

JUnit Assessment No: 01

Name: Gaurav Nandgopal Deshmukh

USN: 72232433C

Problem Statement:

1. Create Testcases/Method in EmployeeServiceTest in the Project Employee Management
 - Use assertions to check if expected and actual output are same or not
 - Provide custom names to all testcases and test class and also check the usage of @Disabled annotation

1.Employee.java

Employee

int id

String name

double monthlySalary

2. EmployeeService.java

EmployeeService

double calculateHike(Employee employee)

double calculateYearlySalary(Employee employee)

You can look into the sample EmployeeService.java implementation as below:

```
public class EmployeeService {
```

```
    public double calculateYearlySalary(Employee employee) {1
```

```
        double yearlySalary = 0;2
```

```

yearlySalary = employee.getMonthlySalary() * 12;3
return yearlySalary;4
}5
public double calculateHike(Employee employee) {6
double hike = 0;7
if (employee.getMonthlySalary() < 10000) {8
hike = 2000;9
} else {10
hike = 1000;11
}12
return 13hike;
}
}

```

2. Utilize Test Fixtures in EmployeeServiceTest class of Employee Management project for creating and destroying object of EmployeeService class.

Also log appropriate statements in Test Fixtures to verify their life cycle.

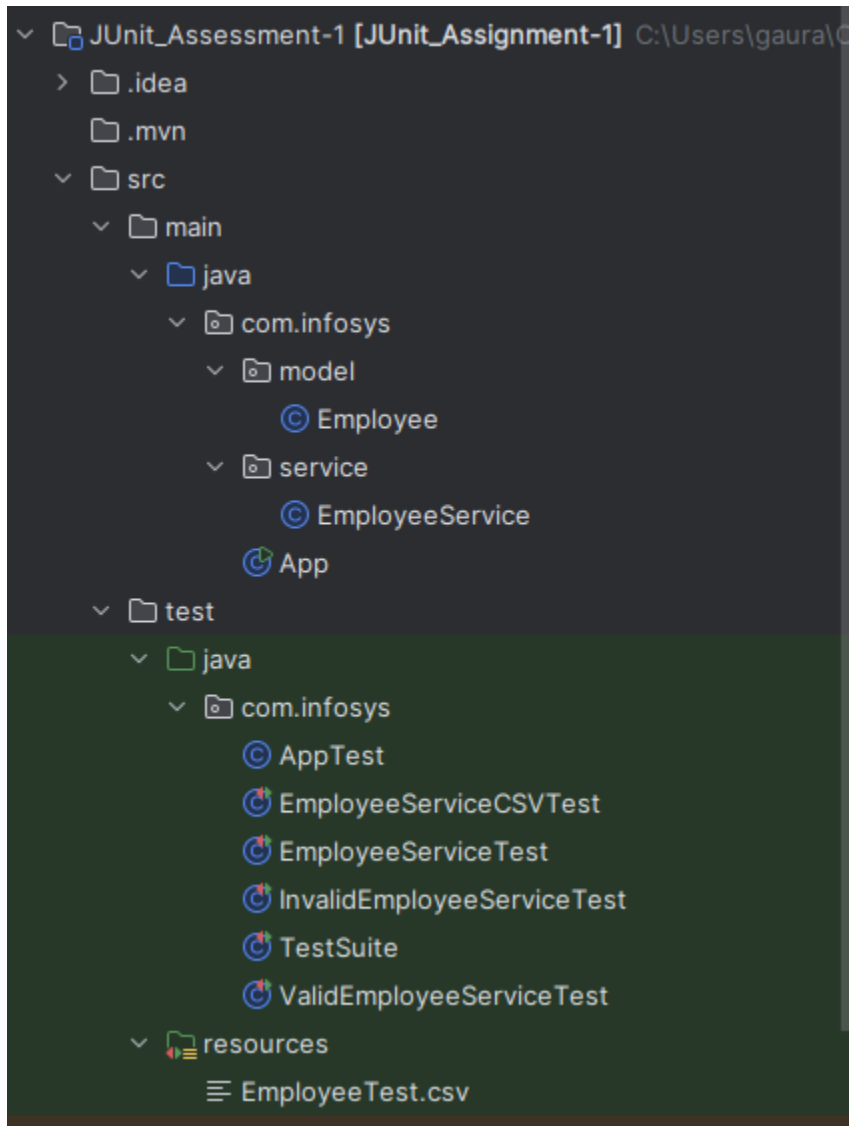
3) Reuse the testcases of Employee Management project and divide valid and invalid testcases for the calculateYearlySalary() and calculateHike() functionalities in separate test classes.

