TOPPERWORLD
LEARN & GROW

LEARN Java

through these

Most Asked Interview Question

www. TOPPERWORLD.in

Q. 1

Why is Java a platform independent language?

Ans.

Java language was developed so that it does not depend on any hardware or software because the compiler compiles the code and then converts it to platform-independent byte code which can be run on multiple systems.

The only condition to run that byte code is for the machine to have a runtime environment (JRE) installed in it.

Q. 2 Write a Java program to create and throw custom exceptions.

Ans.

```
class InterviewBit {
          public static void main(String args[]) throws
CustomException {
    throw new CustomException(" This is my custom Exception
");
  }
}
//Creating Custom Exception Class
class CustomException extends Exception{
  //Defining Constructor to throw exception message
  public CustomException(String message){
    super(message);
  }
}
```

Q. 3

Difference between Heap and Stack Memory in java. And how java utilizes this.

Ans.

Stack memory is the portion of memory that was assigned to every individual program. And it was fixed. On the other hand, Heap memory is the portion that was not allocated to the java program but it will be available for use by the java program when it is required, mostly during the runtime of the program.

Java Utilizes this memory as -

- When we write a java program then all the variables, methods, etc are stored in the stack memory.
- And when we create any object in the java program then that object was created in the heap memory. And it was referenced from the stack memory.

Q. 4 Can java be said to be the complete object-oriented programing language?

Ans.

It is not wrong if we claim that Java is the complete object-oriented programming language because everything in Java is under the classes and we can access them by creating the objects.

But we can even say that Java is not a completely object-oriented programming language because it has the support of primitive data types like int, float, char, boolean, double, etc.

Now for the question: Is Java a completely object-oriented programming language? We can say that - Java is not a pure object-oriented programming language, because it has direct access to primitive data types. And these primitive data types don't directly belong to the Integer classes.

Q. 5 How is Java different from C++?

Ans.

- C++ is only a compiled language, whereas Java is compiled as well as an interpreted language.
- Java programs are machine-independent whereas a c++ program can run only in the machine in which it is compiled.
- C++ allows users to use pointers in the program. Whereas java doesn’t allow it. Java internally uses pointers.
- C++ supports the concept of Multiple inheritances whereas Java doesn't support this. And it is due to avoiding the complexity of name ambiguity that causes the diamond problem.

Q. 6 Pointers are used in C/ C++. Why does Java not make use of pointers?

Ans.

Pointers are quite complicated and unsafe to use by beginner programmers. Java focuses on code simplicity, and the usage of pointers can make it challenging. Pointer utilization can also cause potential errors. Moreover, security is also compromised if pointers are used because the users can directly access memory with the help of pointers.

Thus, a certain level of abstraction is furnished by not including pointers in Java. Moreover, the usage of pointers can make the procedure of garbage collection quite slow and erroneous. Java makes use of references as these cannot be manipulated, unlike pointers.

Q. 7 What do you understand by an instance variable and a local variable?

Ans.

Instance variables are those variables that are accessible by all the methods in the class. They are declared outside the methods and inside the class. These variables describe the properties of an object and remain bound to it at any cost.

Local variables are those variables present within a block, function, or constructor and can be accessed only inside them. The utilization of the variable is restricted to the block scope. Whenever a local variable is declared inside a method, the other class methods don't have any knowledge about the local variable.

Q. 8 What are the default values assigned to variables and instances in java?

Ans.

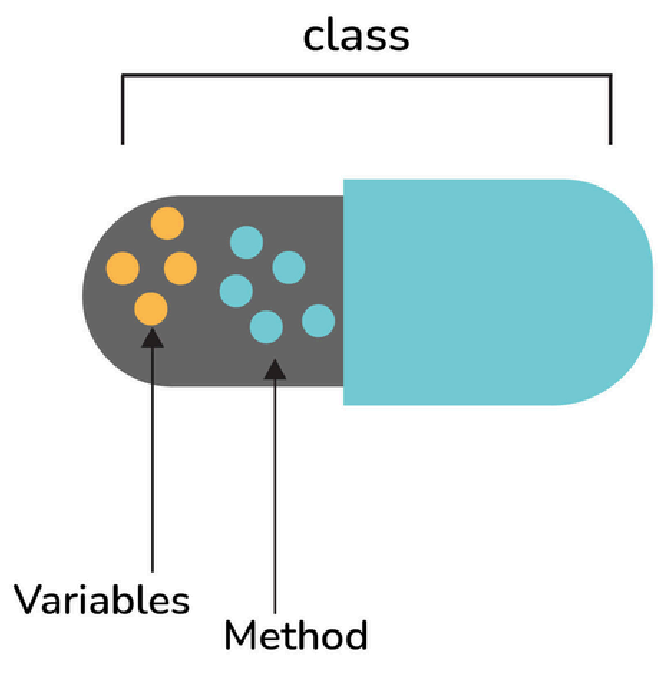
- There are no default values assigned to the variables in java. We need to initialize the value before using it. Otherwise, it will throw a compilation error of (Variable might not be initialized).
- But for instance, if we create the object, then the default value will be initialized by the default constructor depending on the data type.
- If it is a reference, then it will be assigned to null.
- If it is numeric, then it will assign to 0.
- If it is a boolean, then it will be assigned to false. Etc.

