1. The "static" keyword in Java is used to define class-level variables and methods. These are associated with the class itself, rather than instances of the class. For example:

java

class Example {

static int count = 0;

public Example() {

count++;

}

}

In this example, the "count" variable is static, and it's shared among all instances of the "Example" class.

2. Class loading is the process by which the Java Virtual Machine (JVM) loads classes into memory as they are needed during program execution. It involves locating and loading the bytecode of a class. Java executes a program by invoking the "main" method of the main class, and as classes are used, they are loaded into memory.

3. No, you cannot mark a local variable as static. Static variables are associated with the class itself, while local variables are scoped to a specific method or block and have a limited lifetime within that scope.

4. In Java, the static block is executed before the "main" method to perform any initialization tasks that are required before the program starts. This ensures that any static variables or configurations are set up before the main execution begins.

5. A static method is also called a class method because it is associated with the class itself, rather than a specific instance of the class. You can call a static method on the class itself, without needing to create an instance of the class.

6. Static blocks in Java are used for class-level initialization tasks. They are executed when the class is loaded into memory and are typically used for tasks like setting up static variables or performing one-time configuration.

7. The main difference between static and instance variables is their scope and association:

- Static variables (class variables) are associated with the class itself and are shared among all instances of that class.

- Instance variables are associated with a specific instance of the class and have a separate copy for each instance.

8. The key differences between static and non-static (instance) members in Java:

- Static members are associated with the class itself, while non-static members are associated with instances of the class.

- Static members are accessed using the class name, while non-static members are accessed using object references.

- Static members exist for the entire lifetime of the program, while non-static members are created when instances of the class are created and destroyed when the instance is no longer in scope.