Constructor for Singleton Pattern

Write a Java program to create a class called Singleton that ensures only one instance of the class can be created. Implement a private constructor and a public static method to get the single instance of the class. Print a message indicating the creation of the instance.

Sample Solution:

Java Code:

Singleton.java

Copy

```java
// Define the Singleton class
public class Singleton {
    // Private static variable to hold the single instance
    private static Singleton singleInstance = null;

    // Private constructor to prevent instantiation
    private Singleton() {
        // Print a message indicating the creation of the instance
        System.out.println("Singleton instance created.");
    }

    // Public static method to get the single instance of the class
    public static Singleton getInstance() {
        // If the single instance is null, create a new instance
        if (singleInstance == null) {
            singleInstance = new Singleton();
        }
        // Return the single instance
        return singleInstance;
    }

    // Main method to test the Singleton class
    public static void main(String[] args) {
        // Get the single instance of Singleton
        Singleton instance1 = Singleton.getInstance();
        // Try to get another instance of Singleton
        Singleton instance2 = Singleton.getInstance();

        // Check if both instances are the same
        if (instance1 == instance2) {
            System.out.println("Both instances are the same.");
        } else {
            System.out.println("Instances are different.");
        }
    }
}
```

Output:

```
Single
Both i
```

Explanat