Q. 9 What do you mean by data encapsulation?

Ans.

Data Encapsulation is an Object-Oriented Programming concept of hiding the data attributes and their behaviours in a single unit. It helps developers to follow modularity while developing software by ensuring that each object is independent of other objects by having its own methods, attributes, and functionalities. It is used for the security of the private properties of an object and hence serves the purpose of data hiding.

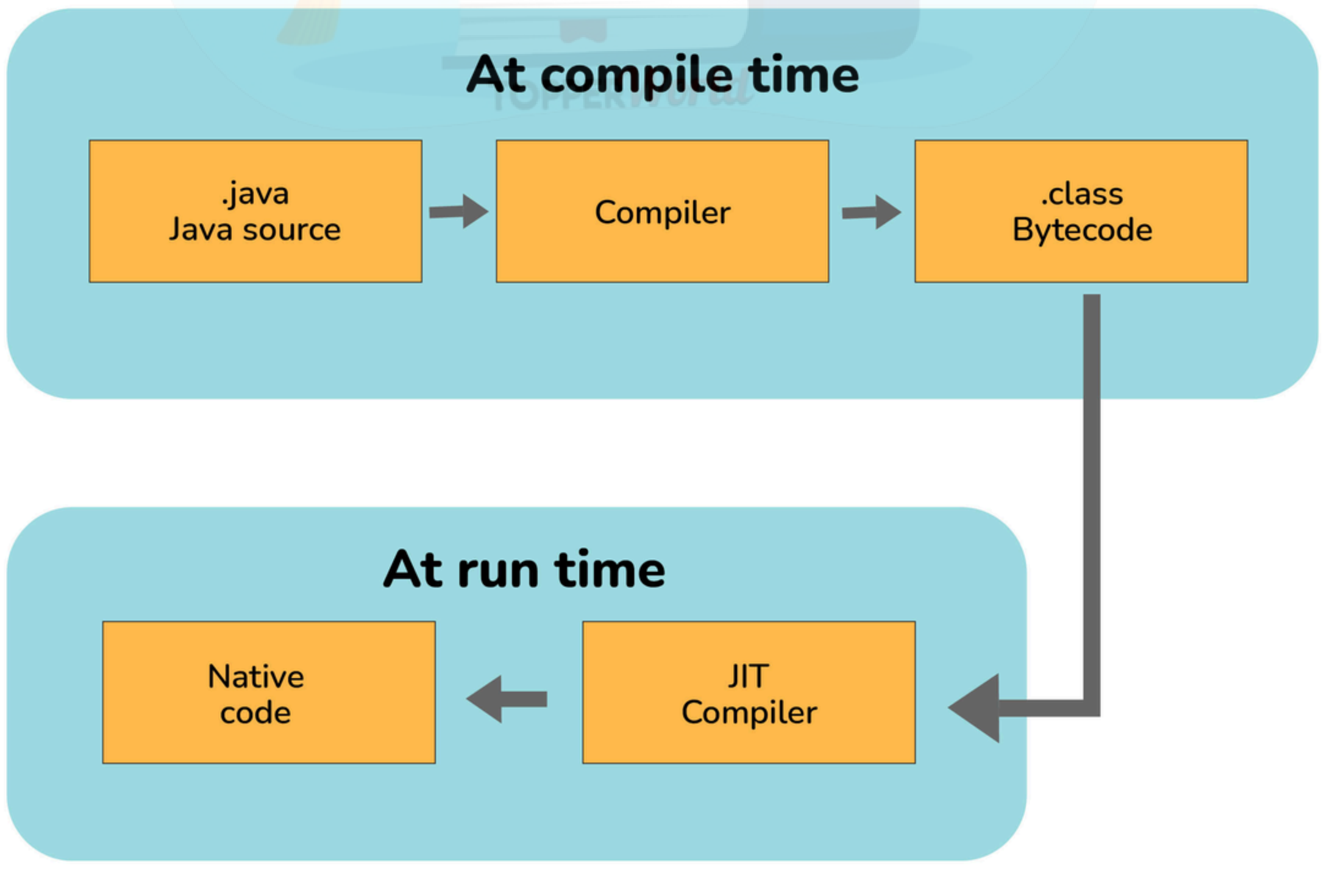
```
class
{

  data members (properties)
             +
   methods (behavior)

}
```



Q. 10 Tell us something about JIT compiler.

Ans.

- JIT stands for Just-In-Time and it is used for improving the performance during run time. It does the task of compiling parts of byte code having similar functionality at the same time thereby reducing the amount of compilation time for the code to run. The compiler is nothing but a translator of source code to machine-executable code.



Q. 11

Can you tell the difference between equals() method and equality operator (==) in Java?

Ans.

equals()	==
This is a method defined in the Object class.	It is a binary operator in Java.
The .equals() Method is present in the Object class, so we can override our custom .equals() method in the custom class, for objects comparison.	It cannot be modified. They always compare the HashCode.
This method is used for checking the equality of contents between two objects as per the specified business logic.	This operator is used for comparing addresses (or references), i.e checks if both the objects are pointing to the same memory location.

