**Q.3.Singleton Class**

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.

The Singleton's purpose is to control object creation, limiting the number of objects to only one. Since there is only one Singleton instance, any instance fields of a Singleton will occur only once per class, just like static fields. Singletons often control access to resources, such as database connections or sockets

Here's how we can use singletons in Java.

1.create a private constructor that restricts to create an object outside of the class

2.create a private attribute that refers to the singleton object.

3.create a public static method that allows us to create and access the object we created. Inside the method, we will create a condition that restricts us from creating more than one object.

### Difference between Normal class and Singleton class

1.To instantiate a normal class, we use a [java constructor](https://techvidvan.com/tutorials/java-constructor/). On the other hand, to instantiate a singleton class, we use the getInstance() method.

2.The other difference is that a normal class vanishes at the end of the lifecycle of the application while the singleton class does not destroy with the completion of an application.