming language because pre-defined data types in Java are not treated as objects. Hence, it is not an object-oriented language.

10) What do you understand by class and object? Also, give example.

Class: A class is a blueprint or template of an object. It is a user-defined data type. Inside a class, we define variables, constants, member functions, and other functionality. It does not consume memory at run time. Note that classes are not considered as a data structure. It is a logical entity. It is the best example of data binding.

Object: An object is a real-world entity that has attributes, behavior, and properties. It is referred to as an instance of the class. It contains member functions, variables that we have defined in the class. It occupies space in the memory. Different objects have different states or attributes, and behaviors.

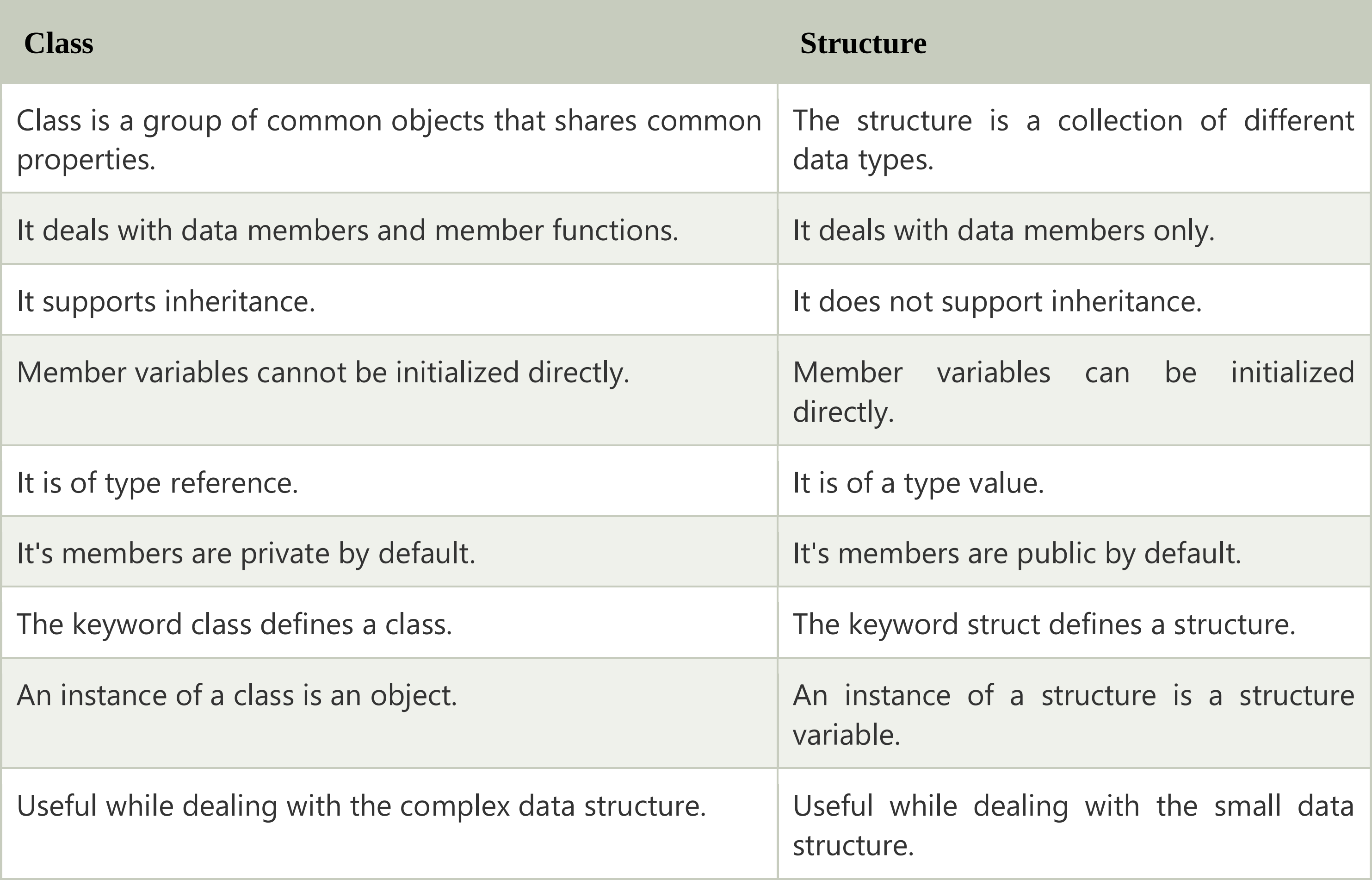
The following figure best illustrates the class and object.



11) What are the differences between class and object?

Class	Object
It is a logical entity.	It is a real-world entity.
It is conceptual.	It is real.
It binds data and methods together into a single unit.	It is just like a variable of a class.
It does not occupy space in the memory.	It occupies space in the memory.
It is a data type that represents the blueprint of an object.	It is an instance of the class.
It is declared once.	Multiple objects can be declared as and when required.
It uses the keyword class when declared.	It uses the new keyword to create an object.
A class can exist without any object.	Objects cannot exist without a class.

