Singleton Pattern

The **Singleton Pattern** is a **creational pattern** that ensures there is only one instance of a class at any point of time. It provides a global access point to that instance. This is useful in cases where we need to control access to a certain resource.

There are two parts to the singleton pattern. First, we need to make the constructor private so that more instances of the class cannot be created. Second, we need to a way to access the single instance of the object that exists. This needs to be a static method since we cannot use any objects to access this method.

public class SingleObject {  
 private static SingleObject *instance* = new SingleObject();  
 private SingleObject() {}  
 public static SingleObject getInstance() {  
 return *instance*;  
 }  
}  
  
public class SingletonPatternDemo {  
 public static void main(String[] args) {  
 SingleObject object = SingleObject.*getInstance*();  
 }  
}

JAVA