Problem 1. Create a Java class Calculator that provides different ways to perform addition. Include three methods: The first method takes two integer numbers, the second method takes three integer numbers, and the third method takes two double numbers. In the main method, create an object of Calculator class.

Code:

```
7   package calculator;
8
9   class Calculator{
10      public int addition(int a, int b){
11          return a+b;
12      }
13      public int addition(int a, int b, int c){
14          return a+b+c;
15      }
16      public double addition(double a, double b){
17          return a+b;
18      }
19  }
20
21  public class Main {
22      public static void main(String[] args) {
23          Calculator cl = new Calculator();
24
25          System.out.println("add two integer number: " + cl.addition(3, 6));
26          System.out.println("add three integer number: " + cl.addition(3, 6, 5));
27          System.out.println("add two double number: " + cl.addition(3.5, 6.6));
28      }
29  }
30
```

calculator.Calculator > addition >

Output - Polymorphism (run)

```
run:
add two integer number: 9
add three integer number: 14
add two double number: 10.1
BUILD SUCCESSFUL (total time: 0 seconds)
```

Problem 2. Create a Java class Shape that provides different ways to calculate the area. Include three methods: the first method takes one parameter (side length) to calculate the area of a square, the second method takes two parameters (length and width) to calculate the area of a rectangle, and the third method takes one decimal parameter (radius) to calculate the area of a circle. In the main method, create an object of Shape class.

Code :

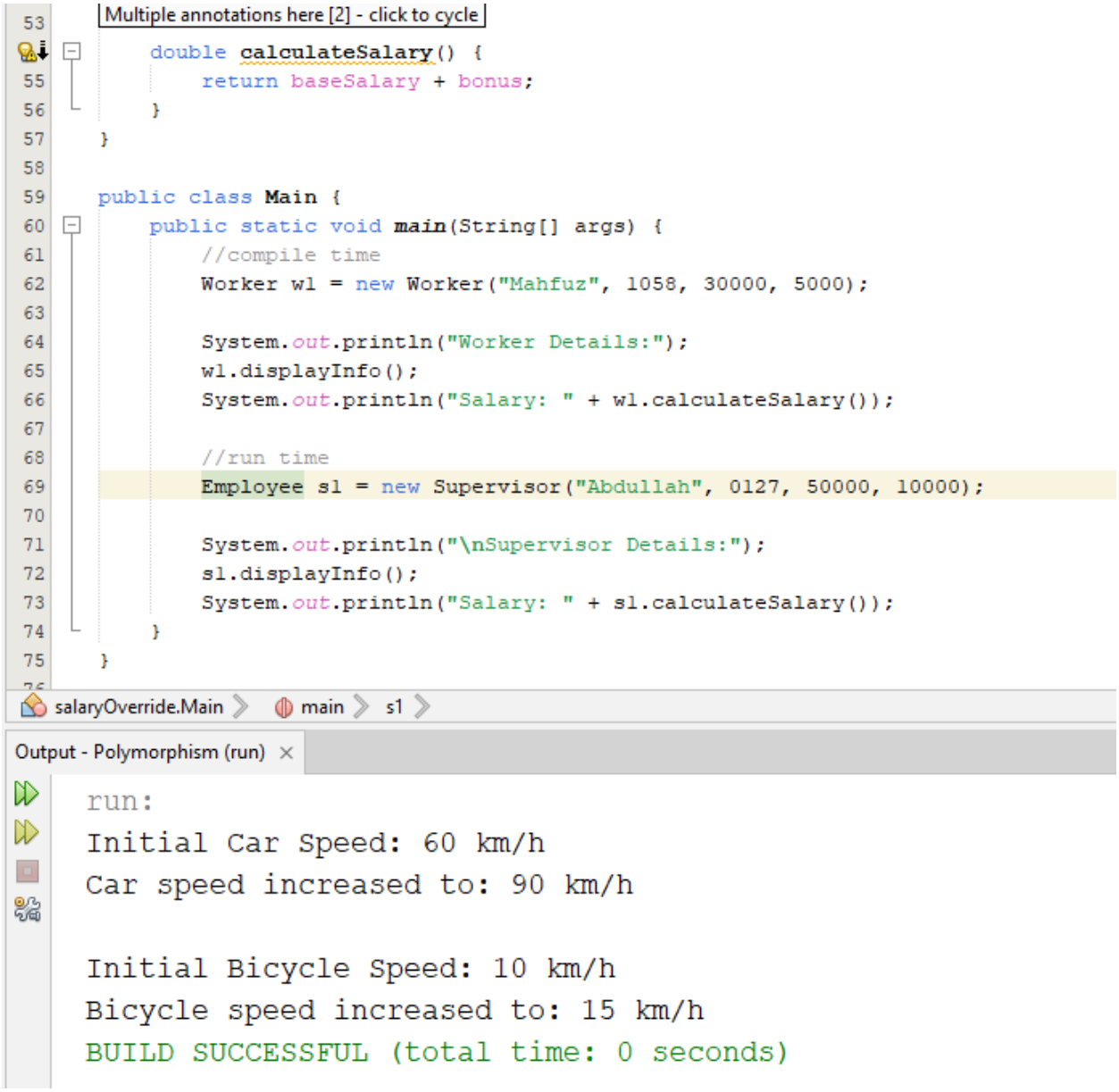
```
Source  History
 9  package areaCalculator;
10
11  class Shape{
12      public void areaCalculate(double length){
13          System.out.println("Area of Squar: " + (length*length) );
14      }
15      public void areaCalculate(double length,double width){
16          System.out.println("Area of rectangle: " + (length*width) );
17      }
18      public void areaCalculate(float radius){
19          System.out.println("Area of circle: " + ((radius*radius)*3.1416f) );
20      }
21  }
22
23  public class Main {
24      public static void main(String[] args) {
25          Shape s =new Shape();
26
27          s.areaCalculate(5.6);
28          s.areaCalculate(5,6);
29          s.areaCalculate(5.5f);
30      }
31  }
32
areaCalculator.Shape  areaCalculate
```

```
Output - Polymorphism (run)
run:
Area of Squar: 31.359999999999996
Area of rectangle: 30.0
Area of circle: 95.033394
BUILD SUCCESSFUL (total time: 0 seconds)
```