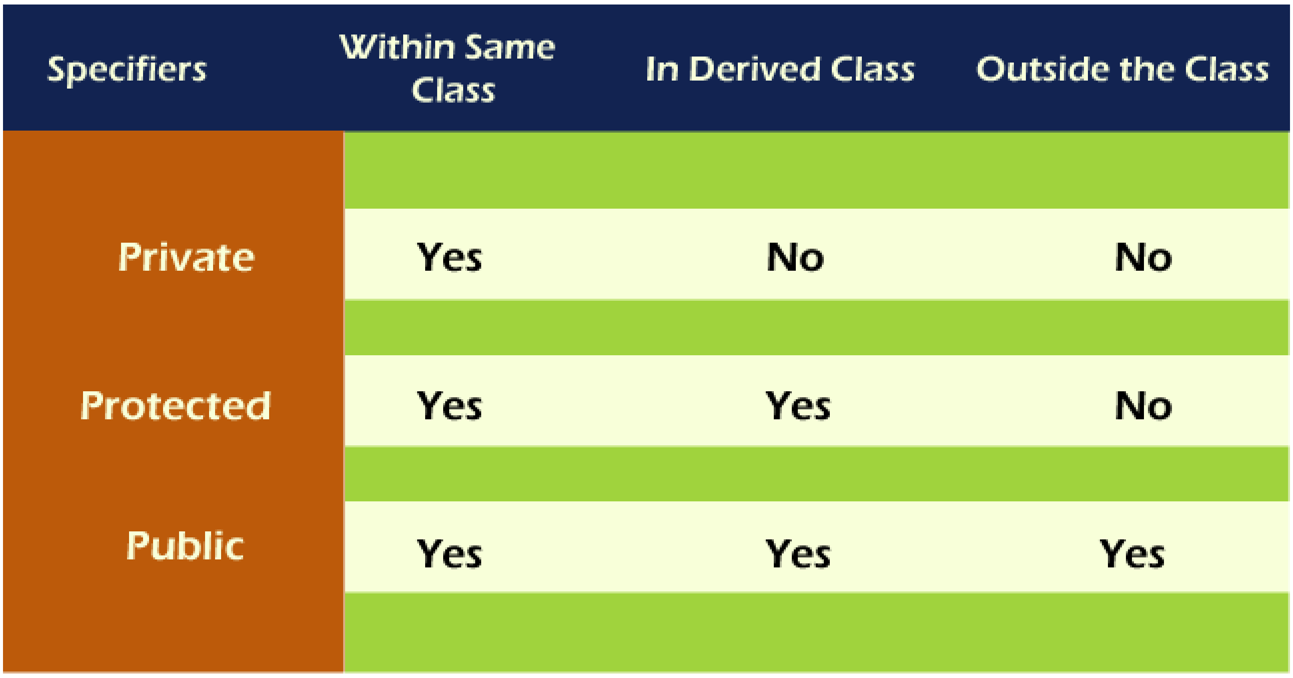
12) What are the key differences between class and structure?



Class	Structure
Class is a group of common objects that shares common properties.	The structure is a collection of different data types.
It deals with data members and member functions.	It deals with data members only.
It supports inheritance.	It does not support inheritance.
Member variables cannot be initialized directly.	Member variables can be initialized directly.
It is of type reference.	It is of a type value.
It's members are private by default.	It's members are public by default.
The keyword class defines a class.	The keyword struct defines a structure.
An instance of a class is an object.	An instance of a structure is a structure variable.
Useful while dealing with the complex data structure.	Useful while dealing with the small data structure.

13) What is the concept of access specifiers when should we use these?

In OOPs language, **access specifiers** are reserved keyword that is used to set the accessibility of the classes, methods and other members of the class. It is also known as **access modifiers**. It includes **public, private,** and **protected**. There is some other access specifier that is language-specific. Such as Java has another access specifier **default**. These access specifiers play a vital role in achieving one of the major functions of OOP, i.e. encapsulation. The following table depicts the accessibility.



Specifiers	Within Same Class	In Derived Class	Outside the Class
Private	Yes	No	No
Protected	Yes	Yes	No
Public	Yes	Yes	Yes

14) What are the manipulators in OOP and how it works?

Manipulators are helping functions. It is used to manipulate or modify the input or output stream. The modification is possible by using the **insertion** (<<) and **extraction** (>>) operators. Note that the modification of input or output stream does not mean to change the values of variables. There are two types of manipulators with **arguments** or **without arguments.**

The example of manipulators that do not have arguments is **endl, ws, flush,** etc. Manipulators with arguments are **setw(val), setfill(c), setbase(val), setiosflags(flag).** Some other manipulators are **showpos, fixed, scientific, hex, dec, oct,** etc.

15) What are the rules for creating a constructor?

- It cannot have a return type.
- It must have the same name as the Class name.
- It cannot be marked as static.
- It cannot be marked as abstract.
- It cannot be overridden.

- It cannot be final.

16) What are the differences between the constructor and the method in Java?