Create a test suite and run valid/invalid test cases based on the name patterns.

4. modify test class EmployeeServiceTest in Employee Management project.

Create a testcase for the calculateYearlySalary() functionality and read multiple test inputs from a CSV file.

Solution:



Pom.xml:

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd"
  <modelVersion>4.0.0</modelVersion>

  <groupId>com.infosys</groupId>
  <artifactId>JUnit_Assignment-1</artifactId>
  <version>1.0-SNAPSHOT</version>
  <packaging>jar</packaging>

  <name>JUnit_Assignment-1</name>
  <url>http://maven.apache.org</url>

  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
  </properties>

  <dependencies>
    <!-- https://mvnrepository.com/artifact/org.junit.jupiter/junit-jupiter-api -->
    <dependency>
      <groupId>org.junit.jupiter</groupId>
      <artifactId>junit-jupiter-api</artifactId>
      <version>5.11.4</version>
      <scope>test</scope>
    </dependency>
    <!-- https://mvnrepository.com/artifact/org.junit.jupiter/junit-jupiter-engine -->
    <dependency>
      <groupId>org.junit.jupiter</groupId>
      <artifactId>junit-jupiter-engine</artifactId>
```

60:11 CRLF UTF-8 2 spaces*

```
<!-- https://mvnrepository.com/artifact/org.junit.jupiter/junit-jupiter-api -->
<dependency>
  <groupId>org.junit.jupiter</groupId>
  <artifactId>junit-jupiter-api</artifactId>
  <version>5.11.4</version>
  <scope>test</scope>
</dependency>
<!-- https://mvnrepository.com/artifact/org.junit.jupiter/junit-jupiter-engine -->
<dependency>
  <groupId>org.junit.jupiter</groupId>
  <artifactId>junit-jupiter-engine</artifactId>
  <version>5.11.4</version>
  <scope>test</scope>
</dependency>

<!-- https://mvnrepository.com/artifact/org.junit.platform/junit-platform-suite -->
<dependency>
  <groupId>org.junit.platform</groupId>
  <artifactId>junit-platform-suite</artifactId>
  <version>1.9.1</version>
  <scope>test</scope>
</dependency>

<!-- https://mvnrepository.com/artifact/org.junit.jupiter/junit-jupiter-params -->
<dependency>
```

```

<!-- https://mvnrepository.com/artifact/org.junit.jupiter/junit-jupiter-params -->
<dependency>
  <groupId>org.junit.jupiter</groupId>
  <artifactId>junit-jupiter-params</artifactId>
  <version>5.11.4</version>
  <scope>test</scope>
</dependency>

</dependencies>

<build>
  <plugins>
    <plugin>
      <!-- https://mvnrepository.com/artifact/org.apache.maven.plugins/maven-surefire-plugin -->
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-surefire-plugin</artifactId>
      <version>3.5.2</version>
    </plugin>
  </plugins>
</build>

</project>

```

Employee.java:

```
package com.infosys.model;

public class Employee { 21 usages  gauravdeshmukh01
    private int id; 3 usages
    private String name; 3 usages
    private double monthlySalary; 3 usages

    public Employee() {} no usages  gauravdeshmukh01

    public Employee(int id, String name, double monthlySalary) { 7 usages  gauravdeshmukh01
        this.id = id;
        this.name = name;
        this.monthlySalary = monthlySalary;
    }

    public int getId() { return id; } no usages  gauravdeshmukh01
    public String getName() { return name; } no usages  gauravdeshmukh01
    public double getMonthlySalary() { return monthlySalary; } 2 usages  gauravdeshmukh01

    public void setId(int id) { this.id = id; } no usages  gauravdeshmukh01
    public void setName(String name) { this.name = name; } no usages  gauravdeshmukh01
    public void setMonthlySalary(double monthlySalary) { this.monthlySalary = monthlySalary; }
}
```

EmployeeService.java:

```

package com.infosys.service;

import com.infosys.model.Employee;

public class EmployeeService { 12 usages  gauravdeshmukh01

    public double calculateYearlySalary(Employee employee) { 4 usages  gauravdeshmukh01
        return employee.getMonthlySalary() * 12;
    }

    public double calculateHike(Employee employee) { 3 usages  gauravdeshmukh01
        if (employee.getMonthlySalary() < 10000) {
            return 2000;
        } else {
            return 1000;
        }
    }
}

```

EmployeeServiceCSVTest:

```

package com.infosys;

import com.infosys.model.Employee;
import com.infosys.service.EmployeeService;
import org.junit.jupiter.params.ParameterizedTest;
import org.junit.jupiter.params.provider.CsvFileSource;

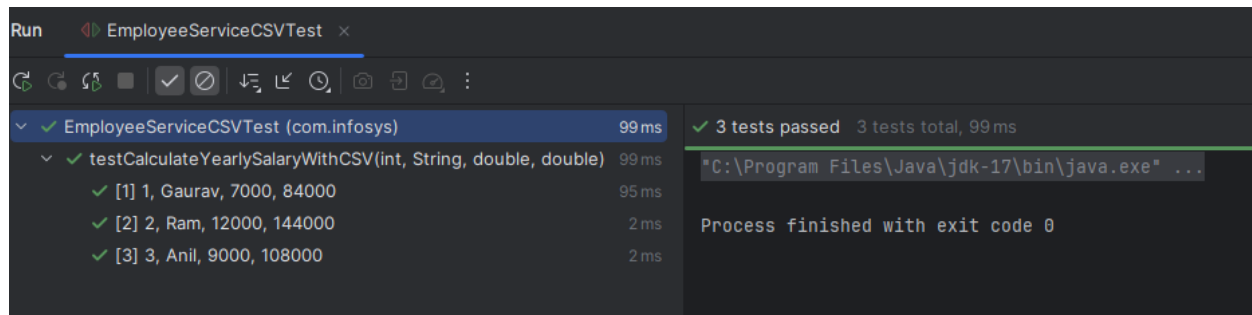
import static org.junit.jupiter.api.Assertions.assertEquals;

public class EmployeeServiceCSVTest {  gauravdeshmukh01

    private EmployeeService employeeService = new EmployeeService(); 1 usage

    @ParameterizedTest  gauravdeshmukh01
    @CsvFileSource(files = "src/test/resources/EmployeeTest.csv", numLinesToSkip = 1)
    void testCalculateYearlySalaryWithCSV(int id, String name, double monthlySalary, double expectedYearlySalary) {
        Employee employee = new Employee(id, name, monthlySalary);
        double actual = employeeService.calculateYearlySalary(employee);
        assertEquals(expectedYearlySalary, actual);
    }
}

```

EmployeeServiceTest:

```
package com.infosys;

import com.infosys.model.Employee;
import com.infosys.service.EmployeeService;
import org.junit.jupiter.api.*;

import static org.junit.jupiter.api.Assertions.*;

@TestMethodOrder(MethodOrderer.OrderAnnotation.class)
@DisplayName("Employee Service Test Suite")
public class EmployeeServiceTest {

    private EmployeeService employeeService;

    @BeforeAll
    static void beforeAll() { System.out.println("Before All - Runs once before all tests"); }

    @BeforeEach
    void setUp() {
        employeeService = new EmployeeService();
        System.out.println("Before Each - Creating EmployeeService instance");
    }

    @Test
    @Order(1)
    @DisplayName("Test Yearly Salary Calculation")
    void testCalculateYearlySalary() {
```

```

void testCalculateYearlySalary() {
    assertEquals(expected, actual, message: "Yearly Salary should be monthlySalary * 12");
}

@Test  👤 gauravdeshmukh01
@Order(2)
@DisplayName("Test Hike Calculation for Low Salary")
void testCalculateHike_LowSalary() {
    Employee employee = new Employee(id: 2, name: "Gaurav", monthlySalary: 7000);
    double expected = 2000;
    double actual = employeeService.calculateHike(employee);
    assertEquals(expected, actual, message: "Hike should be 2000 for salary below 10000");
}

@Test  👤 gauravdeshmukh01
@Order(3)
@Disabled("Disabled for demonstration")
@DisplayName("Test Disabled Example")
void testDisabledExample() { fail("This test should be disabled"); }

@AfterEach  👤 gauravdeshmukh01
void tearDown() { System.out.println("After Each - Destroying EmployeeService instance"); }

@AfterAll  👤 gauravdeshmukh01
static void afterAll() { System.out.println("After All - Runs once after all tests"); }
}

```

Run EmployeeServiceTest

Test Suite	Duration	Status
Employee Service Test Suite (com.infosys)	31 ms	2 tests passed, 1 ignored 3 tests total, 31 ms
Test Yearly Salary Calculation	30 ms	Passed
Test Hike Calculation for Low Salary	1 ms	Passed
Test Disabled Example		Ignored

