1. Why do we need static keyword in Java? Explain with example.

Ans: The static keyword mainly used for memory management in Java. A static keyword can be applied to variables, blocks, methods and classes. It makes our program more efficient, as every object doesn’t allocate separate memory to a static variable.

In this example we don’t need create an object to call the static function.

public class Question1 {

static void fun1(){

System.out.println("Static variable.");

}

void fun2(){

System.out.println("Non-static variable.");

}

public static void main(String []args){

Question1 obj = new Question1();

fun1();

obj.fun2();

}

}

1. What is class loading and how does the java program actually executes?

Ans: In Java, class loading is the process of loading class files into JVM (Java Virtual Machine) at runtime. It is responsible for loading classes from various sources, such as the file system, network and databases and making them available to the JVM for execution.

The class loading process in Java is divided into three phases: loading, linking and initialization.

1. Loading: In the loading phase, the class loader locates the class file using the fully qualified class name, reads the class file, and converts it into a class object. The class object contains metadata of the class such as fields, methods and constructors.
2. Linking: In the linking phase, JVM performs several operations on the class object, such as verifying the class file’s integrity, resolving symbolic references and allocating memory for the class variables.
3. Initialization: In the initialization phase, the JVM initializes the class variables with their values, and runs the class’s static initialization block (if any).
4. Can we mark a local variable as static.

Ans: No, because local variables are declared inside methods and can be inside the method only, but a static variable is a shared memory that can accessed from any instance. So, if we declare a local variable as static then it violates the property of static being shared memory.

1. Why is a static block executed before the main method in Java?

Ans: The static blocks always execute first before the main() method in Java because the compiler stores them in memory at the time of class loading and before the object creation. Here, the compiler executes all the static blocks first, and after finishing the static block execution, it invokes the main() method.

1. Why is a static method also called a class method?

Ans: A static method is a method that belongs to a class rather than an instance of a class. This means you can call a static method without creating an object of the class. Therefore, static method also called a class method.

1. What is the use of static blocks in java?

Ans: Static blocks is used to initialize static data members. It is used to initialize before the main method at the time of class loading. It gets executed only one when the class gets loaded.

1. Difference between static and instance variables.

Ans: Instance variables is essentially variables within a class that is not marked as static. It can only be used after an instance of the class has been made. This is because instance variable belongs to the object.

Static variables are those which belongs to class and you can access these members without instantiating the class.

1. Difference between static and non- static members.

Ans: Static:

* These members are called “class members”.
* These members will get memory in method area.
* The value and behaviour don’t change from object to object then we need to use static members.
* Inside a static area we can access static members only.
* Static members are created using “static” keyword.

Non-Static:

* These members are called “instance members”.
* These members will get memory in head area.
* The value and behaviour change from object to object then we need to use non-static members.
* Inside a non-static area we can access both static and non-static members.
* Non-static members are created without using “static” keyword.