Constructor	Method
Constructor has the same name as the class name.	The method name and class name are not the same.
It is a special type of method that is used to initialize an object of its class.	It is a set of instructions that can be invoked at any point in a program.
It creates an instance of a class.	It is used to execute Java code.
It is invoked implicitly when we create an object of the class.	It gets executed when we explicitly called it.
It cannot be inherited by the subclass.	It can be inherited by the subclass.
It does not have any return type.	It must have a return type.
It cannot be overridden in Java.	It can be overridden in Java.
It cannot be declared as static.	It can be declared as static.
Java compiler automatically provides a default constructor.	Java compiler does not provide any method by default.

Procedural Oriented Programming	Object-Oriented Programming

It is based on functions.	It is based on real-world objects.
It follows a top-down approach.	It follows a bottom-up approach.
It is less secure because there is no proper way to hide data.	It provides more security.
Data is visible to the whole program.	It encapsulates the data.
Reuse of code is not allowed.	The code can be reused.
Modification and extension of code are not easy.	We can easily modify and extend code.
Examples of POP are C, VB, FORTRAN, Pascal, etc.	Examples of OOPs are C++, Java, C#, .NET, etc.

17) How does procedural programming be different from OOP differ?

18) What are the differences between error and exception?

Basis of Comparison	Exception	Error
Recoverable/ Irrecoverable	Exception can be recovered by using the try-catch block.	An error cannot be recovered.
Type	It can be classified into two categories i.e. checked and unchecked.	All errors in Java are unchecked.
Occurrence	It occurs at compile time or run time.	It occurs at run time.
Package	It belongs to java.lang.Exception package.	It belongs to java.lang.Error package.
Known or unknown	Only checked exceptions are known to the compiler.	Errors will not be known to the compiler.
Causes	It is mainly caused by the application itself.	It is mostly caused by the

		environment in which the application is running.
Example	**Checked Exceptions:** SQLException, IOException **Unchecked Exceptions:** ArrayIndexOutOfBoundException, NullPointerException, ArithmaticException	Java.lang.StackOverFlow, java.lang.OutOfMemoryError

19) What are the characteristics of an abstract class?

An abstract class is a class that is declared as abstract. It cannot be instantiated and is always used as a base class. The characteristics of an abstract class are as follows:

- Instantiation of an abstract class is not allowed. It must be inherited.
- An abstract class can have both abstract and non-abstract methods.
- An abstract class must have at least one abstract method.
- You must declare at least one abstract method in the abstract class.
- It is always public.
- It is declared using the **abstract**

The purpose of an abstract class is to provide a common definition of the base class that multiple derived classes can share.

20) Is it possible for a class to inherit the constructor of its base class?

No, a class cannot inherit the constructor of its base class.

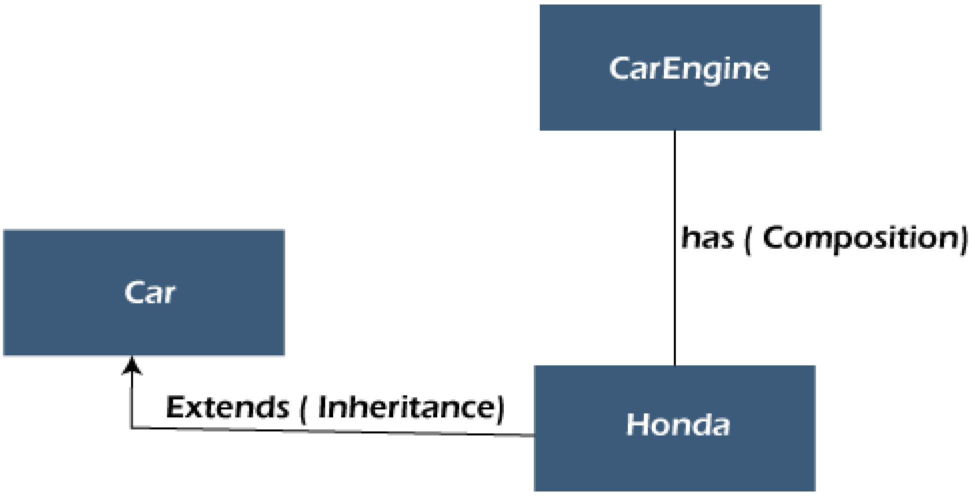
21) Identify which OOPs concept should be used in the following scenario?

A group of 5 friends, one boy never gives any contribution when the group goes for the outing. Suddenly a beautiful girl joins the same group. The boy who never contributes is now spending a lot of money for the group.

Runtime Polymorphism

22) What is composition?

Composition is one of the vital concepts in OOP. It describes a class that references one or more objects of other classes in instance variables. It allows us to model a has-a association between objects. We can find such relationships in the real world. For example, a car has an engine. the following figure depicts the same



The main benefits of composition are:

- Reuse existing code
- Design clean APIs
- Change the implementation of a class used in a composition without adapting any external clients.

23) What are the differences between copy constructor and assignment operator?

The copy constructor and the assignment operator (=) both are used to initialize one object using another object. The main difference between the two is that the copy constructor allocates separate memory to both objects i.e. existing object and newly created object while the assignment operator does not allocate new memory for the newly created object. It uses the reference variable that points to the previous memory block (where an old object is located).