Q. 12 How is an infinite loop declared in Java?

Ans.

Using For Loop:

```
for (;;)
{
// Any break logic
}
```

Using while loop:

```
while(true)
{
// Any break logic
}
```

Using do-while loop:

```
do{
// Any break logic
}while(true);
```

Q. 13 Briefly explain the concept of constructor overloading

Ans.

Constructor overloading is the process of creating multiple constructors in the class consisting of the same name with a difference in the constructor parameters. Depending upon the number of parameters and their corresponding types, distinguishing of the different types of constructors is done by the compiler.

```
class Hospital {

    int variable1, variable2;
    double variable3;

    public Hospital(int doctors, int nurses) {
        variable1 = doctors;
        variable2 = nurses;
    }
    public Hospital(int doctors) {
        variable1 = doctors;
    }
    public Hospital(double salaries) {
        variable3 = salaries
    }
}
```

3 constructors overloaded with different parametes

Q. 14 Define Copy constructor in java.

Ans.

Copy Constructor is the constructor used when we want to initialize the value to the new object from the old object of the same class.

Example:

```
class InterviewBit{
  String department;
  String service;
  InterviewBit(InterviewBit ib){
     this.departments = ib.departments;
     this.services = ib.services;
  }
}
```

Q. 15 Can the main method be Overloaded?

Ans.

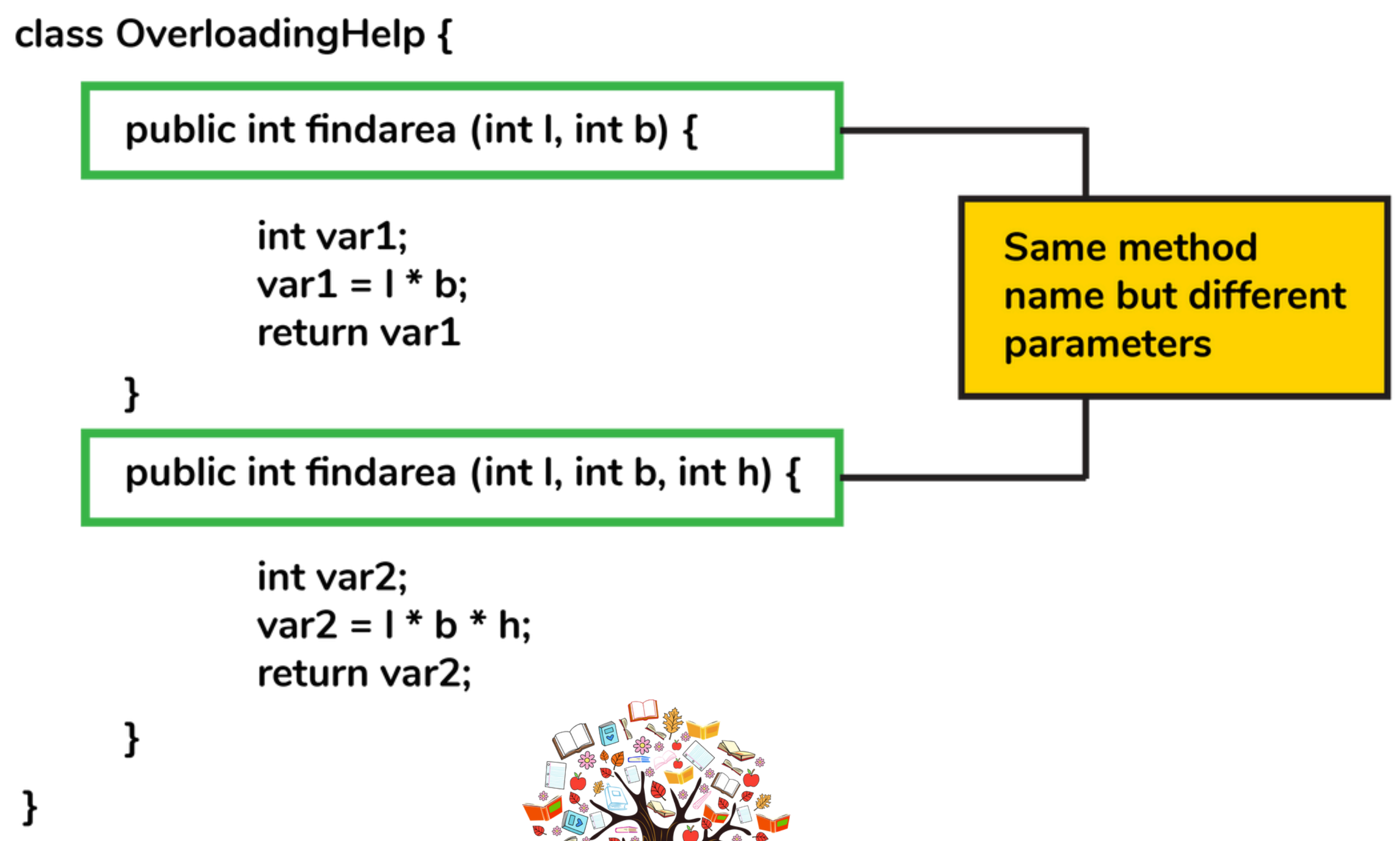
Yes, It is possible to overload the main method. We can create as many overloaded main methods we want. However, JVM has a predefined calling method that JVM will only call the main method with the definition of -

public static void main(string[] args)

