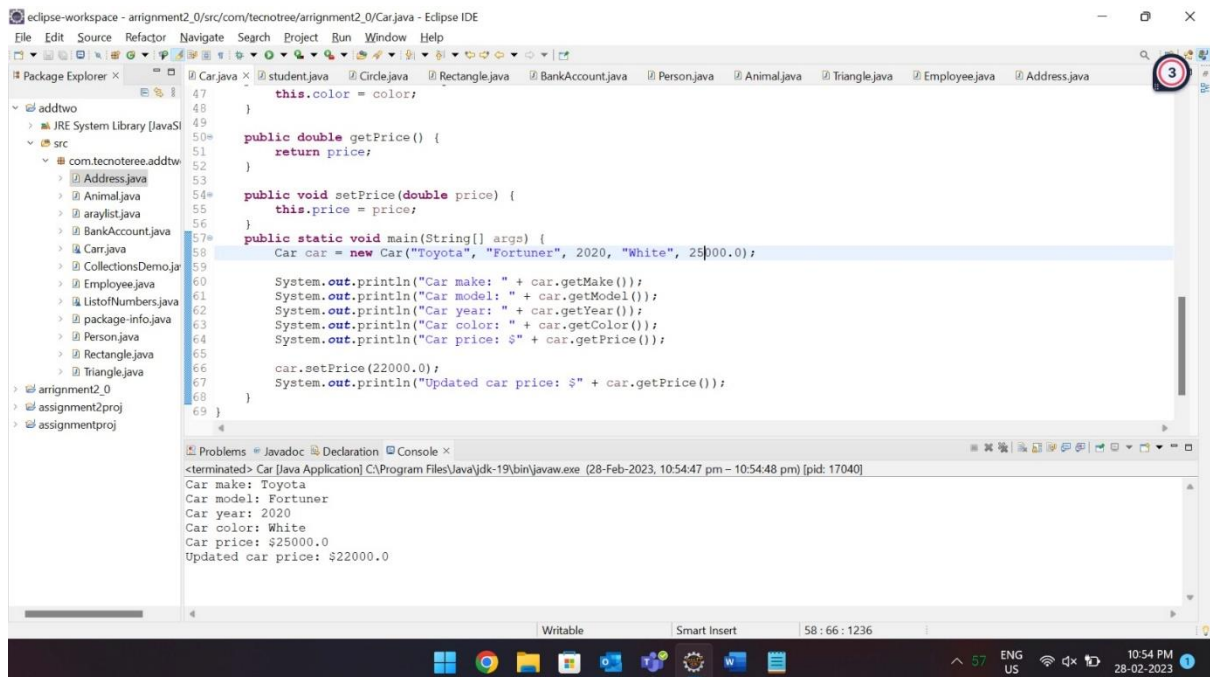


Java Assignment-2

Topic: Object-Oriented Programming (OOP) Concepts

Create a class called "Car" that has the following properties: make, model, year, color, and price. Include a constructor and getter and setter methods for each property.



The screenshot shows the Eclipse IDE with the following components:

- Package Explorer:** Shows a project named 'arrignment2_0' with a package 'com.tecnoteee.addtwo' containing several Java files, including 'Car.java'.
- Editor:** Displays the code for 'Car.java'. The code includes a constructor, getter and setter methods for 'make', 'model', 'year', 'color', and 'price', and a static 'main' method that creates a 'Car' object and prints its details.
- Console:** Shows the output of the program, which prints the details of the 'Car' object and the updated price.

```
47     this.color = color;
48 }
49
50 public double getPrice() {
51     return price;
52 }
53
54 public void setPrice(double price) {
55     this.price = price;
56 }
57
58 public static void main(String[] args) {
59     Car car = new Car("Toyota", "Fortuner", 2020, "White", 25000.0);
60
61     System.out.println("Car make: " + car.getMake());
62     System.out.println("Car model: " + car.getModel());
63     System.out.println("Car year: " + car.getYear());
64     System.out.println("Car color: " + car.getColor());
65     System.out.println("Car price: $" + car.getPrice());
66
67     car.setPrice(22000.0);
68     System.out.println("Updated car price: $" + car.getPrice());
69 }
```

