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

Ans - The `static` keyword in Java is used to create class-level variables and methods that can be accessed without creating an instance of the class. It's associated with the class itself rather than with instances of the class. This is useful for situations where you want to have a common value or behavior that is shared among all instances of the class.

Example:

class Example {

static int add(int a, int b) {

return a + b;

}

}

public class Main {

public static void main(String[] args) {

int sum = Example.add(5, 7); // No need to create an instance

System.out.println(sum); // Output: 12

}

}

2. What is class loading and how does the Java program actually execute?

Ans - Class loading is the process of loading bytecode (compiled Java code) into memory so that the Java Virtual Machine (JVM) can execute it. When a Java program is run, the classes it uses are not loaded all at once but are loaded as needed. The process involves three steps: loading, linking, and initialization. Loading involves finding and loading the bytecode of the class, linking involves verification, preparation (memory allocation for static variables), and resolution of symbolic references, and finally, initialization involves executing the class's static initializer blocks and initializing static fields.

3. Can we mark a local variable as static?

Ans - No, you cannot mark a local variable as `static`. The `static` keyword is used to define class-level members, and local variables are limited in scope to the block in which they are declared. Making a local variable `static` would go against its purpose as a local, temporary storage location.

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

Ans - In Java, when a class is loaded by the class loader, its static blocks are executed before the `main` method is invoked. This is because static blocks are meant to perform class-level initialization tasks before any instance of the class is created or any static method is called. It ensures that any necessary setup is done before the program execution begins.

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

Ans - A static method is also called a class method because it is associated with the class itself rather than with instances of the class. Since the method is not tied to a specific instance, it can be invoked using the class name. It can also access only static members of the class unless explicitly provided an instance to work with.

6. What is the use of static blocks in Java?

Ans - Static blocks in Java are used to perform class-level initialization tasks. They are executed when the class is loaded by the class loader, and they can be used to initialize static variables or perform other operations that need to be done before any instance of the class is created or any static methods are called.

7. Difference between Static and Instance variables:

Ans –

**Static variables:** These are associated with the class itself rather than with instances. They are shared among all instances of the class and can be accessed using the class name. They are initialized only once when the class is loaded.

**Instance variables**: These belong to individual instances of the class. Each instance has its own copy of instance variables. They are initialized when an object is created.

8. Difference between Static and Non-static (Instance) members:

**Static members:** These belong to the class itself, and there's only one copy of them shared among all instances. They are accessed using the class name.

**Non-static (Instance) members:** These belong to individual instances of the class. Each instance has its own set of non-static members. They are accessed using an instance of the class.