**Why is Java a platform independent language?**

Java language was developed in such a way that it does not depend on any hardware or software due to the fact that 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.

**2)Can java be said to be the complete object-oriented programming language?**

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 that by creating the objects.

But also if we 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.

**3) How is Java different from C++?**

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.

**4)Pointers are used in C/ C++. Why does Java not make use of pointers?**

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.

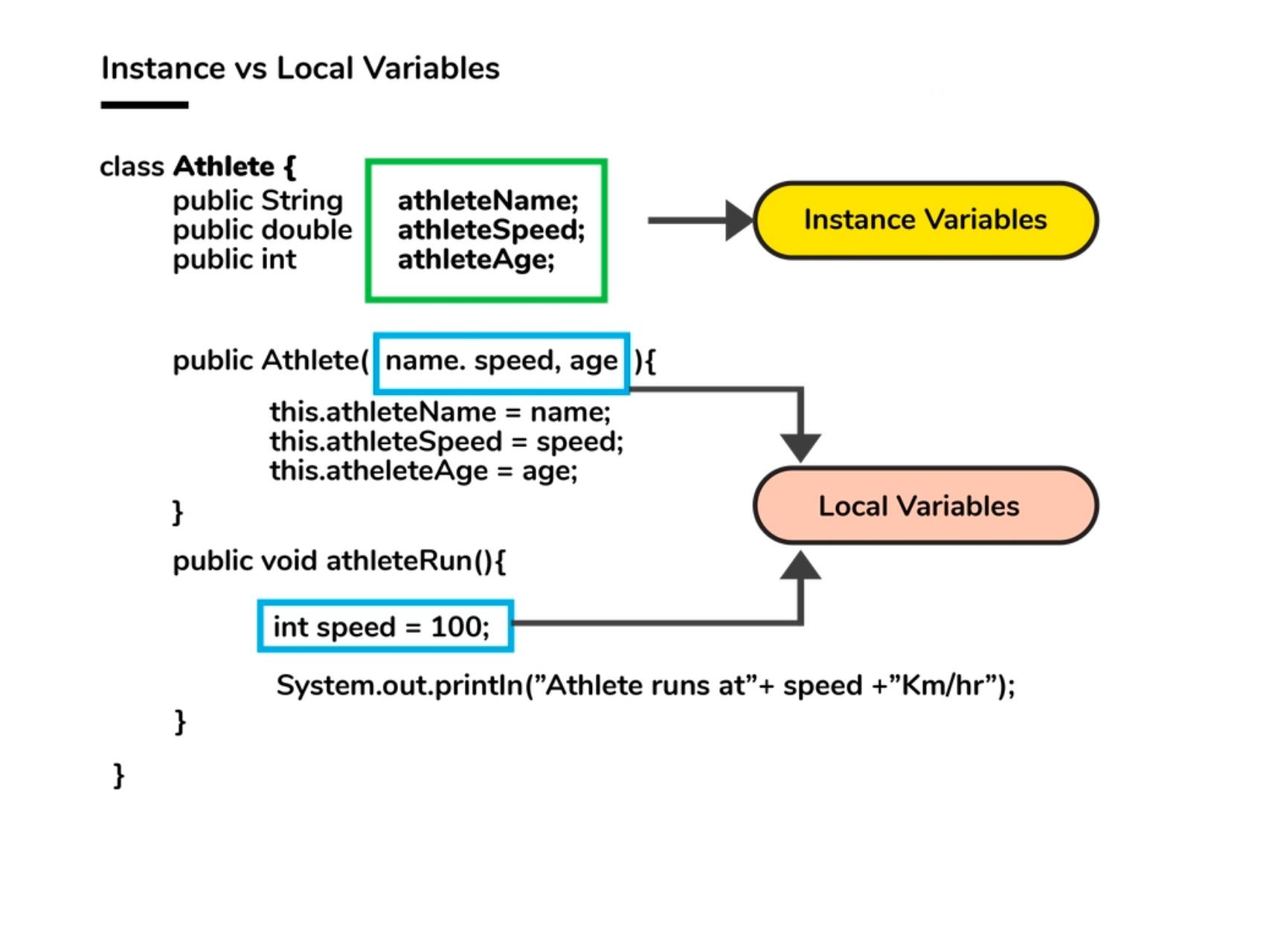
Thus, a certain level of abstraction is furnished by not including pointers in Java. Moreover, the usage of pointers can make the procedure of garbage collection quite slow and erroneous. Java makes use of **references** as these cannot be manipulated, unlike pointers.