Console Output:

```
<terminated> Car [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (28-Feb-2023, 10:54:47 pm - 10:54:48 pm) [pid: 17040]
Car make: Toyota
Car model: Fortuner
Car year: 2020
Car color: White
Car price: $25000.0
Updated car price: $22000.0
```

CODESHARE LINK: <https://codeshare.io/IonJAY>

2. Create a class called "Student" that has the following properties: name, age, gender, grade, and GPA. Include a constructor and getter and setter methods for each property.

The screenshot shows the Eclipse IDE with the `Student.java` file open. The code defines a `Student` class with attributes `name`, `age`, `gender`, `grade`, and `GPA`. It includes methods `setName`, `setAge`, `setGender`, `setGrade`, `setGpa`, `getName`, `getAge`, `getGender`, `getGrade`, and `getGpa`. A `main` method creates a `Student` object, prints its details, updates the GPA, and prints the updated details. The console output shows the initial and updated student information.

```
46 public void setName(String name) {
47     this.name = name;
48 }
49
50 public void setAge(int age) {
51     this.age = age;
52 }
53
54 public void setGender(String gender) {
55     this.gender = gender;
56 }
57
58 public void setGrade(int grade) {
59     this.grade = grade;
60 }
61
62 public double getGpa() {
63     return gpa;
64 }
65
66 public void setGpa(double gpa) {
67     this.gpa = gpa;
68 }
69
70 public static void main(String[] args) {
71     Student student = new Student("Raju", 18, "Male", 11, 3.8);
72
73     System.out.println("Student name: " + student.getName());
74     System.out.println("Student age: " + student.getAge());
75     System.out.println("Student gender: " + student.getGender());
76     System.out.println("Student grade: " + student.getGrade());
77     System.out.println("Student GPA: " + student.getGpa());
78
79     student.setGpa(4.0);
80     System.out.println("Updated student GPA: " + student.getGpa());
81 }
```

Console Output:

```
<terminated> student [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (28-Feb-2023, 10:55:51 pm - 10:55:52 pm) [pid: 1452]
Student name: Raju
Student age: 18
Student gender: Male
Student grade: 11
Student GPA: 3.8
Updated student GPA: 4.0
```

CODESHARE LINK: <https://codeshare.io/r9lwPV>

3.Create a class called "Circle" that has the following properties: radius, diameter, and area. Include a constructor and methods to calculate the diameter and area of the circle.

The screenshot shows the Eclipse IDE with the `Circle.java` file open. The code defines a `Circle` class with attributes `radius`, `diameter`, and `area`. It includes a constructor `Circle` and methods `getRadius`, `getDiameter`, `getArea`, and `main`. The `main` method creates a `Circle` object and prints its details. The console output shows the calculated radius, diameter, and area.

