

### ASSIGNMENT-3

```
1.package com.test;

import java.util.Objects;

public class Employee {
    private int id;
    private String name;
    private double salary;

    public Employee() {
        super();
    }

    public Employee(int id, String name, double salary) {
        super();
        this.id = id;
        this.name = name;
        this.salary = salary;
    }

    public void raiseSalary(double amount) {
        this.salary += amount;
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getName() {
        return name;
    }

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

    public double getSalary() {
        return salary;
    }

    public void setSalary(double salary) {
        this.salary = salary;
    }

    @Override
    public int hashCode() {
        return Objects.hash(id, name, salary);
    }

    @Override
    public boolean equals(Object obj) {
        if (this == obj)
```

```

        return true;
    if (obj == null)
        return false;
    if (getClass() != obj.getClass())
        return false;
    Employee other = (Employee) obj;
    return id == other.id && Objects.equals(name, other.name)
        && Double.doubleToLongBits(salary) ==
Double.doubleToLongBits(other.salary);
    }

    public double calculateYearlyBonus() {
        return this.salary * 0.1; // 10% of the salary as bonus
    }
}

```

2.package com.test;

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

```
import org.junit.After;
import org.junit.Before;
import org.junit.Test;
```

```
public class EmployeeTest {

    @Before
    public void beforeEachTest() {
        System.out.println("Test Started>>>>>");
    }
    @After
    public void afterEachTest() {
        System.out.println("Test ended.....");
    }

    @Test
    public void testEmployeeAttributes() {
        Employee emp1 = new Employee(1, "John Doe", 50000.0);

        assertEquals(1, emp1.getId());
        assertEquals("John Doe", emp1.getName());
        assertEquals(50000.0, emp1.getSalary(), 0.0);
    }
    @Test
    public void testRaiseSalary() {
        Employee emp = new Employee(1, "Jane Smith", 60000.0);

        emp.raiseSalary(10000.0);
        assertEquals(70000.0, emp.getSalary(), 0.0);
    }
    @Test
    public void testEmployeeEquality() {
        Employee emp = new Employee(1, "Jane Smith", 60000.0);
        Employee emp1 = new Employee(1, "John Doe", 50000.0);
        assertEquals(false, emp.equals(emp1));
    }
}

```

```

2.
1.
module Junit_Assignmen {
}

2. package com.test;

import static org.junit.Assert.*;

import org.junit.After;
import org.junit.Before;
import org.junit.jupiter.params.ParameterizedTest;
import org.junit.jupiter.params.provider.ValueSource;

public class EmployeeParameterizedTest {

    @Before
    public void beforeEachTest() {
        System.out.println("Test Started>>>>>");
    }
    @After
    public void afterEachTest() {
        System.out.println("Test ended.....");
    }

    @ParameterizedTest
    @ValueSource(doubles = {50000.0, 60000.0, 75000.0})

    public void testCalculateYearlyBonus(double salary) {
        Employee employee = new Employee(1, "John Doe", salary);
        double expectedBonus = salary * 0.1; // 10% of the salary as
bonus
        assertEquals(expectedBonus,
employee.calculateYearlyBonus(),0.0);
    }

}

```