**5)What do you understand by an instance variable and a local variable?**

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.

All the objects of the class will have their copy of the variables for utilization. If any modification is done on these variables, then only that instance will be impacted by it, and all other class instances continue to remain unaffected.

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.



**6)What are the default values assigned to variables and instances in java?**

There are no default values assigned to the variables(local) 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 (variable), if we create the object, then the default value will be initialized by the default constructor depending on the data type.

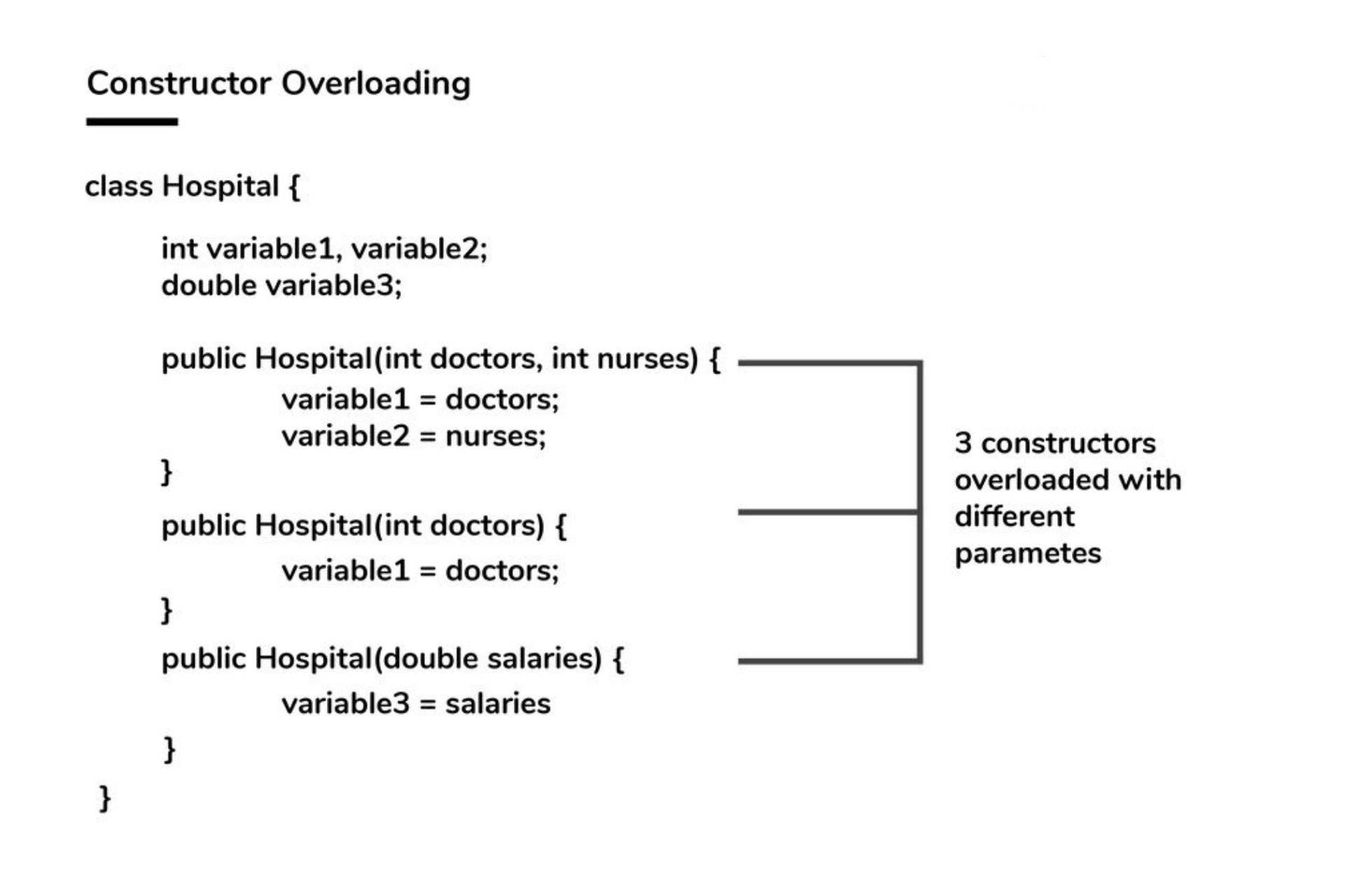
If it is a reference(string), then it will be assigned to null.