```

"C:\Program Files\Java\jdk-17\bin\java.exe" ...
Before All - Runs once before all tests
Before Each - Creating EmployeeService instance
After Each - Destroying EmployeeService instance
Before Each - Creating EmployeeService instance
After Each - Destroying EmployeeService instance

Disabled for demonstration
After All - Runs once after all tests

Process finished with exit code 0

```

InvalidEmployeeServiceTest:

```
import com.infosys.service.EmployeeService;
import org.junit.jupiter.api.*;

import static org.junit.jupiter.api.Assertions.*;

@DisplayName("Employee Service Invalid Output Tests")
@Tag("invalid")
public class InvalidEmployeeServiceTest {

    private EmployeeService employeeService = new EmployeeService();

    @Test
    @DisplayName("Verify Yearly Salary is Not an Incorrect Value (e.g., 100000)")
    @Tag("salary")
    void invalidYearlySalary() {
        Employee e = new Employee(id: 1, name: "Aman", monthlySalary: 5000);
        assertEquals(unexpected: 100000, employeeService.calculateYearlySalary(e),
            message: "Yearly Salary should not be 100000 for monthly 5000");
    }

    @Test
    @DisplayName("Verify Hike is Not an Incorrect Value (e.g., 2000 for high salary)")
    @Tag("hike")
    void invalidHike() {
        Employee e = new Employee(id: 2, name: "Dipak", monthlySalary: 12000);
        assertEquals(unexpected: 2000, employeeService.calculateHike(e),
            message: "Hike should not be 2000 for monthly salary 12000 (should be 1000)");
    }
}
```

Run InvalidEmployeeServiceTest

✓ Employee Service Invalid Output Tests (com.infosys) 32 ms

- ✓ Verify Yearly Salary is Not an Incorrect Value (e.g., 100000) 31 ms
- ✓ Verify Hike is Not an Incorrect Value (e.g., 2000 for high salary) 1 ms

✓ 2 tests passed 2 tests total, 32 ms

"C:\Program Files\Java\jdk-17\bin\java.exe" ...

Process finished with exit code 0

TestSuite:

```

package com.infosys;

import org.junit.platform.suite.api.IncludeTags;
import org.junit.platform.suite.api.SelectClasses;
import org.junit.platform.suite.api.Suite;

@Suite
@SelectClasses({
    ValidEmployeeServiceTest.class,
    InvalidEmployeeServiceTest.class,
    EmployeeServiceCSVTest.class
})
@IncludeTags({"valid"})
public class TestSuite {
}

```

Run TestSuite x

✓ TestSuite (com.infosys) 49 ms

✓ Valid Tests 49 ms

✓ Valid Hike Calculation 48 ms

✓ Valid Yearly Salary Calculation 1 ms

✓ 2 tests passed 2 tests total, 49 ms

"C:\Program Files\Java\jdk-17\bin\java.exe" ...

Process finished with exit code 0

ValidEmployeeServiceTest:

```

import com.infosys.service.EmployeeService;
import org.junit.jupiter.api.*;

import static org.junit.jupiter.api.Assertions.*;

@DisplayName("Valid Tests")
@Tag("valid")
public class ValidEmployeeServiceTest {

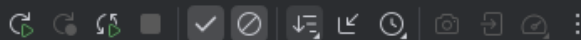
    private EmployeeService employeeService = new EmployeeService();

    @Test
    @DisplayName("Valid Yearly Salary Calculation")
    @Tag("salary")
    void validYearlySalary() {
        Employee e = new Employee(id: 1, name: "Ram", monthlySalary: 12000);
        assertEquals(expected: 144000, employeeService.calculateYearlySalary(e));
    }

    @Test
    @DisplayName("Valid Hike Calculation")
    @Tag("hike")
    void validHike() {
        Employee e = new Employee(id: 2, name: "Anil", monthlySalary: 9000);
        assertEquals(expected: 2000, employeeService.calculateHike(e));
    }
}

```

Run ValidEmployeeServiceTest



Test Name	Duration
Valid Tests (com.infosys)	34 ms
Valid Hike Calculation	33 ms
Valid Yearly Salary Calculation	1 ms

✓ 2 tests passed 2 tests total, 34 ms

"C:\Program Files\Java\jdk-17\bin\java.exe" ...

Process finished with exit code 0

EmployeeTest.csv:

```
id,name,monthlySalary,expectedYearly  
1,Gaurav,7000,84000  
2,Ram,12000,144000  
3,Anil,9000,108000
```

```
|
```