Syntax of Copy Constructor

```
class_name (const class_name &obj)
{
//body
}
```

Syntax of Assignment Operator

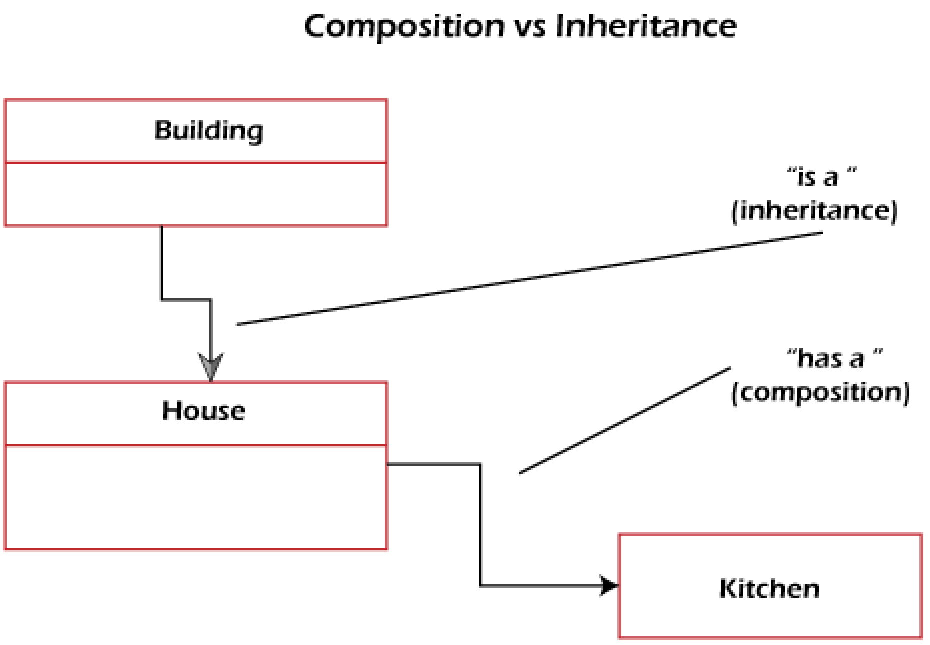
```
class_name obj1, obj2;
obj1=obj2;
```

Copy Constructor	Assignment Operator
It is an overloaded constructor.	It is an operator.
It creates a new object as a copy of an existing object.	It assigns the value of one object to another objec of which already exist.
The copy constructor is used when a new object is created with some existing object.	It is used when we want to assign an existing obje new object.
Both the objects use separate memory locations.	Both objects share the same memory but use th different reference variables that point to the location.

If no copy constructor is defined in the class, the compiler provides one.	If the assignment operator is not overloaded th bitwise copy will be made.

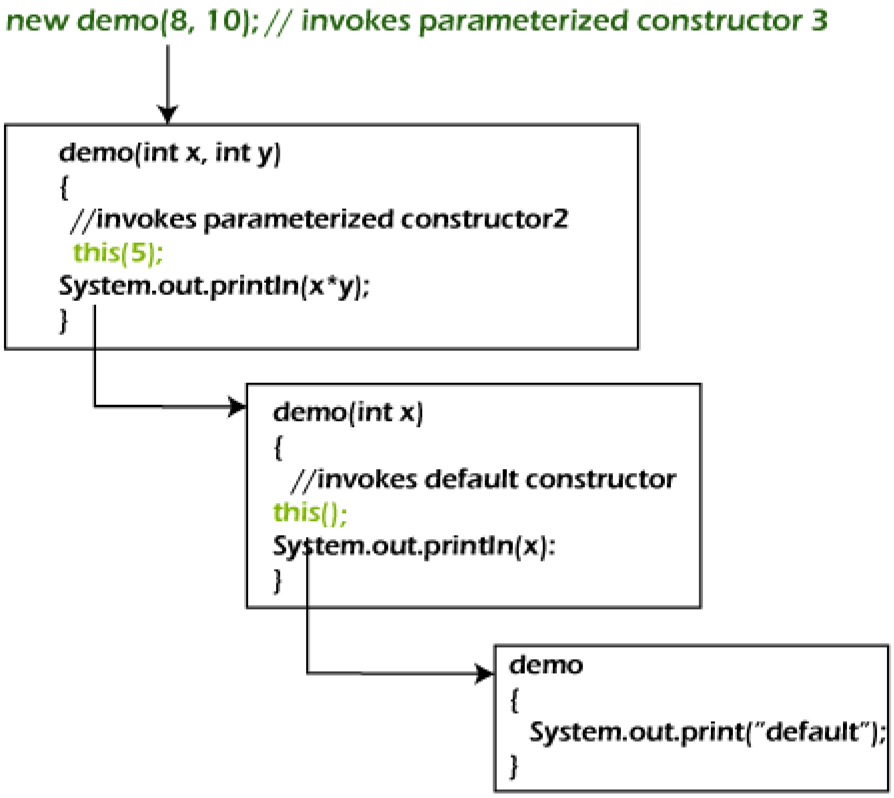
24) What is the difference between Composition and Inheritance?

