Here we will see one complete example of JUnit testing using POJO class, Business logic class, and a test class, which will be run by the test runner.

Create **EmployeeDetails.java** in C:\>JUNIT_WORKSPACE, which is a POJO class.

```
public class EmployeeDetails {

   private String name;
   private double monthlySalary;
   private int age;

   /**
   * @return the name
   */

   public String getName() {
      return name;
   }

   /**
   * @param name the name to set
   */

   public void setName(String name) {
      this.name = name;
   }

   /**
   * @return the monthlySalary
   */
```

```
   public double getMonthlySalary() {
      return monthlySalary;
   }

   /**
   * @param monthlySalary the monthlySalary to set
   */

   public void setMonthlySalary(double monthlySalary) {
      this.monthlySalary = monthlySalary;
   }

   /**
   * @return the age
   */
   public int getAge() {
      return age;
   }

   /**
   * @param age the age to set
   */
   public void setAge(int age) {
      this.age = age;
   }
}
```

EmployeeDetails class is used to –

- get/set the value of employee's name.
- get/set the value of employee's monthly salary.

- get/set the value of employee's age.

Create a file called **EmpBusinessLogic.java** in C:\>JUNIT_WORKSPACE, which contains the business logic.

```
public class EmpBusinessLogic {

   // Calculate the yearly salary of employee
   public double calculateYearlySalary(EmployeeDetails employeeDetails) {
      double yearlySalary = 0;
      yearlySalary = employeeDetails.getMonthlySalary() * 12;
      return yearlySalary;
   }

   // Calculate the appraisal amount of employee
   public double calculateAppraisal(EmployeeDetails employeeDetails) {
      double appraisal = 0;

      if(employeeDetails.getMonthlySalary() < 10000){
         appraisal = 500;
      }else{
         appraisal = 1000;
      }

      return appraisal;
   }
}
```

EmpBusinessLogic class is used for calculating −

- the yearly salary of an employee.
- the appraisal amount of an employee.

Create a file called **TestEmployeeDetails.java** in C:\>JUNIT_WORKSPACE, which contains the test cases to be tested.

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

public class TestEmployeeDetails {
   EmpBusinessLogic empBusinessLogic = new EmpBusinessLogic();
   EmployeeDetails employee = new EmployeeDetails();

   //test to check appraisal
   @Test
   public void testCalculateAppriasal() {
      employee.setName("Rajeev");
      employee.setAge(25);
      employee.setMonthlySalary(8000);

      double appraisal = empBusinessLogic.calculateAppraisal(employee);
      assertEquals(500, appraisal, 0.0);
   }

   // test to check yearly salary
   @Test
   public void testCalculateYearlySalary() {
      employee.setName("Rajeev");
      employee.setAge(25);
      employee.setMonthlySalary(8000);

      double salary = empBusinessLogic.calculateYearlySalary(employee);
      assertEquals(96000, salary, 0.0);
   }
}
```

TestEmployeeDetails class is used for testing the methods of **EmpBusinessLogic** class. It

- tests the yearly salary of the employee.
- tests the appraisal amount of the employee.

Next, create a java class filed 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(TestEmployeeDetails.class);


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

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

      }


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

   }

}
```

Compile the test case and Test Runner classes using javac.

```
C:\JUNIT_WORKSPACE>javac EmployeeDetails.java
EmpBusinessLogic.java TestEmployeeDetails.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.

```
true
```