CORE JAVA ASSIGNMENT
- Submitted by (Kavita Goodwani INT 438)

Q1 Create an object of Singleton Class and make it thread safe. Explain how can you use Double Check locking

Ans.

File included: Singleton_Class.java

In object-oriented programming, a singleton class is a class that can have only one object (an instance of the class) at a time.
After first time, if we try to instantiate the Singleton class, the new variable also points to the first instance created. So whatever modifications we do to any variable inside the class through any instance, it affects the variable of the single instance created and is visible if we access that variable through any variable of that class type defined.

To design a singleton class:

1. Make constructor as private.
2. Write a static method that has return type object of this singleton class. Here, the concept of Lazy Initialisation is used to write this static method.

Double checked locking of Singleton

Double checked locking of Singleton is a way to ensure only one instance of Singleton class is created through an application life cycle. As the name suggests, in double-checked locking, code checks for an existing instance of Singleton class twice with and without locking to double ensure that no more than one instance of singleton gets created.

```
 public static Singleton getInstanceDC()
{ if (single_instance == null)
{ synchronized (Singleton.class)
{ if (single_instance == null)
{ single_instance = new Singleton();
} } }
```

```
run:
String from Singleton Class Object x is FIRST STRING
String from Singleton Class Object y is FIRST STRING
String from Singleton Class Object z is FIRST STRING


String from Singleton Class Object x is first string
String from Singleton Class Object y is first string
String from Singleton Class Object z is first string
BUILD SUCCESSFUL (total time: 2 seconds)
```

Q2. Count number of times each string has occurred in an Array of Strings

Ans File included: Count_Strings.java

```
run:
Enter the number of strings you wish to enter
3
Enter the Strings
Kavita
You
Kavita
The String You has occured 1 times
The String Kavita has occured 2 times
BUILD SUCCESSFUL (total time: 14 seconds)
```

Q3. Create an interface telephone_model
telephone implements the interface and provide three functionalities:

- **Making a call**
- **Terminating a call**
- **Redialling the last call**
- **Showing dialled call history**

File included: Main.java