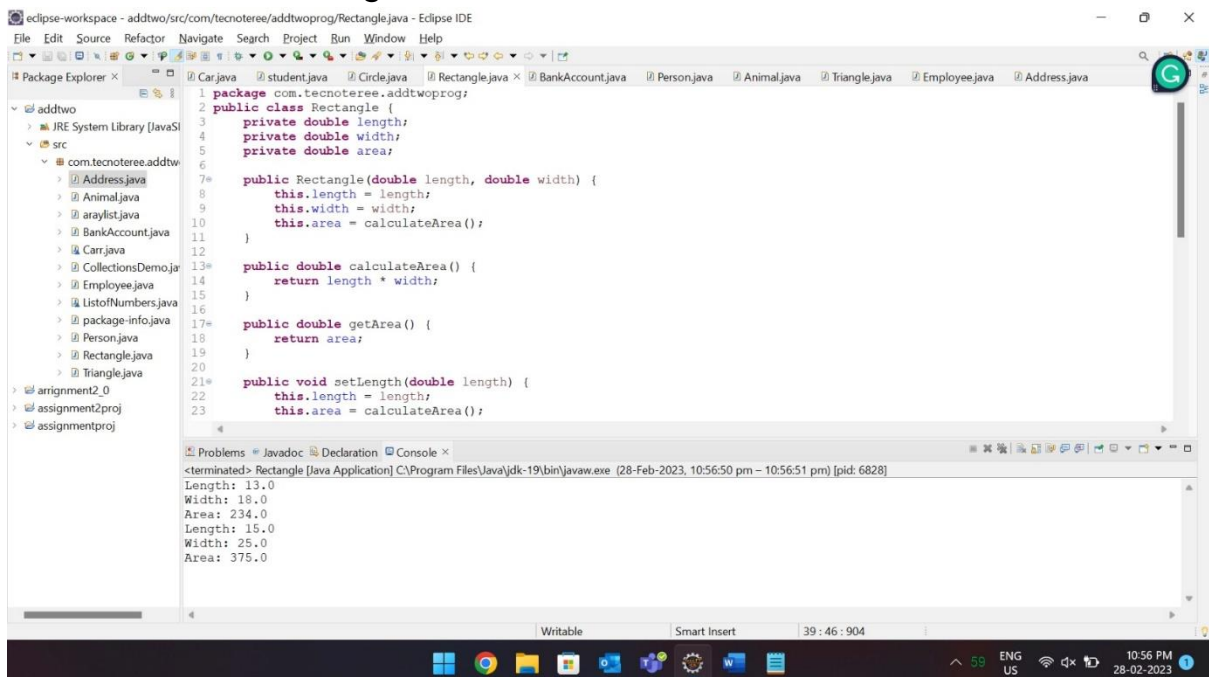
```
11 this.area = Math.PI * Math.pow(radius, 2);
12 }
13
14 public double getRadius() {
15     return radius;
16 }
17
18 public double getDiameter() {
19     return diameter;
20 }
21
22 public double getArea() {
23     return area;
24 }
25
26 public static void main(String[] args) {
27     Circle circle = new Circle(9.0);
28
29     System.out.println("Circle radius: " + circle.getRadius());
30     System.out.println("Circle diameter: " + circle.getDiameter());
31     System.out.println("Circle area: " + circle.getArea());
32 }
33 }
```

Console Output:

```
<terminated> Circle [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (28-Feb-2023, 10:56:16 pm - 10:56:17 pm) [pid: 18748]
Circle radius: 9.0
Circle diameter: 18.0
Circle area: 254.46900494077323
```

CODESHARE LINK: <https://codeshare.io/4eonqE>

4. Create a class called "Rectangle" that has the following properties: length, width, and area. Include a constructor and a method to calculate the area of the rectangle.



The screenshot shows the Eclipse IDE with the `Rectangle.java` file open. The code defines a `Rectangle` class with private attributes `length`, `width`, and `area`. It includes a constructor `Rectangle(double length, double width)` that initializes these values and calls `calculateArea()`. The `calculateArea()` method returns `length * width`. There are also `getArea()` and `setLength(double length)` methods. The console output shows the results of running the application: `Length: 13.0`, `Width: 18.0`, `Area: 234.0`, `Length: 15.0`, `Width: 25.0`, and `Area: 375.0`.

