SENG 301 Software Analysis and Design
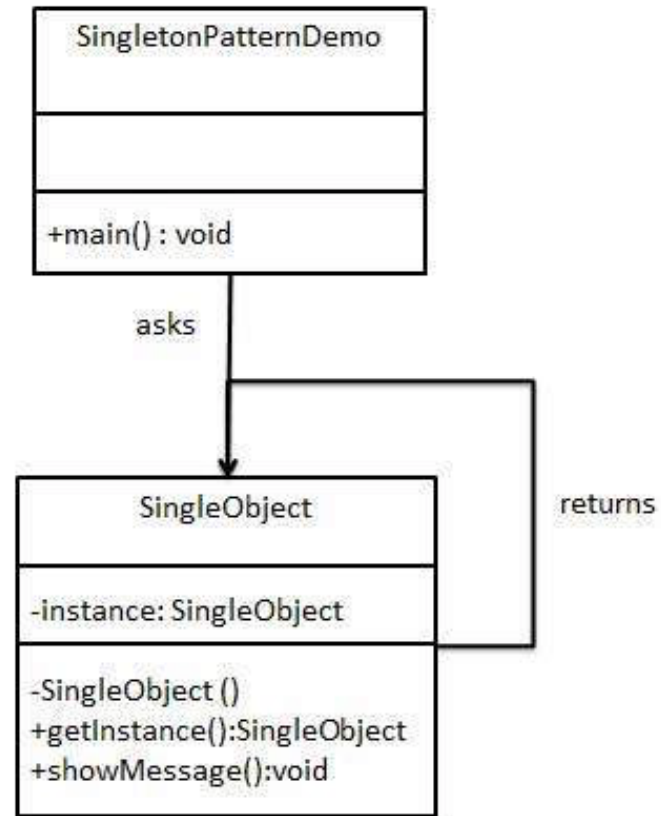
Lab 12 Singleton Design Pattern

May Mahmoud

Singleton

- Ensure a class has only one instance, and provide a global point of access to it

- Creational pattern

Implementation

Implementation

- Step 1
  - Create a Singleton Class

```java
public class SingleObject {

    //create an object of SingleObject
    private static SingleObject instance = new SingleObject();

    //make the constructor private so that this class cannot be
    //instantiated
    private SingleObject(){}

    //Get the only object available
    public static SingleObject getInstance(){
        return instance;
    }

    public void showMessage(){
        System.out.println("Hello World!");
    }
}
```

Implementation

- Step 2
  - Get the only object from the singleton class

```java
public class SingletonPatternDemo {
    public static void main(String[] args) {

        //illegal construct
        //Compile Time Error: The constructor SingleObject() is not visible
        //SingleObject object = new SingleObject();

        //Get the only object available
        SingleObject object = SingleObject.getInstance();

        //show the message
        object.showMessage();
    }
}
```