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

The `static` keyword is used to declare class-level members that belong to the class itself, rather than any instance of the class. This means you can access these members without creating an instance of the class.

Example:

```
public class StaticExample {
    static int count = 0; // Static variable
    static void increment() { // Static method
        count++;
    }
    public static void main(String[] args) {
        StaticExample.increment();
        System.out.println("Count: " + StaticExample.count);
    }
}
```

2. What is Class Loading and How Does the Java Program Actually Execute:

Class loading is the process where the Java Virtual Machine (JVM) loads, verifies, and initializes classes at runtime. The steps are:

- **Loading:** The JVM loads the class file into memory.
- **Linking:** The JVM verifies and prepares the class, including resolving references.
- **Initialization:** The JVM initializes static variables and executes static blocks.

The Java program execution starts with the `main` method.

3. Can We Mark a Local Variable as `static`:

No, local variables cannot be marked as `static`. Static is used for class-level variables and methods, not for local variables.

4. Why is the Static Block Executed Before the Main Method in Java:

Static blocks are executed when the class is first loaded into memory, before any objects are created and before the `main` method is executed. They initialize static variables and perform one-time setup.

Example:

```
public class StaticBlockExample {
    static {
        System.out.println("Static block executed.");
    }
    public static void main(String[] args) {
        System.out.println("Main method executed.");
    }
}
```

5. Why is a Static Method Also Called a Class Method:

A static method is called a class method because it belongs to the class itself rather than to any instance of the class. You can call it using the class name.

Example:

```
public class ClassMethodExample {
    static void display() {
        System.out.println("Static method called.");
    }

    public static void main(String[] args) {
        ClassMethodExample.display();
    }
```

}

6. What is the Use of Static Blocks in Java:

Static blocks are used for initializing static variables and performing setup tasks when the class is first loaded. They run once per class load.

Example:

```
public class StaticBlockUsage {
    static int value;
    static {
        value = 10;
        System.out.println("Static block initialized.");
    }
    public static void main(String[] args) {
        System.out.println("Value: " + StaticBlockUsage.value);
    }
}
```

7. Difference Between Static and Instance Variables:

- **Static Variables:**
  - Belong to the class.
  - Shared among all instances.
  - Initialized once when the class is loaded.
  - Accessed using the class name.

- **Instance Variables:**
  - Belong to instances of the class.

- Each instance has its own copy.
- Initialized when an object is created.
- Accessed through object references.

8. Difference Between Static and Non-Static (Instance) Members:

- **Static Members:**
  - Belong to the class.
  - Shared among all instances.
  - Accessed using the class name.
  - Cannot directly access non-static members.

- **Non-Static (Instance) Members:**
  - Belong to instances of the class.
  - Each instance has its own copy.
  - Accessed through object references.
  - Can access both static and non-static members.