Inheritance means an object inheriting reusable properties of the base class. Compositions mean that an object holds other objects. In Inheritance, there is only one object in memory (derived object) whereas, in Composition, the parent object holds references of all composed objects. From a design perspective, inheritance is "is a" relationship among objects whereas Composition is "has a" relationship among objects.



25) What is constructor chaining?

In OOPs, constructor chaining is a sequence of invoking constructors (of the same class) upon initializing an object. It is used when we want to invoke a number of constructors, one after another by using only an instance. In other words, if a class has more than one constructor (overloaded) and one of them tries to invoke another constructor, this process is known as constructor chaining. In C++, it is known as constructor delegation and it is present from C++ 11.



```
new demo(8, 10); // invokes parameterized constructor 3
```

```
demo(int x, int y)
{
 //invokes parameterized constructor2
 this(5);
System.out.println(x*y);
}
```

```
demo(int x)
{
  //invokes default constructor
this();
System.out.println(x):
}
```

```
demo
{
  System.out.print("default");
}
```

26) What are the limitations of inheritance?

- The main disadvantage of using inheritance is two classes get tightly coupled. That means one cannot be used independently of the other. If a method or aggregate is deleted in the Super Class, we have to refactor using that method in SubClass.
- Inherited functions work slower compared to normal functions.
- Need careful implementation otherwise leads to improper solutions.

Inheritance	Polymorphism

Inheritance is one in which a derived class inherits the already existing class's features.	Polymorphism is one that you can define in different forms.
It refers to using the structure and behavior of a superclass in a subclass.	It refers to changing the behavior of a superclass in the subclass.
It is required in order to achieve polymorphism.	In order to achieve polymorphism, inherence is not required.
It is applied to classes.	It is applied to functions and methods.
It can be single, hybrid, multiple, hierarchical, multipath, and multilevel inheritance.	There are two types of polymorphism compile time and run time.
It supports code reusability and reduces lines of code.	It allows the object to decide which form of the function to be invoked at run-time (overriding) and compile-time (overloading).

27) What are the differences between Inheritance and Polymorphism?

28) What is Coupling in OOP and why it is helpful?

In programming, separation of concerns is known as **coupling**. It means that an object cannot directly change or modify the state or behavior of other objects. It defines how closely two objects are connected together. There are two types of coupling, **loose** coupling, and **tight** coupling.

Objects that are independent of one another and do not directly modify the state of other objects is called loosely coupled. Loose coupling makes the code more flexible, changeable, and easier to work with.

Objects that depend on other objects and can modify the states of other objects are called tightly coupled. It creates conditions where modifying the code of one object also requires changing the code of other objects. The reuse of code is difficult in tight coupling because we cannot separate the code.

Since using loose coupling is always a good habit.

29) Name the operators that cannot be overload.

1. Scope Resolution Operator (::)
2. Ternary Operator (? :)
3. Member Access or Dot Operator (.)
4. Pointer to Member Operator (.*)
5. sizeof operator

30) What is the difference between new and override?

The new modifier instructs the compiler to use the new implementation instead of the base class function. Whereas, Override modifier helps to override the base class function.

virtual: indicates that a method may be overridden by an inheritor

override: Overrides the functionality of a virtual method in a base class, providing different functionality.

new: Hides the original method (which doesn't have to be virtual), providing different functionality. This should only be used where it is absolutely necessary.

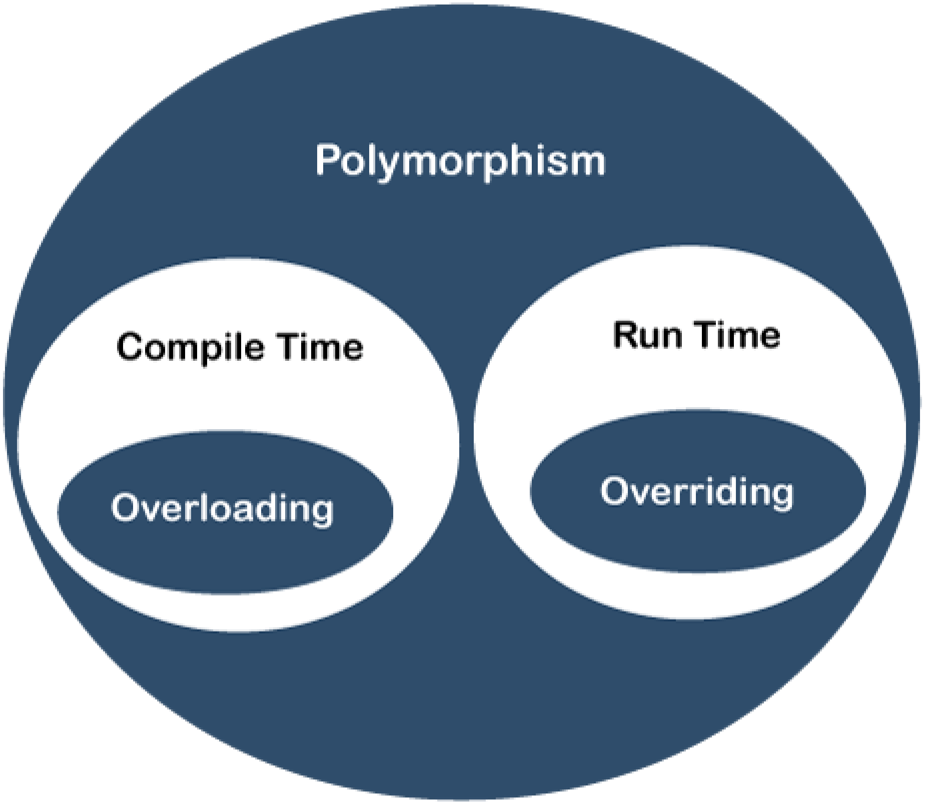
When you hide a method, you can still access the original method by upcasting to the base class. This is useful in some scenarios, but dangerous.

