Q: Why do we declare the main method as static in Java? Explain with an example.

Answer: The main method is declared as static so the JVM can call it without needing to create an instance of the class. This is essential because the JVM needs an entry point to start executing a Java program.

Example:

java
Copy code
```java
public class Example {
    public static void main(String[] args) {
        System.out.println("Hello, World!");
    }
}
```

Here, the `main` method is static, allowing the JVM to invoke it directly using `Example.main()`.

Q: What is class loading and how does the Java program actually execute it?

Answer: Class loading is the process by which the JVM loads classes into memory when required during the execution of a Java program. The JVM uses class loaders to load class files. The main steps are:

1. Loading: The class loader reads the .class file.
2. Linking: Verifies the bytecode and prepares it for execution.
3. Initialization: Executes static blocks and initializes static variables.

Q: Can we mark a local variable as static?

Answer: No, local variables cannot be marked as static. Static variables belong to the class and have a class-level scope, whereas local variables are confined to the method scope.

Q: Why is the static block executed before the main method in Java?

Answer: Static blocks are executed when the class is loaded into memory, before any instance of the class is created or any static method is called, including the main method. This ensures that any static initialization needed for the class is done before the main method starts executing.

Q: Why is a static method also called a class method?

Answer: A static method is called a class method because it belongs to the class itself rather than any instance of the class. It can be called using the class name and does not require an object of the class.

Q: What is the use of static blocks in Java?

Answer: Static blocks are used for initializing static variables or executing code that needs to run once, such as setting up static resources or logging configuration, when the class is first loaded.

Example:

java
Copy code
```
public class Example {
    static {
        System.out.println("Static block executed");
    }

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

Q: Difference between static and instance variables?

Answer:

- **Static Variables**:
    - Belong to the class.
    - Shared among all instances.
    - Declared using the `static` keyword.
- **Instance Variables**:
    - Belong to instances (objects).
    - Each object has its own copy.
    - No `static` keyword.

Example:

java

Copy code
```
public class Example {
    static int staticVar;
    int instanceVar;
}
```

Q: Difference between static and non-static methods?

Answer:

- **Static Methods**:
    - Belong to the class.
    - Can be called without an instance.
    - Cannot access instance variables/methods directly.
- **Non-Static Methods**:
    - Belong to instances.
    - Need an instance to be called.
    - Can access instance and static variables/methods.

Example:

java
Copy code
```
public class Example {
    static void static method() {
        System.out.println("Static method");
    }

    void instanceMethod() {
        System.out.println("Instance method");
    }
}
```