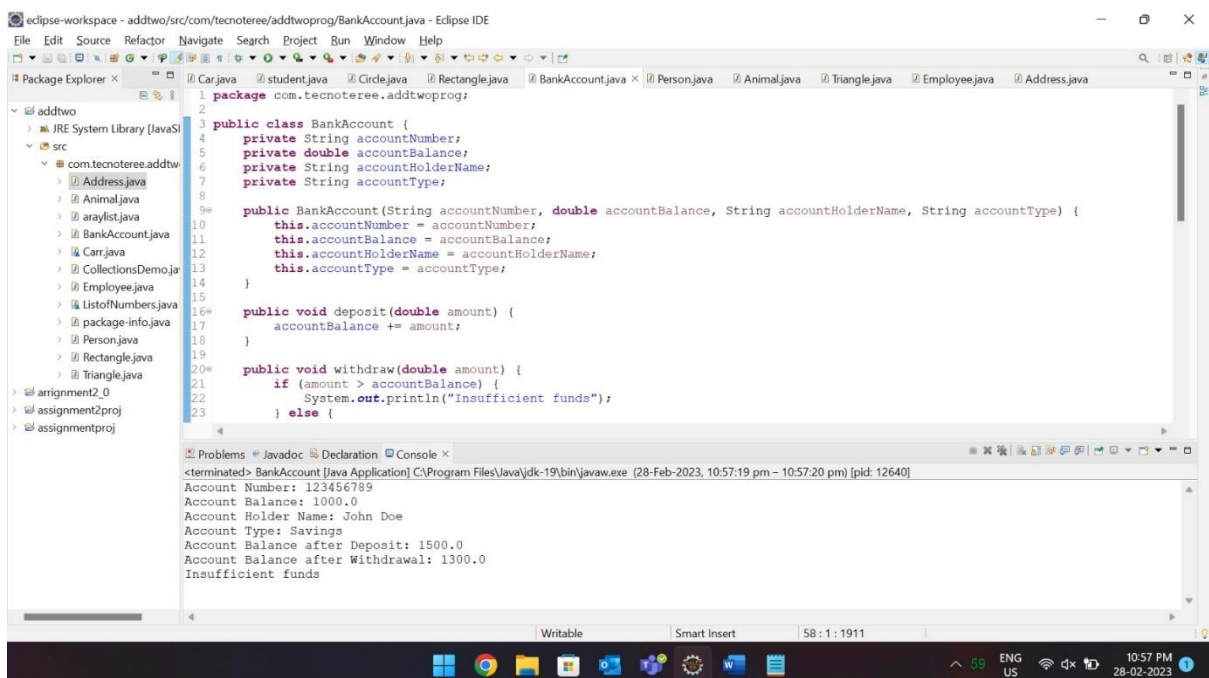
```
1 package com.tecnotereee.addtwoprogram;
2 public class Rectangle {
3     private double length;
4     private double width;
5     private double area;
6
7     public Rectangle(double length, double width) {
8         this.length = length;
9         this.width = width;
10        this.area = calculateArea();
11    }
12
13    public double calculateArea() {
14        return length * width;
15    }
16
17    public double getArea() {
18        return area;
19    }
20
21    public void setLength(double length) {
22        this.length = length;
23        this.area = calculateArea();
24    }
25 }
```

Console Output:

```
<terminated> Rectangle [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (28-Feb-2023, 10:56:50 pm - 10:56:51 pm) [pid: 6828]
Length: 13.0
Width: 18.0
Area: 234.0
Length: 15.0
Width: 25.0
Area: 375.0
```

CODESHARE LINK: <https://codeshare.io/VZENm8>

5. Create a class called "BankAccount" that has the following properties: account number, account balance, account holder name, and account type. Include a constructor and methods to deposit and withdraw money from the account.



The screenshot shows the Eclipse IDE with the `BankAccount.java` file open. The code defines a `BankAccount` class with private attributes `accountNumber`, `accountBalance`, `accountHolderName`, and `accountType`. It includes a constructor `BankAccount(String accountNumber, double accountBalance, String accountHolderName, String accountType)` that initializes these values. There are also `deposit(double amount)` and `withdraw(double amount)` methods. The `withdraw` method includes a check for `Insufficient funds` if the withdrawal amount is greater than the current balance. The console output shows the results of running the application: `Account Number: 123456789`, `Account Balance: 1000.0`, `Account Holder Name: John Doe`, `Account Type: Savings`, `Account Balance after Deposit: 1500.0`, `Account Balance after Withdrawal: 1300.0`, and `Insufficient funds`.