31) Explain overloading and overriding with example?

Overloading

Overloading is a concept in OOP when two or more methods in a class with the same name but the method signature is different. It is also known as **compile-time polymorphism**. For example, in the following code snippet, the method **add()** is an overloaded method.

```
public class Sum
{
int a, b, c;
public int add();
{
c=a+b;
return c;
}
add(int a, int b);
{
//logic
}
add(int a, int b, int c);
{
//logic
}
add(double a, double b, double c);
{
//logic
}
//statements
}
```



Overriding

If a method with the same method signature is presented in both child and parent class is known as method **overriding**. The methods must have the same number of parameters and the same type of parameter. It overrides the value of the parent class method. It is also known as **runtime polymorphism**. For example, consider the following program.

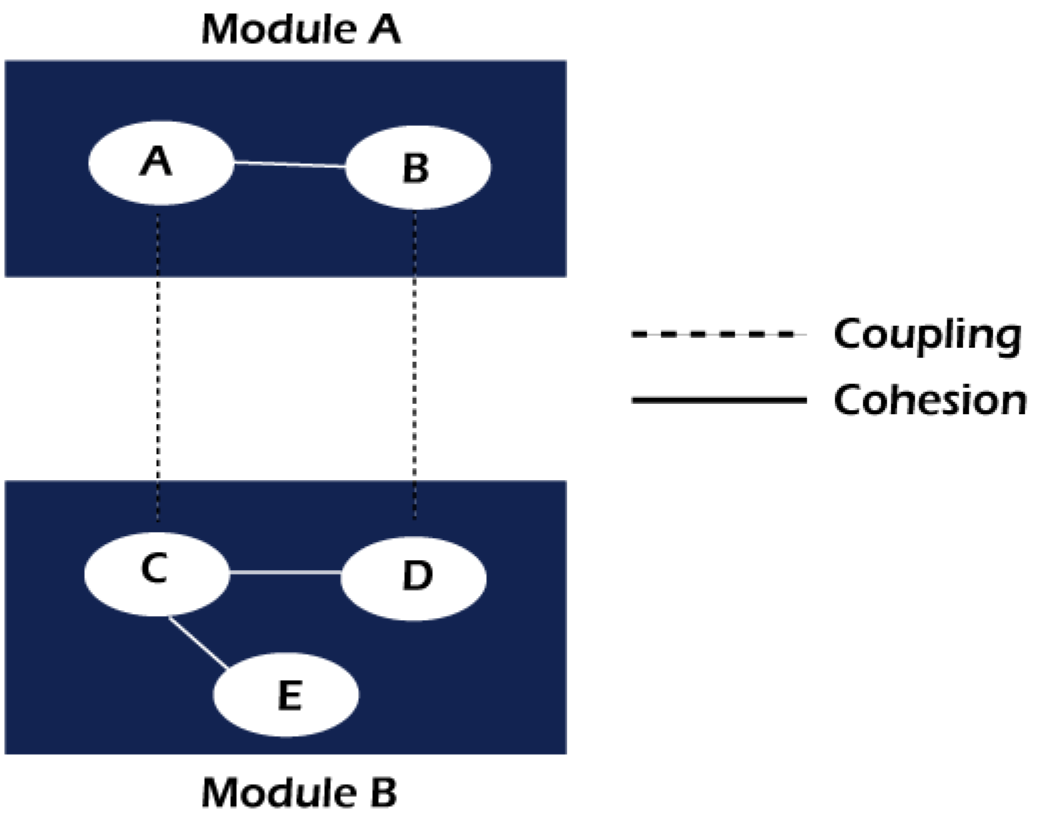
```
class Dog
{
public void bark()
{
System.out.println("woof ");
}
}
class Hound extends Dog
{
```

```java
public void sniff()
{
System.out.println("sniff ");
}
//overrides the method bark() of the Dog class
public void bark()
{
System.out.println("bowl");
}
}
public class OverridingExample
{
public static void main(String args[])
{
Dog dog = new Hound();
//invokes the bark() method of the Hound class
dog.bark();
}
}
```

32) What is Cohesion in OOP?