Q. 16 Comment on method overloading and overriding by citing relevant examples.

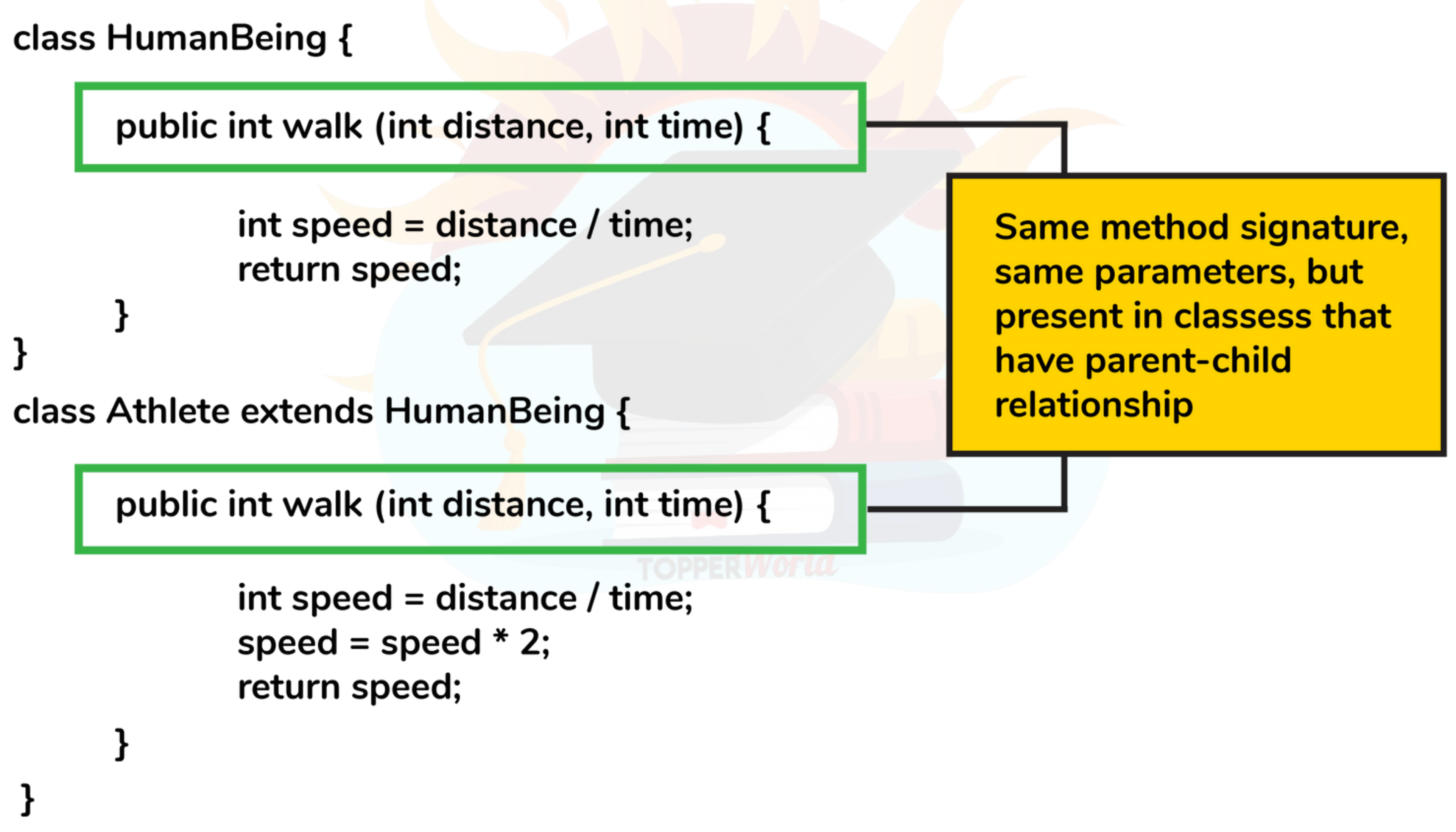
Ans.

In Java, **method overloading** is made possible by introducing different methods in the same class consisting of the same name. Still, all the functions differ in the number or type of parameters. It takes place inside a class and enhances program readability. The only difference in the return type of the method does not promote method overloading. The following example will furnish you with a clear picture of it.



```
class OverloadingHelp {
    public int findarea (int l, int b) {

            int var1;
            var1 = l * b;
            return var1
    }
    public int findarea (int l, int b, int h) {

            int var2;
            var2 = l * b * h;
            return var2;
    }
}
```

Method overriding is the concept in which two methods having the same method signature are present in two different classes in which an inheritance relationship is present. A particular method implementation (already present in the base class) is possible for the derived class by using method overriding.



```
class HumanBeing {
    public int walk (int distance, int time) {
            int speed = distance / time;
            return speed;
    }
}
class Athlete extends HumanBeing {
    public int walk (int distance, int time) {
            int speed = distance / time;
            speed = speed * 2;
            return speed;
    }
}
```

Same method signature, same parameters, but present in classess that have parent-child relationship

Q. 17 A single try block and multiple catch blocks can co-exist in a Java Program. Explain.

Ans.

