**Core Java Question**

**1.Java OOPs Concepts**

### **Inheritance**

When one object acquires all the properties and behaviors of a parent object, it is known as inheritance. It provides code reusability. It is used to achieve runtime polymorphism.

### **Polymorphism**

If one task is performed in different ways, it is known as polymorphism. For example: to convince the customer differently, to draw something, for example, shape, triangle, rectangle, etc .In Java, we use method overloading and method overriding to achieve polymorphism.

#### **Abstraction**

Hiding internal details and showing functionality is known as abstraction. For example, phone call, we don't know the internal processing. Hiding internal details and showing functionality is known as abstraction.

### **Encapsulation**

Binding (or wrapping) code and data together into a single unit are known as encapsulation. For example, a capsule, it is wrapped with different medicines. A java class is the example of encapsulation. Java bean is the fully encapsulated class because all the data members are private here.

**2.Can we overload the main method?**

Yes, we can overload the main method in java but JVM only calls the original main method, it will never call our overloaded main method.

As from above example, it is clear that every time original main method executes but not the overloaded methods because JVM only executes the original main method by default but not the overloaded one.

So, to execute overloaded methods of main, we must call them from the original main method.

**3. Can we execute a java program without a main method?**

Yes, we can execute a java program without a main method by using a static block.

Static block in Java is a group of statements that gets executed only once when the class is loaded into the memory by Java ClassLoader, It is also known as a static initialization block. Static initialization block is going directly into the stack memory.

**4.** **Can we declare a main method as private in Java?**

Yes, we can declare the main method as private in Java.

It compiles successfully without any errors but at the runtime, it says that the main method is not public.

**5.Can We declare main() method as Non-Static in java?**

The public static void main(String ar[]) method is the entry point of the execution in Java. When we run a .class file JVM searches for the main method and executes the contents of it line by line.

You can write the main method in your program without the static modifier, the program gets compiled without compilation errors.

But, at the time of execution JVM does not consider this new method (without static) as the entry point of the program. It searches for the main method which is public, static, with return type void, and a String array as an argument.

**6.Can we change the order of public static void main() to static public void main() in Java?**

Yes, we can change the order of public static void main() to static public void main() in Java, the compiler doesn't throw any compile-time or runtime error. In Java, we can declare access modifiers in any order, the method name comes last, the return type comes second to last and then after it's our choice. But it's recommended to put access modifier (public, private and protected) at the forefront as per Java coding standards.

**7.Why main() method must be static in java?**

Static − If you declare a method, subclass, block, or a variable static it is loaded along with the class.

In Java whenever we need to call an (instance) method we should instantiate the class (containing it) and call it. If we need to call a method without instantiation it should be static. Moreover, static methods are loaded into the memory along with the class.

The public static void main(String ar[]) method is the entry point of the execution in Java. When we run a .class file JVM searches for the main method and executes the contents of it line by line.

You can write the main method in your program without the static modifier, the program gets compiled without compilation errors. But, at the time of execution JVM does not consider this new method (without static) as the entry point of the program.

**8.Difference Between Checked and Unchecked Exception in Java**

Checked Exceptions

They occur at compile time.

The compiler checks for a checked exception.

These exceptions can be handled at the compilation time.

It is a sub-class of the exception class.

The JVM requires that the exception be caught and handled.

Example of Checked exception- ‘File Not Found Exception’

Unchecked Exceptions

These exceptions occur at runtime.

The compiler doesn’t check for these kinds of exceptions.

These kinds of exceptions can’t be caught or handled during compilation time.

This is because the exceptions are generated due to the mistakes in the program.

These are not a part of the ‘Exception’ class since they are runtime exceptions.

The JVM doesn’t require the exception to be caught and handled.

Example of Unchecked Exceptions- ‘No Such Element Exception'