In OOP, **cohesion** refers to the degree to which the elements inside a module belong together. It measures the strength of the relationship between the module and data. In short, cohesion represents the clarity of the responsibilities of a module. It is often contrasted with coupling.

It focuses on a how single module or class is intended. Higher the cohesiveness of the module or class, better is the object-oriented design.



There are two types of cohesion, i.e. **High** and **Low**.

- High cohesion is associated with several required qualities of software including **robustness, reliability**, and **understandability**.
- Low cohesion is associated with unwanted qualities such as being difficult to **maintain, test, reuse,** or even **understand**.

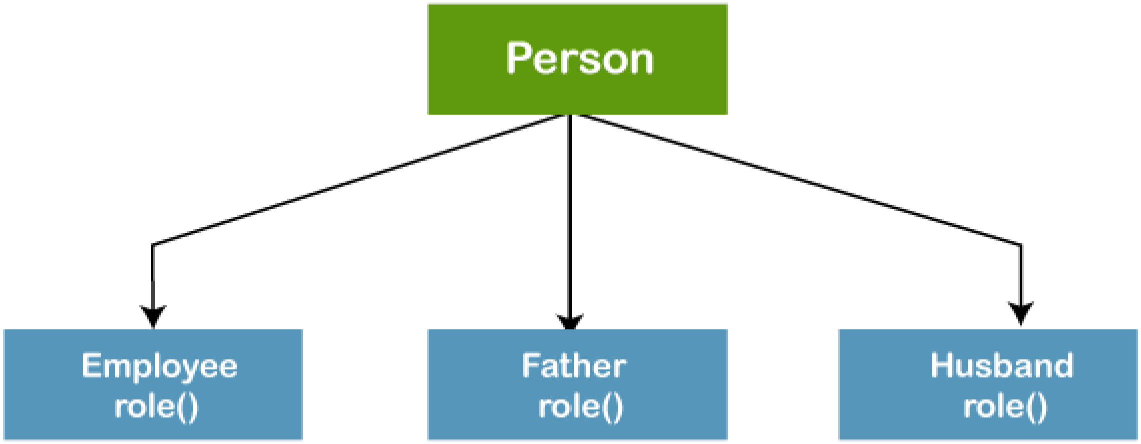
High cohesion often associates with loose coupling and vice versa.

33) Give a real-world example of polymorphism?

The general meaning of Polymorphism is one that has different forms. The best real-world example of polymorphism is a **person** that plays different roles at different palaces or situations.

- At home a person can play the role of father, husband, and son.
- At the office the same person plays the role of boss or employee.
- In public transport, he plays the role of passenger.
- In the hospital, he can play the role of doctor or patient.

- At the shop, he plays the role of customer.



Hence, the same person possesses different behavior in different situations. It is called polymorphism.

34) What is the difference between a base class and a superclass?

The base class is the root class- the most generalized class. At the same time, the superclass is the immediate parent class from which the other class inherits.

35) What is data abstraction and how can we achieve data abstraction?

It is one of the most important features of OOP. It allows us to show only essential data or information to the user and hides the implementation details from the user. A real-world example of abstraction is driving a car. When we drive a car, we do not need to know how the engine works (implementation) we only know how ECG works.

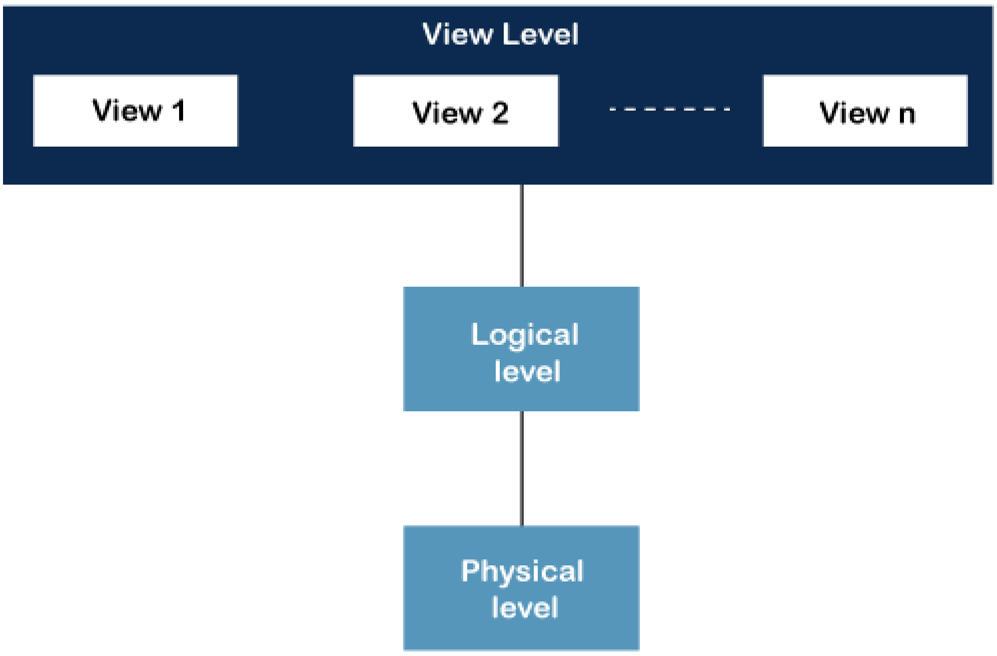
There are two ways to achieve data abstraction

- Abstract class
- Abstract method

36) What are the levels of data abstraction?