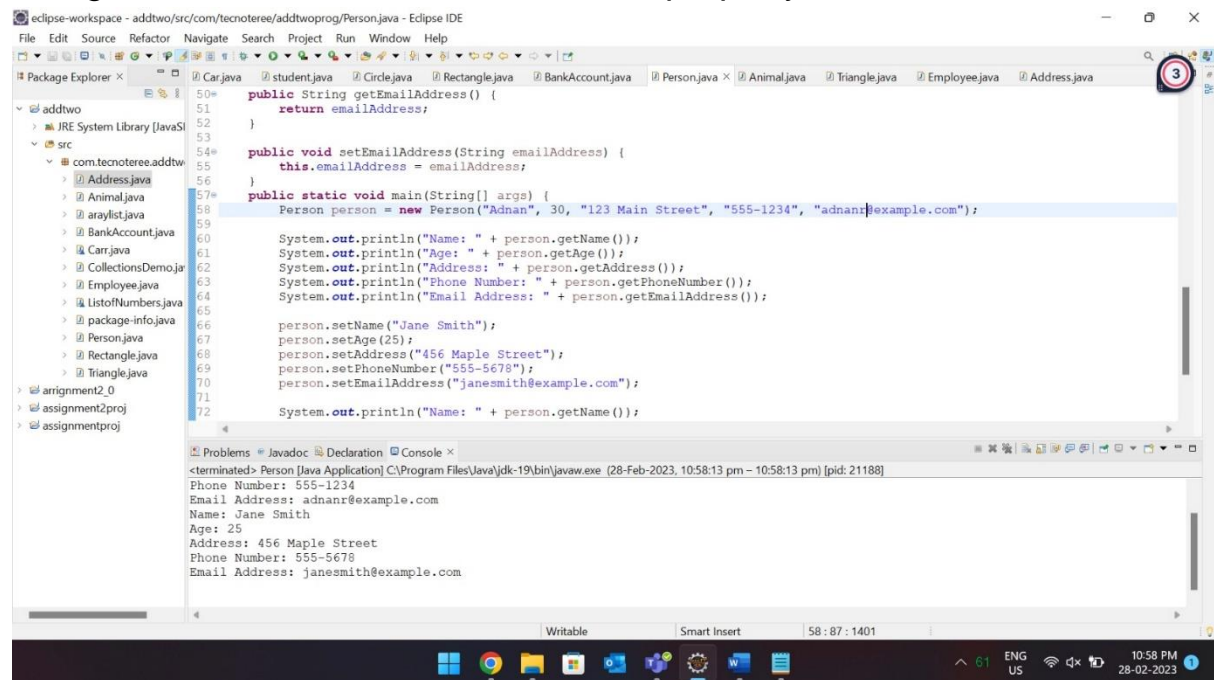
```
1 package com.tecnotereee.addtwoprogram;
2
3 public class BankAccount {
4     private String accountNumber;
5     private double accountBalance;
6     private String accountHolderName;
7     private String accountType;
8
9     public BankAccount(String accountNumber, double accountBalance, String accountHolderName, String accountType) {
10        this.accountNumber = accountNumber;
11        this.accountBalance = accountBalance;
12        this.accountHolderName = accountHolderName;
13        this.accountType = accountType;
14    }
15
16    public void deposit(double amount) {
17        accountBalance += amount;
18    }
19
20    public void withdraw(double amount) {
21        if (amount > accountBalance) {
22            System.out.println("Insufficient funds");
23        } else {
24            accountBalance -= amount;
25        }
26    }
27 }
```

Console Output:

```
<terminated> BankAccount [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (28-Feb-2023, 10:57:19 pm - 10:57:20 pm) [pid: 12640]
Account Number: 123456789
Account Balance: 1000.0
Account Holder Name: John Doe
Account Type: Savings
Account Balance after Deposit: 1500.0
Account Balance after Withdrawal: 1300.0
Insufficient funds
```

CODESHARE LINK: <https://codeshare.io/qPm6lk>

6. Create a class called "Person" that has the following properties: name, age, address, phone number, and email address. Include a constructor and getter and setter methods for each property.



