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

   The static keyword in Java is used to indicate that a particular attribute or method belongs to the class rather than to any instance of the class. This means the field, method, or block can be accessed without creating an object of the class.

   - The static members are shared across all instances of a class.
   - Static members can be accessed without creating an instance of the class.

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

   Class loading is the process where the Java Virtual Machine (JVM) loads a class into memory when it is needed during program execution. Here's how a Java program executes:

   - Compilation: The Java source file (.java) is compiled into bytecode (.class file).
   - Class Loading: When a class is first used, the JVM loads it into memory using a class loader.
   - Initialization: The JVM runs static blocks and initializes static variables.
   - Execution: The main method is called to start the program's execution.

3. Can we mark a local variable as static?

   Local variables cannot be marked as static. The static keyword is used to define class-level members that are shared across all instances of the class, while local variables are method-specific and do not have a class context.

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

   The static block is executed before the main method because it is a part of the class initialization process. When a class is loaded into memory, any static blocks are executed first to initialize static variables and perform other setup tasks, before the main method is executed.

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

   A static method is called a class method because it is associated with the class itself, rather than an instance of the class. It can be called using the class name without creating an object, and it can only access static members of the class.

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

   Static blocks are used for initializing static variables or for performing one-time setup tasks that need to be done before any method is called or any instance is created. They are executed only once when the class is loaded.

```
public class StaticBlockExample {
    static int count;

    static {
        count = 10;  // Static block to initialize static variable
        System.out.println("Static block executed");
    }

    public static void main(String[] args) {
        System.out.println("Count: " + count);  // Output: Count: 10
    }
}
```

7. Difference between Static and Instance variable.
   or
8. Difference between static and non static members

   Static

   - Static variables are associated with the class itself.
   - Static variables are allocated memory once when the class is loaded, and they are shared

- Static variables exist as long as the class is loaded in memory.
- Static variables can be accessed using the class name, without creating an object.

Instance / Non-static

- Instance variables are associated with individual objects of the class.
- Instance variables are allocated memory for each object created from the class, and each object has its own copy.
- Instance variables exist as long as the object exists, i.e., they are destroyed when the object is garbage collected.
- Instance variables require an object to be accessed, as they belong to a specific instance of the class.