

# JUnit - Basic Usage

Let us now have a basic example to demonstrate the step-by-step process of using JUnit.

## Create a Class

Create a java class to be tested, say, MessageUtil.java in **C:\>JUNIT\_WORKSPACE**

```
/*
 * This class prints the given message on console.
 */

public class MessageUtil {

    private String message;

    //Constructor
    //@param message to be printed

    public MessageUtil(String message){
        this.message = message;
    }

    // prints the message
    public String printMessage(){
        System.out.println(message);
        return message;
    }
}
```

## Create Test Case Class

Create a java test class, say, TestJUnit.java.

Add a test method testPrintMessage() to your test class.

Add an Annotation @Test to method testPrintMessage().

Implement the test condition and check the condition using assertEquals API of JUnit.

Create a java class file name TestJUnit.java in **C:\>JUNIT\_WORKSPACE**.

```
import org.junit.Test;
import static org.junit.Assert.assertEquals;

public class TestJUnit {

    String message = "Hello World";
    MessageUtil messageUtil = new MessageUtil(message);

    @Test
    public void testPrintMessage() {
        assertEquals(message,messageUtil.printMessage());
    }
}
```

Explore our **latest online courses** and learn new skills at your own pace. Enroll and become a certified expert to boost your career.

## Create Test Runner Class

Create a TestRunner java class.

Use runClasses method of JUnitCore class of JUnit to run the test case of the above created test class.

Get the result of test cases run in Result Object.

Get failure(s) using the getFailures() method of Result object.

Get Success result using the wasSuccessful() method of Result object.

Create a java class file named TestRunner.java in **C:\>JUNIT\_WORKSPACE** to execute test case(s).

```
import org.junit.runner.JUnitCore;
import org.junit.runner.Result;
import org.junit.runner.notification.Failure;

public class TestRunner {
    public static void main(String[] args) {
        Result result = JUnitCore.runClasses(TestJUnit.class);

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

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

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

Compile the MessageUtil, Test case and Test Runner classes using javac.

```
C:\JUNIT_WORKSPACE>javac MessageUtil.java TestJunit.java TestRunner.java
```

Now run the Test Runner, which will run the test case defined in the provided Test Case class.

```
C:\JUNIT_WORKSPACE>java TestRunner
```

Verify the output.

```
Hello World
true
```

Now update TestJunit in **C:\>JUNIT\_WORKSPACE** so that the test fails. Change the message string.

```
import org.junit.Test;
import static org.junit.Assert.assertEquals;

public class TestJunit {

    String message = "Hello World";
    MessageUtil messageUtil = new MessageUtil(message);

    @Test
    public void testPrintMessage() {
        message = "New Word";
        assertEquals(message,messageUtil.printMessage());
    }
}
```

Let's keep the rest of the classes as is, and try to run the same Test Runner.

```
import org.junit.runner.JUnitCore;
import org.junit.runner.Result;
import org.junit.runner.notification.Failure;

public class TestRunner {

    public static void main(String[] args) {
        Result result = JUnitCore.runClasses(TestJunit.class);

        for (Failure failure : result.getFailures()) {
            System.out.println(failure.toString());
        }

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

Now run the Test Runner, which will run the test case defined in the provided Test Case class.

```
C:\JUNIT_WORKSPACE>java TestRunner
```

Verify the output.

```
Hello World
testPrintMessage(TestJunit): expected:<[New Wor]d> but was:<[Hello Worl]d>
false
```