The screenshot shows the Eclipse IDE with the 'Person.java' file open. The code defines a 'Person' class with private attributes 'name', 'age', 'address', 'phoneNumber', and 'emailAddress'. It includes a constructor, getter methods, and setter methods. The 'main' method creates a 'Person' object and prints its details. The console output shows the execution results.

```
public class Person {
    private String name;
    private int age;
    private String address;
    private String phoneNumber;
    private String emailAddress;

    public Person(String name, int age, String address, String phoneNumber, String emailAddress) {
        this.name = name;
        this.age = age;
        this.address = address;
        this.phoneNumber = phoneNumber;
        this.emailAddress = emailAddress;
    }

    public String getName() {
        return name;
    }

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

    public int getAge() {
        return age;
    }

    public void setAge(int age) {
        this.age = age;
    }

    public String getAddress() {
        return address;
    }

    public void setAddress(String address) {
        this.address = address;
    }

    public String getPhoneNumber() {
        return phoneNumber;
    }

    public void setPhoneNumber(String phoneNumber) {
        this.phoneNumber = phoneNumber;
    }

    public String getEmailAddress() {
        return emailAddress;
    }

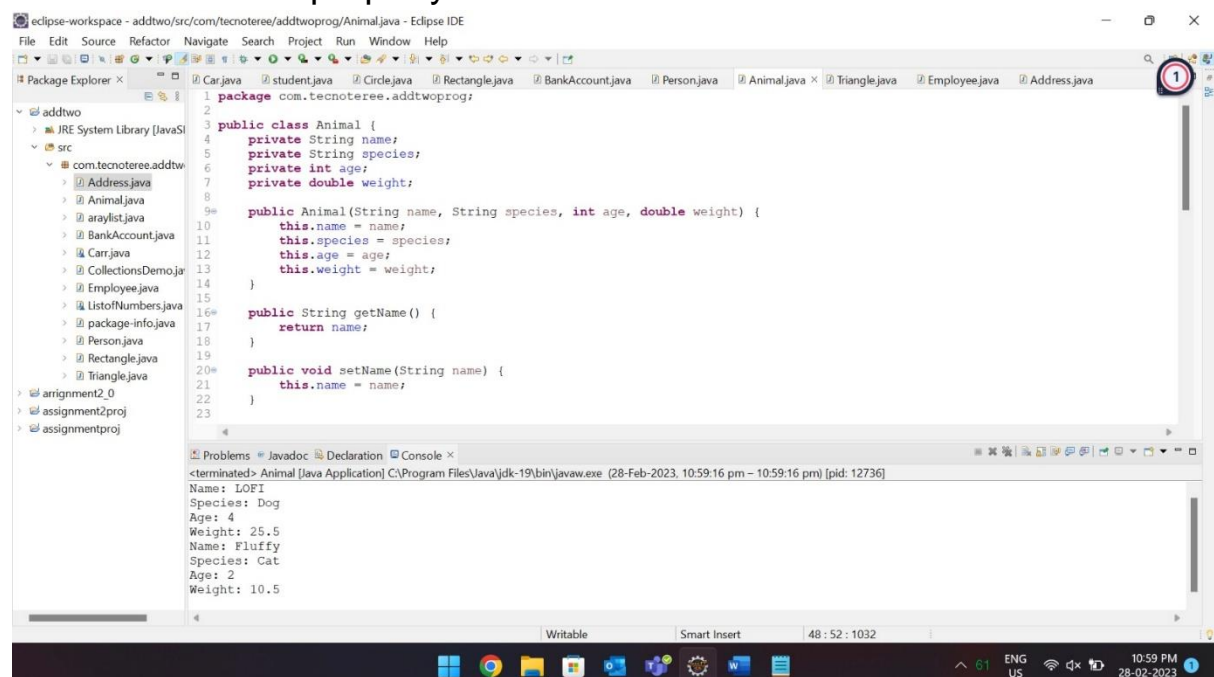
    public void setEmailAddress(String emailAddress) {
        this.emailAddress = emailAddress;
    }
}
```

Console Output:

```
<terminated> Person [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (28-Feb-2023, 10:58:13 pm - 10:58:13 pm) [pid: 21188]
Phone Number: 555-1234
Email Address: adnan@example.com
Name: Jane Smith
Age: 25
Address: 456 Maple Street
Phone Number: 555-5678
Email Address: janesmith@example.com
```

CODESHARE LINK: <https://codeshare.io/MNEzgg>

7. Create a class called "Animal" that has the following properties: name, species, age, and weight. Include a constructor and getter and setter methods for each property.



