



MUHAMMAD ALI KHAN

REG# 16550

SEMESTER : 6TH

PAPER : Software **Verification and validation.**

QUESTION # 1:

TOPIC :LOAD RUNNER

ANSWER:

1.PROS/CONS:

=> The ease with which the Test scripts can be recorded. Tests run and results analyzed when simultaneous users are accessing the application.

=> Easy integration with the other HP suite tools.

=> Ease of creating Virtual Users to run the test cases and even no limitation on the number of VUsers.

⇒ Extensive vendor documentation and help system included.

⇒ Load Runner can work with most Enterprise Resource Planning Systems.

* The tool cost is a bit more expensive when compared to other tools and it is a bit difficult for the first-time installation.

* Though good support is available from the HP team, it's expensive if 24x7 support is required.

* Too much RAM resource is required for the tool to work efficiently to avoid crashing of application.

* It takes a lot of time in scripting and it is an even more complex job when the application build is changing frequently.

2. FUNCTIONALITY:

⇒ Load runner is used to test applications, measuring system behavior and performance under load.

⇒ Load Runner can simulate thousands of users concurrently using application software, recording and later analyzing the performance of key components of the application.

⇒ Load Runner simulates user activity by generating messages between application components or by simulating interactions with the user interface such as key presses or mouse movements.

⇒ The messages and interactions to be generated are stored in scripts.

⇒ Load Runner can generate the scripts by recording them, such as logging HTTP requests between a client web browser and an application's web server.

3. SUPPORTING LANGUAGES:

Following languages are supported by Load Runner:

1. C Language
2. Visual Basic Scripting
3. Visual Basic For Applications
4. Java Scripting

4. SUPPORTING TESTS:

Following are the tests Load Runner can support are:

- NET Record / Display
- Database
- DCOM
- Network
- Oracle E-Business
- SAP
- SOA
- Web and Multimedia
- Wireless
- GUI
- Java Record and replay
- Remote desktop
- Web 2.0

5. A SHORT (FAULTY) CODE, TEST USING THIS TOOL, AND SHOW THE BUGS IN THE CODE:

TEST : 1:

```
package testRunnerDemo;

import org.junit.Test;

public class Test1 {

    @Test
    public void test() {
        System.out.println("Test 1 is going executed");
    }

}

|

}
```

TEST : 2 :

```
package testRunnerDemo;
```

```
import org.junit.Test;

public class Test2 {

    @Test
    public void test() {
        System.out.println("Test 2 is going executed");
    }

}

|

}
```

