# Practical 4

Name: Arya Vishal
Roll no: 22BCE501
Subject: Software quality testing and Assurance

AIM: To study and perform sample tests using JUnit Testing tool for writing and executing automated unit tests:

Original code : Student Registration

Code:

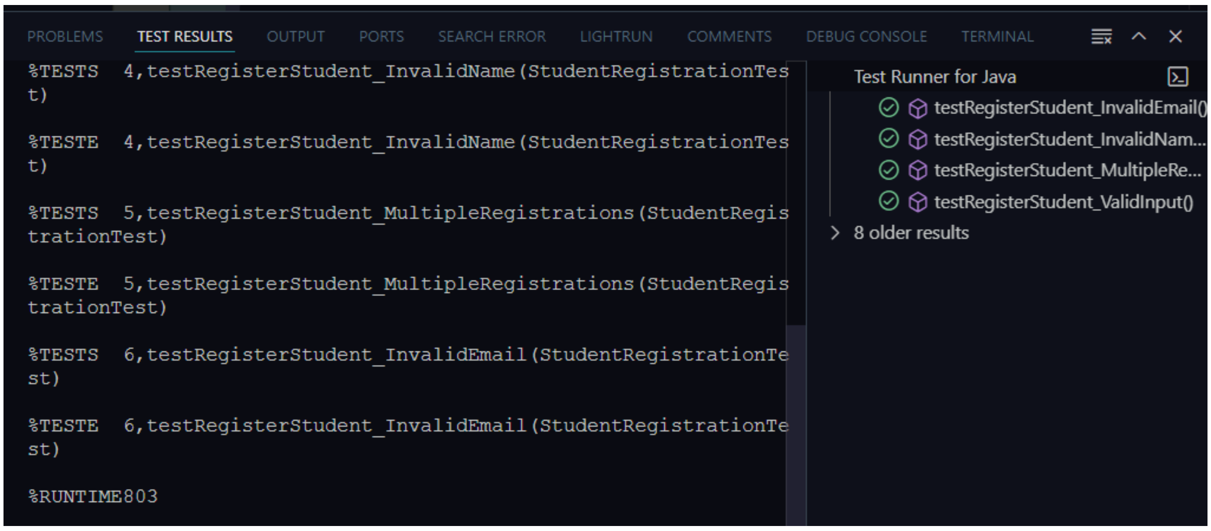
```
// Student.java
public class Student {
    private String name;
    private String email;

    public Student(String name, String email) {
        this.name = name;
        this.email = email;
    }

    public String getName() {
        return name;
    }

    public String getEmail() {
        return email;
    }
}
```

```
// StudentRegistration.java
import java.util.ArrayList;
import java.util.List;
```

```java
public class StudentRegistration {
    private List<Student> students = new ArrayList<>();

    public boolean registerStudent(String name, String
email) {
        if (name == null || email == null || email.isEmpty()
|| name.isEmpty()) {
            return false;
        }
        Student student = new Student(name, email);
        students.add(student);
        return true;
    }

    public List<Student> getRegisteredStudents() {
        return students;
    }
}
```

```java
// StudentRegistrationTest.java
import static org.junit.jupiter.api.Assertions.*;

import java.util.List;

import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Test;

public class StudentRegistrationTest {
    private StudentRegistration registration;

    @BeforeEach
    public void setUp() {
        registration = new StudentRegistration();
    }
    @Test
    public void testRegisterStudent_ValidInput() {
        boolean result = registration.registerStudent("John
Doe", "john.doe@example.com");
        assertTrue(result);

        List<Student> registeredStudents =
registration.getRegisteredStudents();
        assertEquals(1, registeredStudents.size());
        assertEquals("John Doe",
registeredStudents.get(0).getName());
        assertEquals("john.doe@example.com",
registeredStudents.get(0).getEmail());
    }

    @Test
    public void testRegisterStudent_InvalidName() {
        boolean result = registration.registerStudent("",
"john.doe@example.com");
        assertFalse(result);

        result = registration.registerStudent(null,
"john.doe@example.com");
        assertFalse(result);
    }

    @Test
    public void testRegisterStudent_InvalidEmail() {
        boolean result = registration.registerStudent("John
Doe", "");
        assertFalse(result);

        result = registration.registerStudent("John Doe",
null);
        assertFalse(result);
    }
    @Test
    public void testRegisterStudent_MultipleRegistrations()
{
        registration.registerStudent("Jane Doe",
"jane.doe@example.com");
        registration.registerStudent("John Smith",
"john.smith@example.com");

        List<Student> registeredStudents =
registration.getRegisteredStudents();
        assertEquals(2, registeredStudents.size());
    }
}
```

Output:



```
%TESTS  4,testRegisterStudent_InvalidName(StudentRegistrationTes
t)

%TESTE  4,testRegisterStudent_InvalidName(StudentRegistrationTes
t)

%TESTS  5,testRegisterStudent_MultipleRegistrations(StudentRegis
trationTest)

%TESTE  5,testRegisterStudent_MultipleRegistrations(StudentRegis
trationTest)

%TESTS  6,testRegisterStudent_InvalidEmail(StudentRegistrationTe
st)

%TESTE  6,testRegisterStudent_InvalidEmail(StudentRegistrationTe
st)

%RUNTIME803
```

Test Runner for Java
- ✓ testRegisterStudent_InvalidEmail()
- ✓ testRegisterStudent_InvalidNam...
- ✓ testRegisterStudent_MultipleRe...
- ✓ testRegisterStudent_ValidInput()

> 8 older results