The screenshot shows the Eclipse IDE with the 'Animal.java' file open. The code defines an 'Animal' class with private attributes 'name', 'species', 'age', and 'weight'. It includes a constructor, a getter method, and a setter method. The 'main' method creates an 'Animal' object and prints its details. The console output shows the execution results.

```
package com.tecnoteer.addtwoprog;

public class Animal {
    private String name;
    private String species;
    private int age;
    private double weight;

    public Animal(String name, String species, int age, double weight) {
        this.name = name;
        this.species = species;
        this.age = age;
        this.weight = weight;
    }

    public String getName() {
        return name;
    }

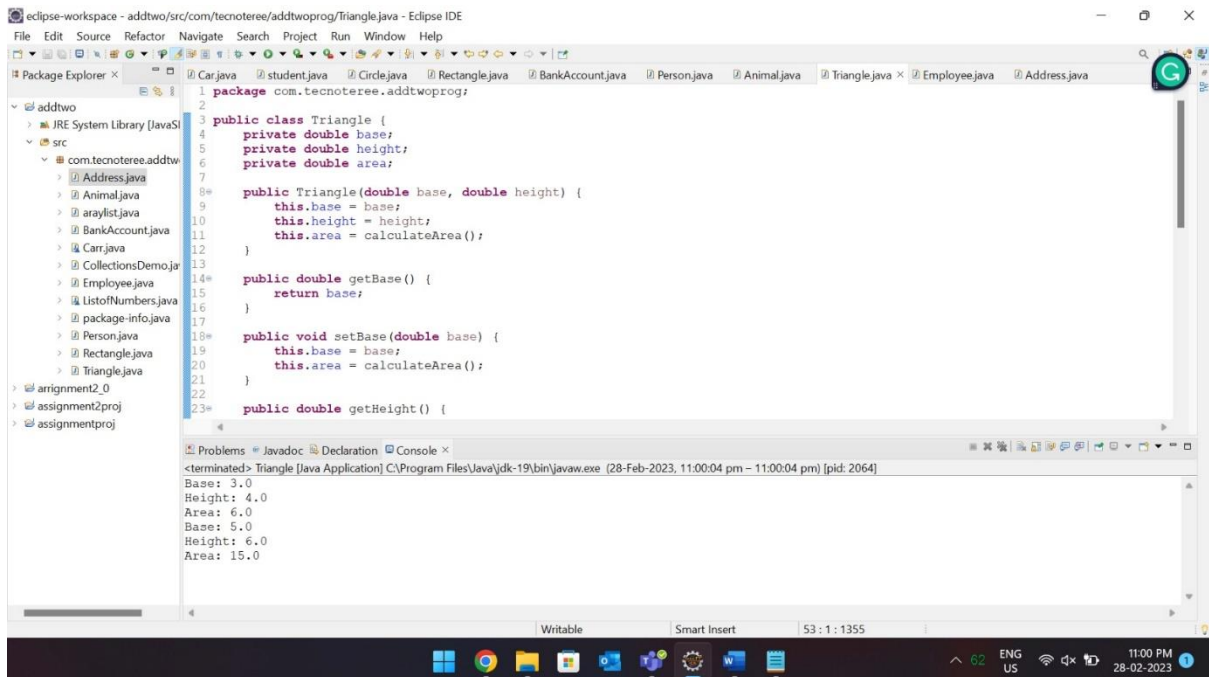
    public void setName(String name) {
        this.name = name;
    }
}
```

Console Output:

```
<terminated> Animal [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (28-Feb-2023, 10:59:16 pm - 10:59:16 pm) [pid: 12736]
Name: LOFI
Species: Dog
Age: 4
Weight: 25.5
Name: Fluffy
Species: Cat
Age: 2
Weight: 10.5
```

CODESHARE LINK: <https://codeshare.io/xv4nqD>

8. Create a class called "Triangle" that has the following properties: base, height, and area. Include a constructor and a method to calculate the area of the triangle.