Yes, multiple catch blocks can exist but specific approaches should come prior to the general approach because only the first catch block satisfying the catch condition is executed.

```
public class MultipleCatchExample {
    public static void main(String[] args) {
        try {
            System.out.println(new int[]{1, 2, 3}[5]);
            System.out.println(10 / 0);
        } catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("Array index out of bounds!");
        } catch (ArithmeticException e) {
            System.out.println("Arithmetic error!");
        }
```

Q. 18 Explain the use of final keyword in variable, method and class.

Ans.

final variable:

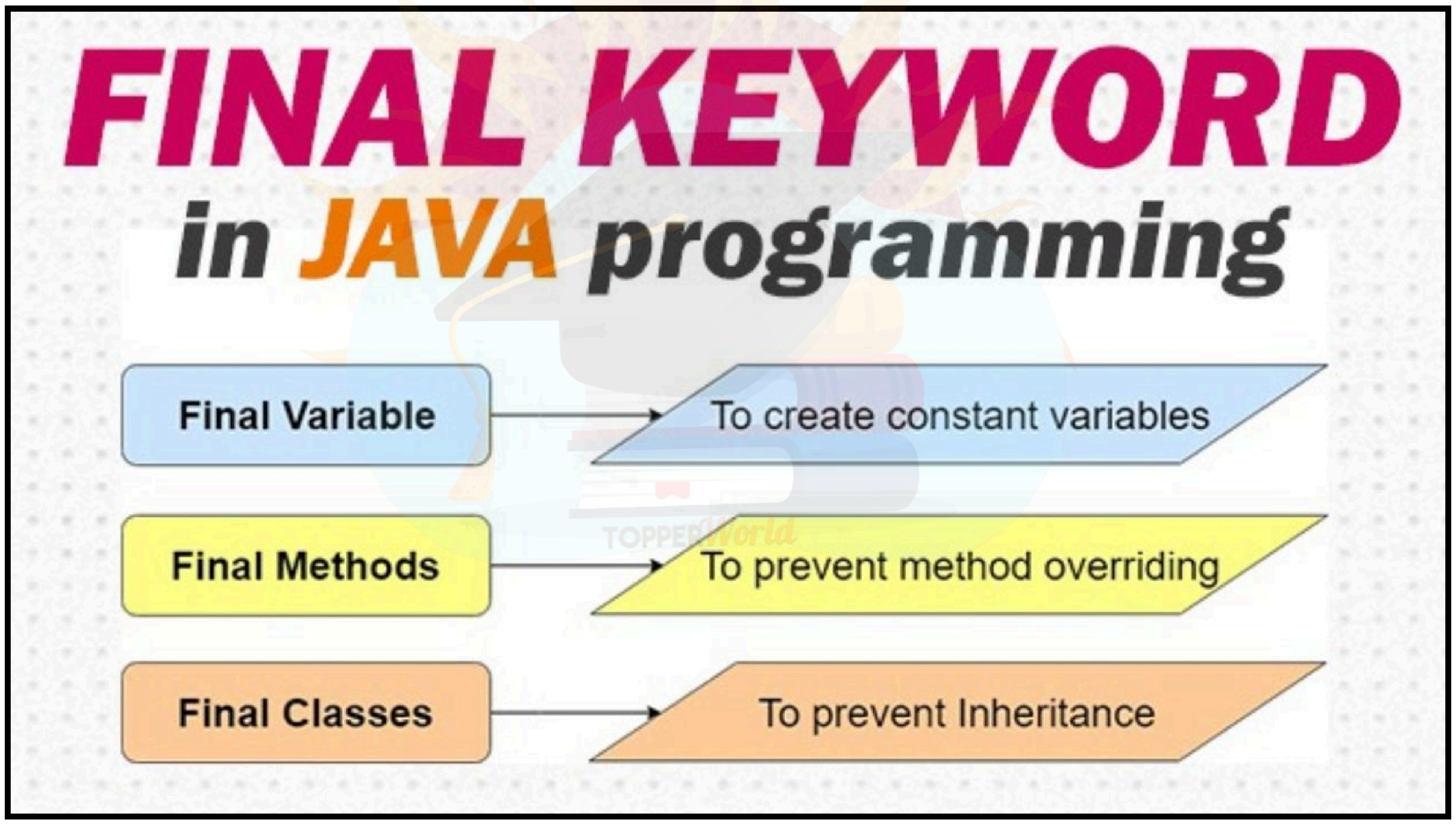
When a variable is declared as final in Java, the value can't be modified once it has been assigned. If any value has not been assigned to that variable, then it can be assigned only by the constructor of the class.

final method:

A method declared as final cannot be overridden by its children's classes. A constructor cannot be marked as final because whenever a class is inherited, the constructors are not inherited.

final class:

No classes can be inherited from the class declared as final. But that final class can extend other classes for its usage.