If it is numeric, then it will assign to 0.

If it is a boolean, then it will be assigned to false.

**7)Briefly explain the concept of constructor overloading**

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.



**Can the static methods be overloaded?**

Yes! There can be two or more static methods in a class with the same name but differing input parameters.

**9)Why is the main method static in Java?**

The main method is always static because static methods 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.

**10)What is a ClassLoader?**

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.

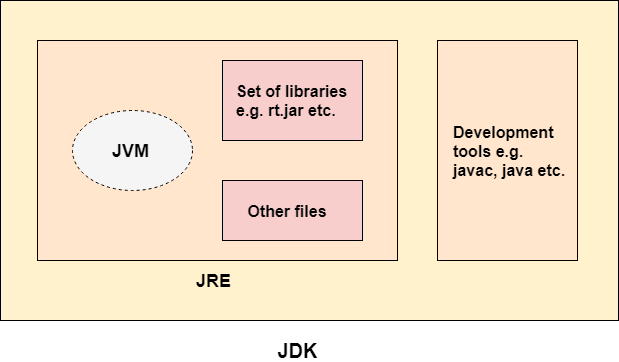
**Example**- To get input from the console, we require the scanner class. And the Scanner class is loaded by the ClassLoader.

**11)What are JDK JRE AND JVM?**

**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.

**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.



**12)What are the differences between constructor and method of a class in Java?**

1)Constructor has no return type. Method should have a return type. Even if it does not return anything, return type is void.

2)Constructor gets invoked implicitly. Method has to be invoked on the object explicitly.

3)Constructor is used for initializing the object state.

Method is used for exposing the object's behavior.

4)If the constructor is not defined, then a default constructor is provided by the java compiler.

If a method is not defined, then the compiler does not provide it.

5)The constructor name should be equal to the class name.

The name of the method can have any name or have a class name too.

6)A constructor cannot be marked as final because whenever a class is inherited, the constructors are not inherited. Hence, marking it final doesn't make sense. Java throws compilation error saying - modifier final not allowed here

A method can be defined as final but it cannot be overridden in its subclasses.

**13)What happens if the static modifier is not included in the main method signature in Java?**

There wouldn't be any compilation error. But then the program is run, since the JVM cant map the main method signature, the code throws “NoSuchMethodError” error at the runtime.

**14)Consider the below program, identify the output, and also state the reason for that.**

**class Adi**

{

public static void main(String[] args)

{

System.out.println(" Hello. Main Method. ");

}

public static void main(int[] args)

{

System.out.println(" Hello. Main Method2. ");

}

}

The output of the above program will be ***Hello. Main Method.***

This is because JVM will always call the main method based on the definition it already has. Doesn't matter how many main methods we overload it will only execute one main method based on its declaration in JVM.

15)