The screenshot shows the Eclipse IDE with the following code in `Triangle.java`:

```
1 package com.tecnoteree.addtwoprogram;  
2  
3 public class Triangle {  
4     private double base;  
5     private double height;  
6     private double area;  
7  
8     public Triangle(double base, double height) {  
9         this.base = base;  
10        this.height = height;  
11        this.area = calculateArea();  
12    }  
13  
14    public double getBase() {  
15        return base;  
16    }  
17  
18    public void setBase(double base) {  
19        this.base = base;  
20        this.area = calculateArea();  
21    }  
22  
23    public double getHeight() {
```

The console output shows the results of running the program:

```
<terminated> Triangle [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (28-Feb-2023, 11:00:04 pm - 11:00:04 pm) [pid: 2064]  
Base: 3.0  
Height: 4.0  
Area: 6.0  
Base: 5.0  
Height: 6.0  
Area: 15.0
```

CODESHARE LINK: <https://codeshare.io/YLE3BL>

9. Create a class called "Employee" that has the following properties: name, employee ID, department, job title, and salary. Include a constructor and getter and setter methods for each property.

The screenshot shows the Eclipse IDE with the 'Employee.java' file open. The code defines a class 'Employee' with private attributes: name, employeeID, department, jobTitle, and salary. It includes a constructor and two methods: 'getName()' and 'setName()'. The console output shows the results of running the application: Name: Riya, Employee ID: 4444, Department: IT, Job Title: Software Engineer, Salary: 70000.0, and Salary: 85000.0.

```
package com.tecnoteeee.addtwoprog;

public class Employee {
    private String name;
    private int employeeID;
    private String department;
    private String jobTitle;
    private double salary;

    public Employee(String name, int employeeID, String department, String jobTitle, double salary) {
        this.name = name;
        this.employeeID = employeeID;
        this.department = department;
        this.jobTitle = jobTitle;
        this.salary = salary;
    }

    public String getName() {
        return name;
    }

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

Console Output:

```
<terminated> Employee [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (28-Feb-2023, 11:26:18 pm - 11:26:22 pm) [pid: 22720]
Name: Riya
Employee ID: 4444
Department: IT
Job Title: Software Engineer
Salary: 70000.0
Salary: 85000.0
```

CODESHARE LINK: <https://codeshare.io/OdE6AW>

10. Create a class called "Address" that has the following properties: street, city, state, zip code, and country. Include a constructor and getter and setter methods for each property.

The screenshot shows the Eclipse IDE with the 'Address.java' file open. The code defines a class 'Address' with private attributes: street, city, state, zipCode, and country. It includes a constructor and three methods: 'getStreet()', 'setStreet()', and 'setCity()'. The console output shows the results of running the application: Street: 123 Main St, City: Anytown, State: CA, Zip Code: 12345, and Country: INDIA.

```
package com.tecnoteeee.addtwoprog;

public class Address {
    private String street;
    private String city;
    private String state;
    private String zipCode;
    private String country;

    public Address(String street, String city, String state, String zipCode, String country) {
        this.street = street;
        this.city = city;
        this.state = state;
        this.zipCode = zipCode;
        this.country = country;
    }

    public String getStreet() {
        return street;
    }

    public void setStreet(String street) {
        this.street = street;
    }

    public void setCity(String city) {
        this.city = city;
    }
}
```

Console Output:

```
<terminated> Address [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (28-Feb-2023, 11:01:47 pm - 11:01:48 pm) [pid: 19388]
Street: 123 Main St
City: Anytown
State: CA
Zip Code: 12345
Country: INDIA
Zip Code: 510001
```

CODESHARE LINK: <https://codeshare.io/6pkoXY>