Q. 19 Do final, finally and finalize keywords have the same function?

Ans.

Final: If any restriction is required for classes, variables, or methods, the final keyword comes in handy. Inheritance of a final class and overriding of a final method is restricted by the use of the final keyword. The variable value becomes fixed after incorporating the final keyword.

Finally: It is the block present in a program where all the codes written inside it get executed irrespective of handling of exceptions.

Finalize: Prior to the garbage collection of an object, the finalize method is called so that the clean-up activity is implemented.

Q. 20 Write a Java Program to find the factorial of a given number.

Ans.

```
public class FindFactorial {
  public static void main(String[] args) {
     int num = 10;
     long factorialResult = 1l;
     for(int i = 1; i <= num; ++i)
     {
        factorialResult *= i;
     }
     System.out.println("Factorial: "+factorialResult);
  }
}
```

Q. 21 When can you use super keyword?

Ans.

- The super keyword is used to access hidden fields and overridden methods or attributes of the parent class.
- Following are the cases when this keyword can be used:
  - Accessing data members of parent class when the member names of the class and its child subclasses are same.
  - To call the default and parameterized constructor of the parent class inside the child class.
  - Accessing the parent class methods when the child classes have overridden them.

Q. 22 Why is the main method static in Java?

Ans.

The main method is always static because static members are those methods that belong to the classes, not to an individual object. So if the main method will not be static then for every object, It is available. And that is not acceptable by JVM. JVM calls the main method based on the class name itself. Not by creating the object.

Because there must be only 1 main method in the java program as the execution starts from the main method. So for this reason the main method is static.

Q. 23 Can the static methods be overridden?

Ans.

- No! Declaration of static methods having the same signature can be done in the subclass but run time polymorphism can not take place in such cases.
- Overriding or dynamic polymorphism occurs during the runtime, but the static methods are loaded and looked up at the compile time statically. Hence, these methods cant be overridden.

Q. 24 Difference between static methods, static variables, and static classes in java.

Ans.

Static Methods and Static variables are those methods and variables that belong to the class of the java program, not to the object of the class. This gets memory where the class is loaded. And these can directly be called with the help of class names.

For example - We have used mathematical functions in the java program like - max(), min(), sqrt(), pow(), etc. And if we notice that, then we will find that we call it directly with the class name. Like - Math.max(), Math.min(), etc.

Static classes - A class in the java program cannot be static except if it is the inner class. If it is an inner static class, then it exactly works like other static members of the class

Q. 25 What is a ClassLoader?

Ans.

Java Classloader is the program that belongs to JRE (Java Runtime Environment). The task of ClassLoader is to load the required classes and interfaces to the JVM when required.

Example- To get input from the console, we require the scanner class. And the Scanner class is loaded by the ClassLoader.

Q. 26 Using relevant properties highlight the differences between interfaces and abstract classes.

Ans.

Availability of methods: Only abstract methods are available in interfaces, whereas non-abstract methods can be present along with abstract methods in abstract classes.

Variable types: Static and final variables can only be declared in the case of interfaces, whereas abstract classes can also have non-static and non-final variables.

Inheritance: Multiple inheritances are facilitated by interfaces, whereas abstract classes do not promote multiple inheritances.

Data member accessibility: By default, the class data members of interfaces are of the public- type. Conversely, the class members for an abstract class can be protected or private also.

Q. 27 What is a Comparator in java?

Ans.

Consider the example where we have an ArrayList of employees like(EId, Ename, Salary), etc. Now if we want to sort this list of employees based on the names of employees. Then that is not possible to sort using the Collections.sort() method. We need to provide something to the sort() function depending on what values we have to perform sorting. Then in that case a comparator is used.

Comparator is the interface in java that contains the compare method. And by overloading the compare method, we can define that on what basis we need to compare the values.

