**What are types of variable in java?**

**Division 1:**

**Primitive variable**

**Reference variable**

**Division 2:**

**Instance variable**

**Static variable**

**Local variable**

**Where instance variable will be stored and how we can acess in both static and instance area?**

**Instance variable will be stored in heap area.**

**Refer InstanceVariableTest.java for more details**

**Would JVM provide any default value for instance variable?**

**Yes, JVM always provides some default value for instance variables.**

**Refer InstanceVariableTest11.java for more details**

**What is static variable and where it will be declared and when it will be loaded and destroyed?**

Static variables should be declared within the class directly but outside of any method or block or constructor.

It will be created at the time of class loading and destroyed at the time of class unloading, hence scope of static variable is exactly same as scope of .class file.

**Where static variable will be stored?**

In method area(Local variable will be stored in stack memory).

For static variables JVM will provide default values and we are not required to perform initializaing explicitly.

**Refer StaticVariableTest.java for more details**

**What is local variables?**

Sometimes to meet temporary requirements of the programmer, we can declare variables inside a method, inside a block or inside a constructor are called local variable.

Stored in stack memory.

**Would JVM initialize or provide default value to local variable?**

Nope, JVM won’t provide default value for local variable and also if we are not-using the local variable then no-issues.

Refer **LocalVariableTest.java** for more details

It is highly recommended to perform initialization of local variable at the time of declaration atleast with default values.

**Which modifier is applicable for local variable?**

**Only final modifier**

**Which variables are thread-safe and which variables are not?**

Local variables are thread-safe, instance and static variables can be accessed by multiple threads and hence these are not thread safe.