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

- The *static* keyword in Java is used to indicate that a particular member belongs to the class itself, rather than to instances of the class. This means that static members are shared among all instances of the class. For example:

```
class Example {
    static int count = 0;
    Example() {
        count++;
    }
    static void displayCount() {
        System.out.println("Count: " + count);
    }
}
public class Main {
    public static void main(String[] args) {
        Example obj1 = new Example();
        Example obj2 = new Example();
        Example.displayCount(); // Output: Count: 2
    }
}
```

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

- Class loading is the process by which the Java Virtual Machine (JVM) loads classes into memory. The JVM uses a class loader to load classes. The execution of a Java program involves several steps: loading, linking, and initializing classes, followed by executing the *main* method.

3. **Can we mark a local variable as static?**

- No, local variables cannot be marked as *static*. The *static* keyword is only applicable to class-level variables and methods.

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

- Static blocks are executed when the class is loaded into memory, before any instances of the class are created and before the *main* method is executed. This is useful for initializing static variables.

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 specific instance of the class. It can be called without creating an instance of the class.

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

- Static blocks are used for static initialization of a class. They are executed when the class is loaded and can be used to initialize static variables or perform other static setup tasks.

7. **Difference between Static and Instance Variables**

- Static variables are shared among all instances of a class, while instance variables are unique to each instance. Static variables are declared with the *static* keyword, whereas instance variables are not.

8. **Difference between static and non-static members**

- Static members belong to the class and can be accessed without creating an instance of the class. Non-static members belong to instances of the class and require an instance to be accessed.