TEST RUNNER :

```
package testRunnerDemo ;

import org.junit.runner.JUnitCore;

public class TestRunner {

    public static void main(String[] args) {

        Result result = .JUnitCore.runClasses(Test1.class,Test2.class);

        for (Failure failure : result.getFailures()) {

            System.out.println(failure.toString() );

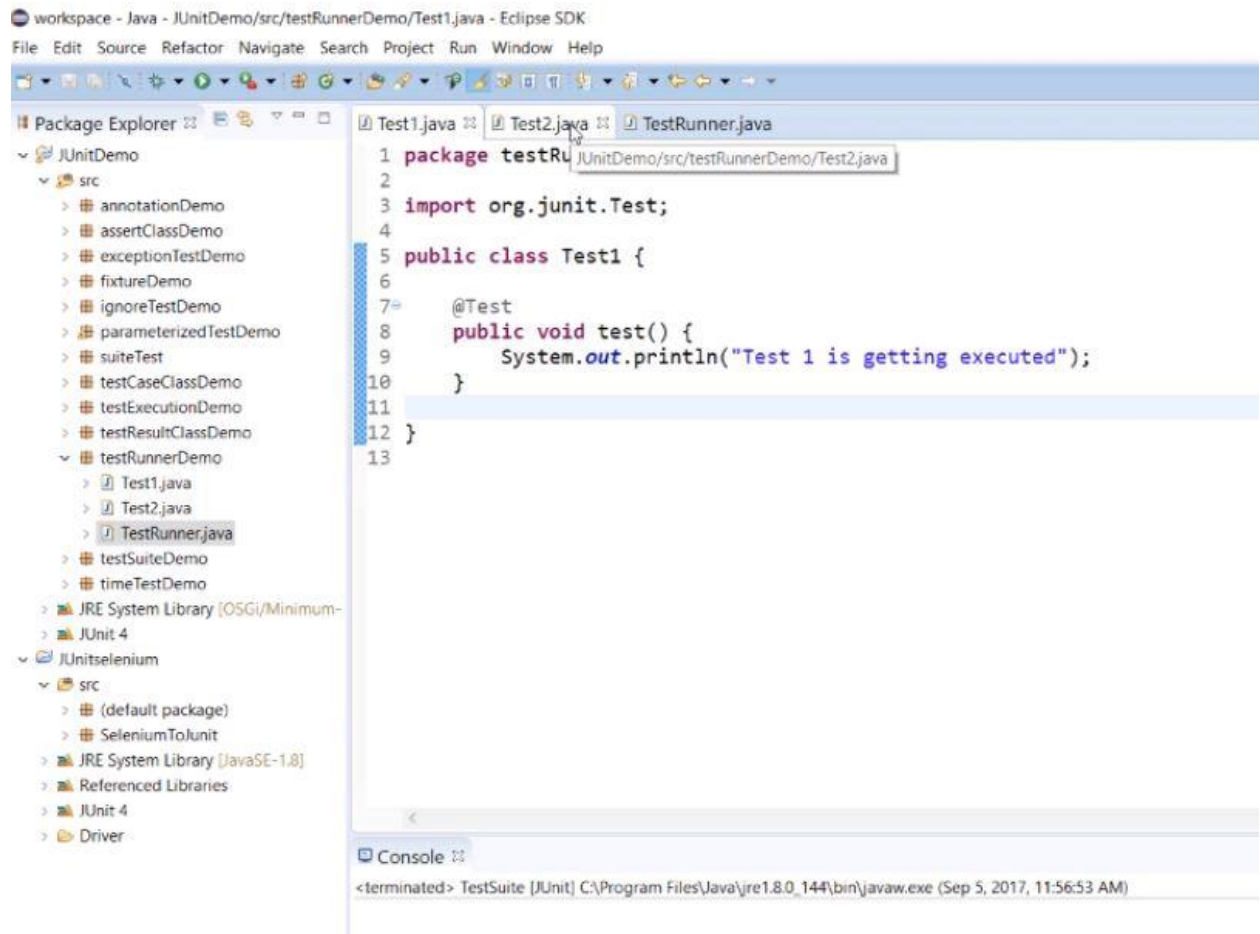
        }

        System.out.println(result.wasSuccessful());

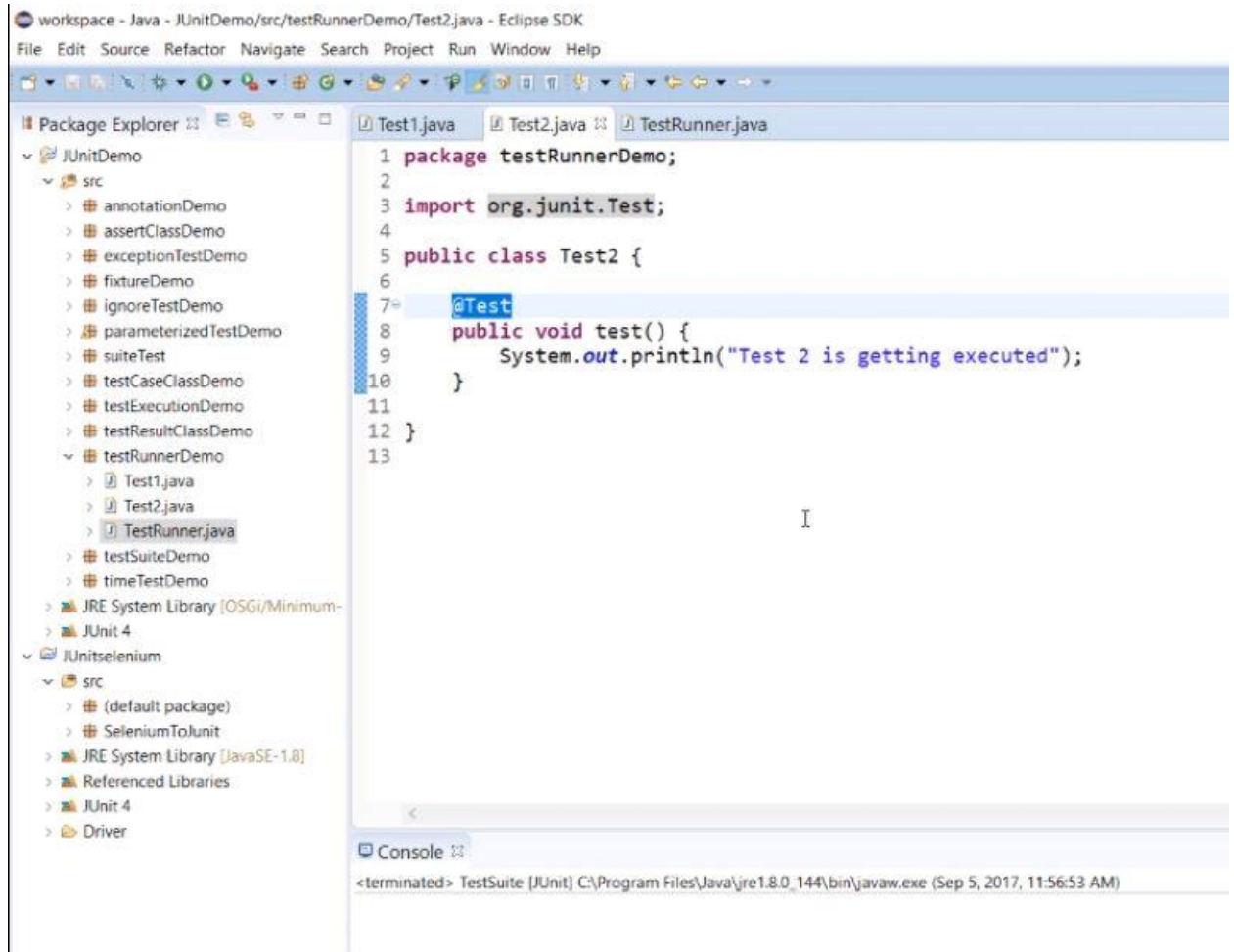
    }

}
```

