Name:- Kshitij Sonawane
Roll no:- 42
Div:- TE-15

Practical No: -11

Aim: - Singleton pattern.

Lab Outcome: - CSL501.3 Use computer-aided software engineering (CASE) tools.

Theory: -

Design Pattern:

- Design patterns represent the best practices used by experienced object-oriented software developers.
- Design patterns are solutions to common issues that software developers face when developing software.
- Several software engineers used trial and error to arrive at these solutions over a long period of time.

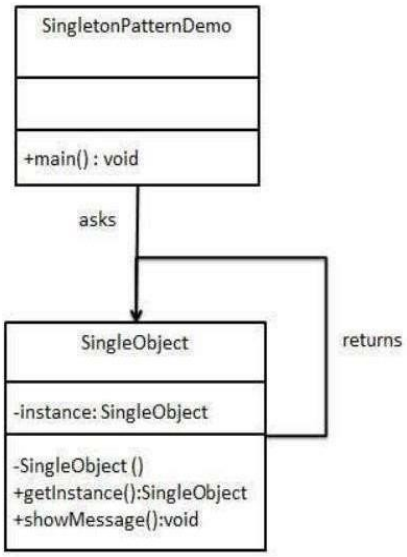
Singleton Pattern:

- The singleton pattern is a design pattern that restricts the instantiation of a class to one object.
- Singleton pattern is one of the simplest design patterns.
- This type of design pattern comes under creational pattern as this pattern provides one of the best ways to create an object.
- This pattern involves a single class which is responsible to create an object while making sure that only single object gets created.

Name:- Kshitij Sonawane
Roll no:- 42
Div:- TE-15

- This class provides a way to access its only object which can be accessed directly without need to instantiate the object of the class.

Diagram of Singleton Pattern:

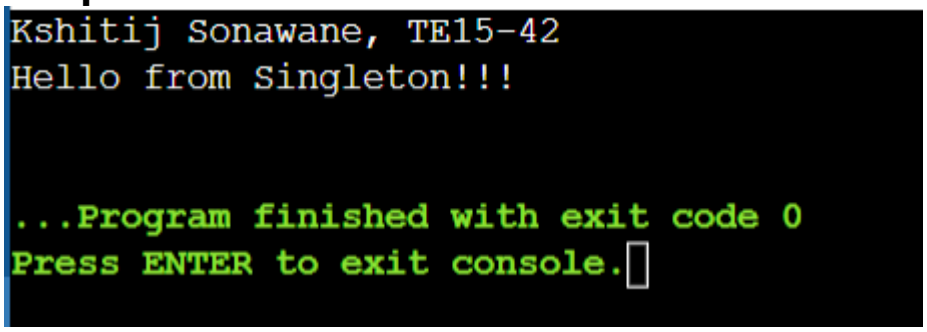


Code of Singleton Pattern:

```
public class Singleton {
private static Singleton single = new
Singleton();
private Singleton(){}
public static Singleton getInstance(){ return single; }
public void message(){
System.out.println("Hello from Singleton!!!");
}
public static void main(String[] args) {
System.out.println("Kshitij Sonawane, TE15-42");
Singleton singleObj = Singleton.getInstance();
singleObj.message();
}
```

Name:- Kshitij Sonawane
Roll no:- 42
Div:- TE-15

Output: -



Conclusion: - Here, we successfully implemented the Singleton pattern using java.