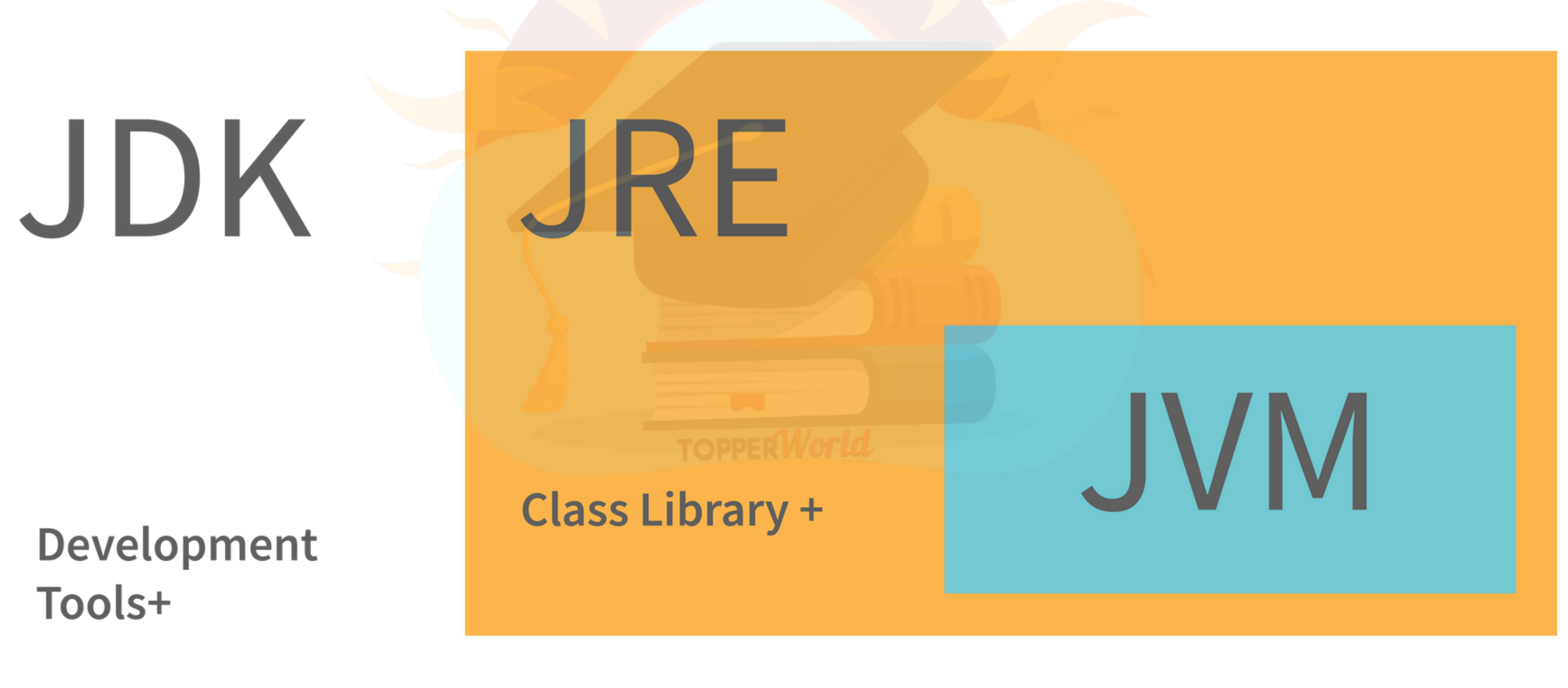
Q. 28 What do we get in the JDK file?

Ans.

JDK- For making java programs, we need some tools that are provided by JDK (Java Development Kit). JDK is the package that contains various tools, Compiler, Java Runtime Environment, etc.

JRE - To execute the java program we need an environment. (Java Runtime Environment) JRE contains a library of Java classes + JVM. **What are JAVA Classes?** It contains some predefined methods that help Java programs to use that feature, build and execute. For example - there is a system class in java that contains the print-stream method, and with the help of this, we can print something on the console.

JVM - (Java Virtual Machine) JVM is a part of JRE that executes the Java program at the end. Actually, it is part of JRE, but it is software that converts bytecode into machine-executable code to execute on hardware.



Q. 29

What are the differences between HashMap and HashTable in Java?

Ans.

HashMap	HashTable
HashMap is not synchronized thereby making it better for non-threaded applications.	HashTable is synchronized and hence it is suitable for threaded applications.
Allows only one null key but any number of null in the values.	This does not allow null in both keys or values.
Supports order of insertion by making use of its subclass LinkedHashMap.	Order of insertion is not guaranteed in HashTable.

Q. 30 What are the different types of Thread Priorities in Java? And what is the default priority of a thread assigned by JVM?

Ans.

There are a total of 3 different types of priority available in Java.

MIN_PRIORITY: It has an integer value assigned with 1.

MAX_PRIORITY: It has an integer value assigned with 10.