There are **three** levels of data abstraction:

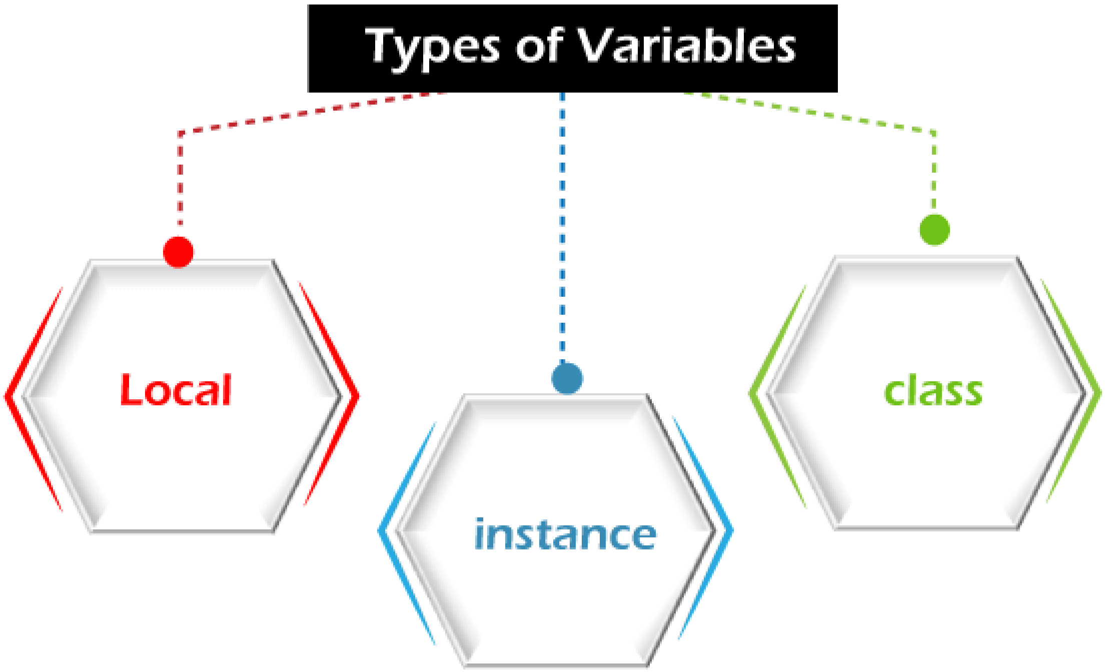
- **Physical Level:** It is the lowest level of data abstraction. It shows how the data is actually stored in memory.
- **Logical Level:** It includes the information that is actually stored in the database in the form of tables. It also stores the relationship among the data entities in relatively simple structures. At this level, the information available to the user at the view level is unknown.
- **View Level:** It is the highest level of data abstraction. The actual database is visible to the user. It exists to ease the availability of the database by an individual user.



Levels of data abstraction

37) What are the types of variables in OOP?

There are three types of variables:



Instance Variable: It is an object-level variable. It should be declared inside a class but must be outside a method, block, and constructor. It is created when an object is created by using the new keyword. It can be accessed directly by calling the variable name inside the class.

Static Variable: It is a class-level variable. It is declared with keyword **static** inside a class but must be outside of the method, block, and constructor. It stores in static memory. Its visibility is the same as the instance variable. The default value of a static variable is the same as the instance variable. It can be accessed by calling the **class_name.variable_name**.

Local Variable: It is a method-level variable. It can be declared in method, constructor, or block. Note that the use of an access modifier is not allowed with local variables. It is visible only to the method, block, and constructor in which it is declared. Internally, it is implemented at the stack level. It must be declared and initialized before use.

Another type of variable is used in object-oriented programming is the **reference** variable.

Reference Variable: It is a variable that points to an object of the class. It points to the location of the object that is stored in the memory.

38) Is it possible to overload a constructor?

Yes, the constructors can be overloaded by changing the number of arguments accepted by the constructor or by changing the data type of the parameters. For example:

```
public class Demo
{
Demo()
{
//logic
}
Demo(String str) //overloaded constructor
{
//logic
}
Demo(double d) //overloaded constructor
{
//logic
}
//statements
}
```

39) Can we overload the main() method in Java also give an example?

Yes, we can also overload the main() method in Java. Any number of main() methods can be defined in the class, but the method signature must be different. Consider the following code.

```
class OverloadMain
{
public static void main(int a)  //overloaded main method
{
System.out.println(a);
}
public static void main(String args[])
{
System.out.println("main method invoked");
main(6);
}
}
```

~Navnath

THANK YOU !!!