TEST 1:



TEST 2:



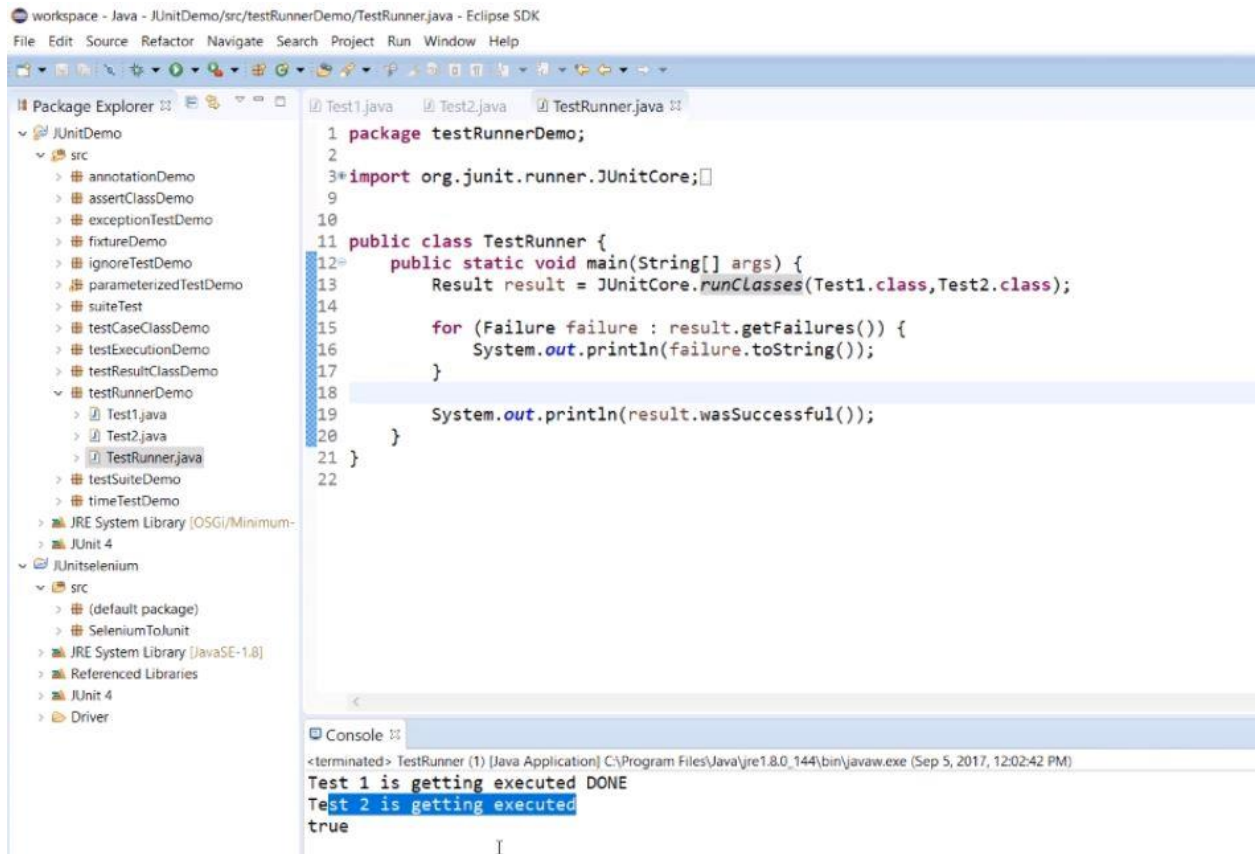
The screenshot shows the Eclipse IDE interface. The Package Explorer on the left displays the project structure, including the testRunnerDemo package and its sub-packages. The main editor window shows the source code for Test2.java, which is a Java class with a single test method. The code is as follows:

```
1 package testRunnerDemo;
2
3 import org.junit.Test;
4
5 public class Test2 {
6
7     @Test
8     public void test() {
9         System.out.println("Test 2 is getting executed");
10    }
11
12 }
13
```

The console at the bottom shows the output of the test suite, indicating that the test was terminated successfully.

```
<terminated> TestSuite [JUnit] C:\Program Files\Java\jre1.8.0_144\bin\javaw.exe (Sep 5, 2017, 11:56:53 AM)
```

TEST 3: “Runner test”



The screenshot shows the Eclipse IDE interface. The Package Explorer on the left displays a project structure with a 'testRunnerDemo' package containing 'Test1.java', 'Test2.java', and 'TestRunner.java'. The main editor window shows the code for 'TestRunner.java':

```
1 package testRunnerDemo;
2
3 import org.junit.runner.JUnit4;
9
10
11 public class TestRunner {
12     public static void main(String[] args) {
13         Result result = JUnit4.runClasses(Test1.class, Test2.class);
14
15         for (Failure failure : result.getFailures()) {
16             System.out.println(failure.toString());
17         }
18
19         System.out.println(result.wasSuccessful());
20     }
21 }
22
```

The Console window at the bottom shows the execution output:

```
<terminated> TestRunner (1) [Java Application] C:\Program Files\Java\jre1.8.0_144\bin\javaw.exe (Sep 5, 2017, 12:02:42 PM)
Test 1 is getting executed DONE
Test 2 is getting executed
true
```

We wrote this runner code to check whether the code is successfully executed or not.

It returned true.