Problem 3. Write a Java program to define a class Employee with instance variables name and id, along with a method calculateSalary(). Create two subclasses, Worker and Supervisor, each having additional instance variables baseSalary and bonus. In both subclasses, override the calculateSalary() method to compute and return the salary.

Code:

```
Source  History
8   package salaryOverride;
9
11  class Employee {
12      String name;
13      int id;
14
15      Employee(String name, int id) {
16          this.name = name;
17          this.id = id;
18      }
20      double calculateSalary() {
21          return 0.0;
22      }
23      void displayInfo() {
24          System.out.println("Name: " + name);
25          System.out.println("ID: " + id);
26      }
27  }
28
29  class Worker extends Employee {
30      double baseSalary;
31      double bonus;
32
33      Worker(String name, int id, double baseSalary, double bonus) {
34          super(name, id);
35          this.baseSalary = baseSalary;
36          this.bonus = bonus;
37      }
38
40      double calculateSalary() {
41          return baseSalary + bonus;
42      }
43  }
44
45  class Supervisor extends Employee {
46      double baseSalary;
47      double bonus;
48
49      Supervisor(String name, int id, double baseSalary, double bonus) {
50          super(name, id);
51          this.baseSalary = baseSalary;
52          this.bonus = bonus;
53      }
```



Problem 4. Write a Java program to define a class Vehicle with a method speedUp(). Create two subclasses: Car and Bicycle, each having an instance variable currentSpeed. In both subclasses, override the speedUp() method to increase the vehicle's speed differently.

Code:

```java
package Vehicle;

class Vehicle {
    void speedUp() {
        System.out.println("Vehicle is speeding up...");
    }
}

class Car extends Vehicle {
    int currentSpeed;

    Car(int speed) {
        this.currentSpeed = speed;
    }

    void speedUp() {
        currentSpeed += 30;
        System.out.println("Car speed increased to: " + currentSpeed + " km/h");
    }
}

class Bicycle extends Vehicle {
    int currentSpeed;

    Bicycle(int speed) {
        this.currentSpeed = speed;
    }

    void speedUp() {
        currentSpeed += 5;
        System.out.println("Bicycle speed increased to: " + currentSpeed + " km/h");
    }
}

public class Main {
    public static void main(String[] args) {
        Car car = new Car(60);
        System.out.println("Initial Car Speed: " + car.currentSpeed + " km/h");
        car.speedUp();

        Bicycle bicycle = new Bicycle(10);
        System.out.println("\nInitial Bicycle Speed: " + bicycle.currentSpeed + " km/h");
        bicycle.speedUp();
    }
}
```

Output - Polymorphism (run)

```
run:
Initial Car Speed: 60 km/h
Car speed increased to: 90 km/h

Initial Bicycle Speed: 10 km/h
Bicycle speed increased to: 15 km/h
BUILD SUCCESSFUL (total time: 0 seconds)
```