**WEEK-1**

**Design Patterns and Principles**

**Exercise 1: Implementing the Singleton Pattern**

**Code:**

**Java Project (Package)** :SingletonPatternExample

**Logger.java:**

public class Logger {

private static Logger singleobj;

private Logger() {

System.out.println("Singleton Logger class instance created");

}

public static Logger getinstance() {

if (singleobj == null) {

singleobj = new Logger();

}

return singleobj;

}

public void displayMessage() {

System.out.println("ID: 6421064");

}

}

**Test.java:**

public class Test {

public static void main(String[] args) {

Logger firstobj = Logger.getinstance();

Logger secondobj = Logger.getinstance();

firstobj.displayMessage();

secondobj.displayMessage();

if (firstobj == secondobj) {

System.out.println("follows Singleton");

} else {

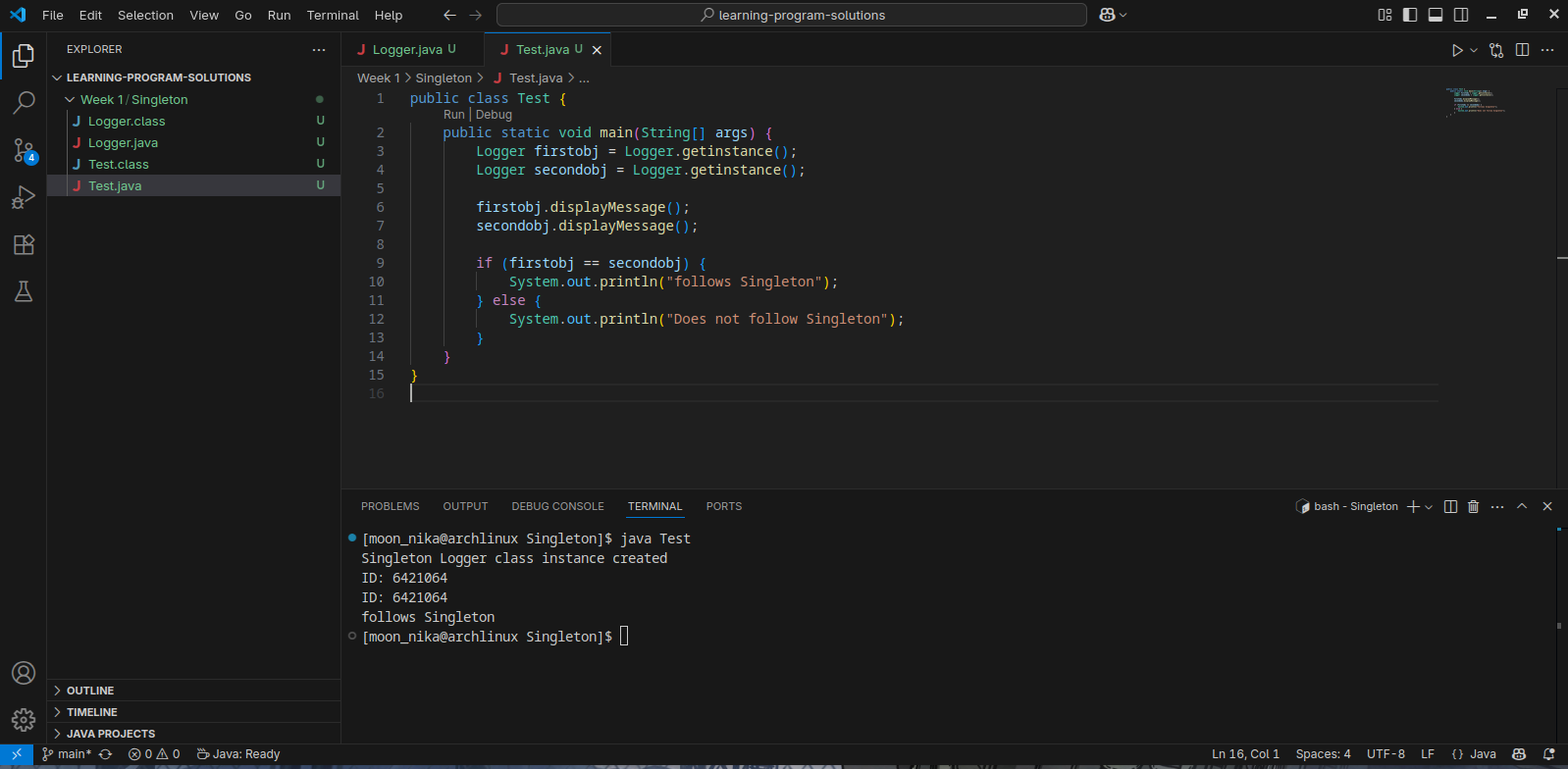
System.out.println("Does not follow Singleton");

}

}

}

**Output 1 Which Follows Singleton:-**



**Output 2 Which Not Follows Singleton:-**