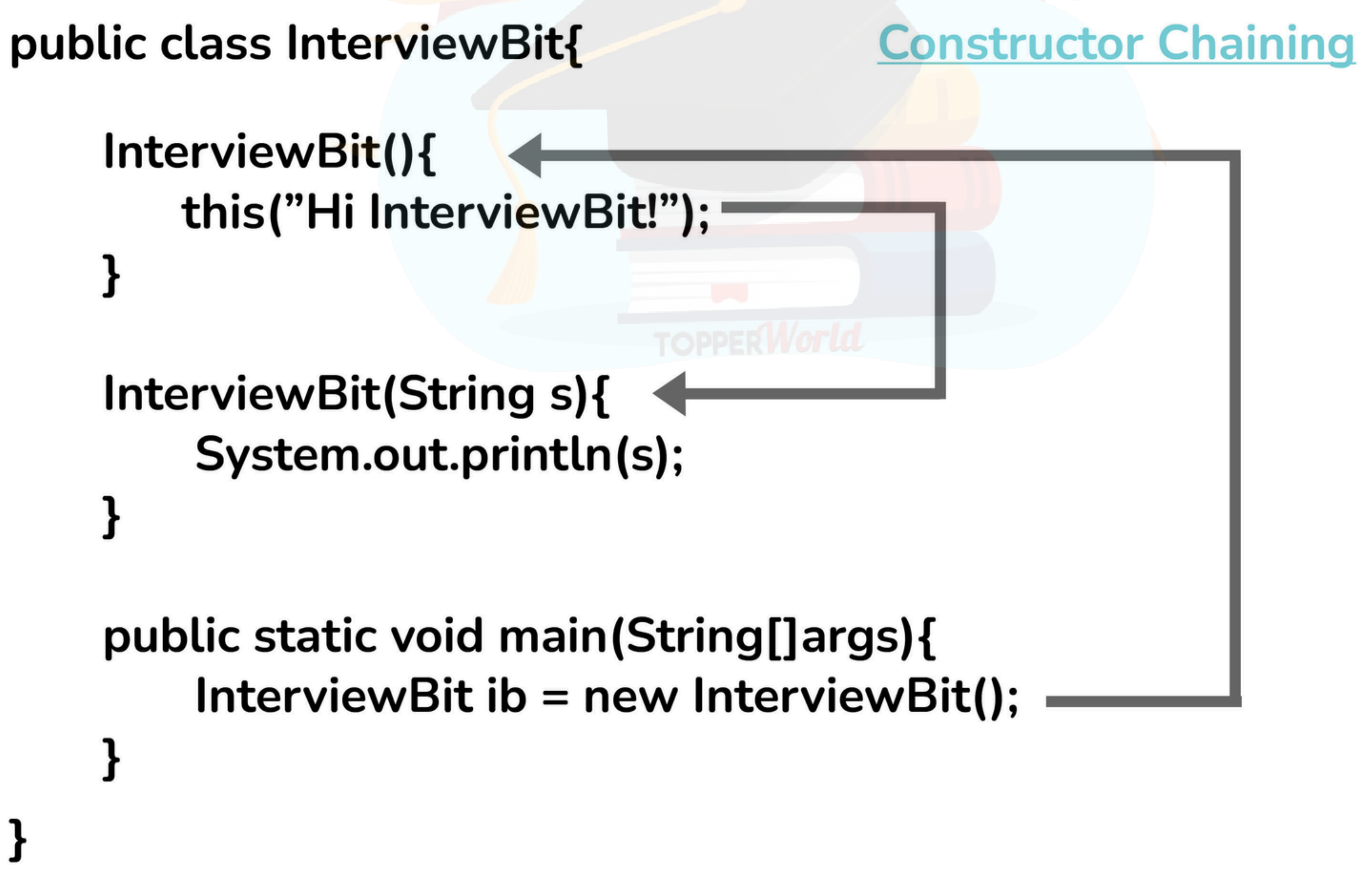
NORM_PRIORITY: It has an integer value assigned with 5.

In Java, Thread with MAX_PRIORITY gets the first chance to execute. But the default priority for any thread is NORM_PRIORITY assigned by JVM.

Q. 31 Can you call a constructor of a class inside the another constructor?

Ans.

Yes, the concept can be termed as constructor chaining and can be achieved using this().



```
public class InterviewBit{

    InterviewBit(){
        this("Hi InterviewBit!");
    }

    InterviewBit(String s){
        System.out.println(s);
    }

    public static void main(String[]args){
        InterviewBit ib = new InterviewBit();
    }
}
```

Q. 32 How is the 'new' operator different from the 'newInstance()' operator in java?

Ans.

Both **'new'** and **'newInstance()'** operators are used to creating objects. The difference is- that when we already know the class name for which we have to create the object then we use a new operator. But suppose we don't know the class name for which we need to create the object, Or we get the class name from the command line argument, or the database, or the file. Then in that case we use the 'newInstance()' operator.

The 'newInstance()' keyword throws an exception that we need to handle. It is because there are chances that the class definition doesn't exist, and we get the class name from runtime. So it will throw an exception.

Q. 33 Define System.out.println().

Ans.

System.out.println() is used to print the message on the console. System - It is a class present in java.lang package. Out is the static variable of type PrintStream class present in the System class. println() is the method present in the PrintStream class.

So if we justify the statement, then we can say that if we want to print anything on the console then we need to call the println() method that was present in PrintStream class. And we can call this using the output object that is present in the System class.

Q. 34 What is the best way to inject dependency? Also, state the reason.

Ans.

There is no boundation for using a particular dependency injection. But the recommended approach is -

Setters are mostly recommended for optional dependencies injection, and constructor arguments are recommended for mandatory ones. This is because constructor injection enables the injection of values into immutable fields and enables reading them more easily.

Q. 35

How we can set the spring bean scope. And what supported scopes does it have?

Ans.

A scope can be set by an annotation such as the @Scope annotation or the "scope" attribute in an XML configuration file.

Spring Bean supports the following five scopes:

- Singleton
- Prototype
- Request
- Session
- Global-session

"UNLOCK YOUR POTENTIAL"

With- TOPPERWORLD

Explore More

www. topperworld.in

DSA TUTORIAL

C TUTORIAL

C++ TUTORIAL

JAVA TUTORIAL

PYTHON TUTORIAL

www. TOPPERWORLD.in

E-mail

topperworld.in@gmail.com

Follow Us On