Academic Year: 2023-24 **Semester: II** **Class: FYMCA**

Course Code: MC506 **Course Name: Java Programming**

Name: Durgesh Dilip Mandge **UCID : 2023510032**

Experiment No.1.3

Date: 05-02-2024

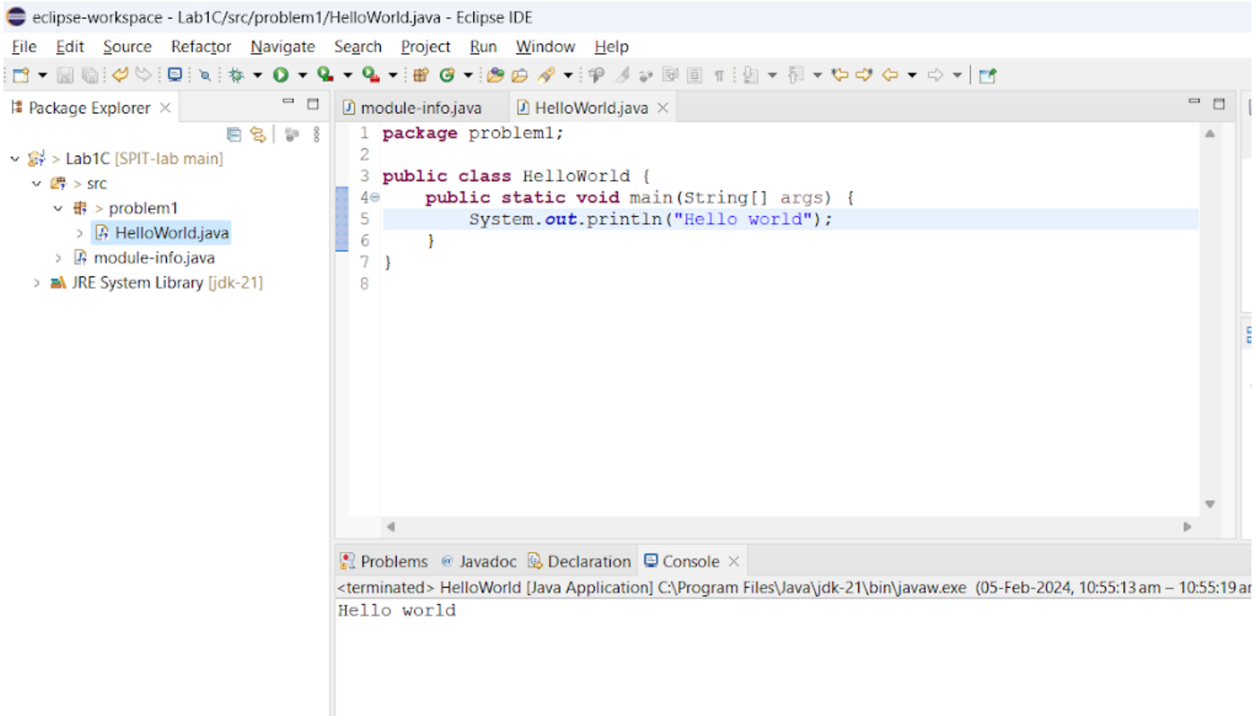
Aim: Fundamentals of Java Programming
CO Mapping – CO 1

Objective:

- To understand declaration of Classes, and Methods with its all features such as Constructors, Access Specifier
- To understand Classes, Instance variables, Methods, Constructors, Access
- Specifiers as basic fundamentals
- Implement Abstract Classes and Wrapper Classes for given problem statement
- Design and implement Inheritance, Polymorphism in JAVA
- Demonstrate Use of Static, final, super and this keyword
- Demonstrate creating user defined package, Access control protection,
- Defining interface, Implementing interface

Lab Exercise:Fundamentals of java programming using Eclipse- IDE.

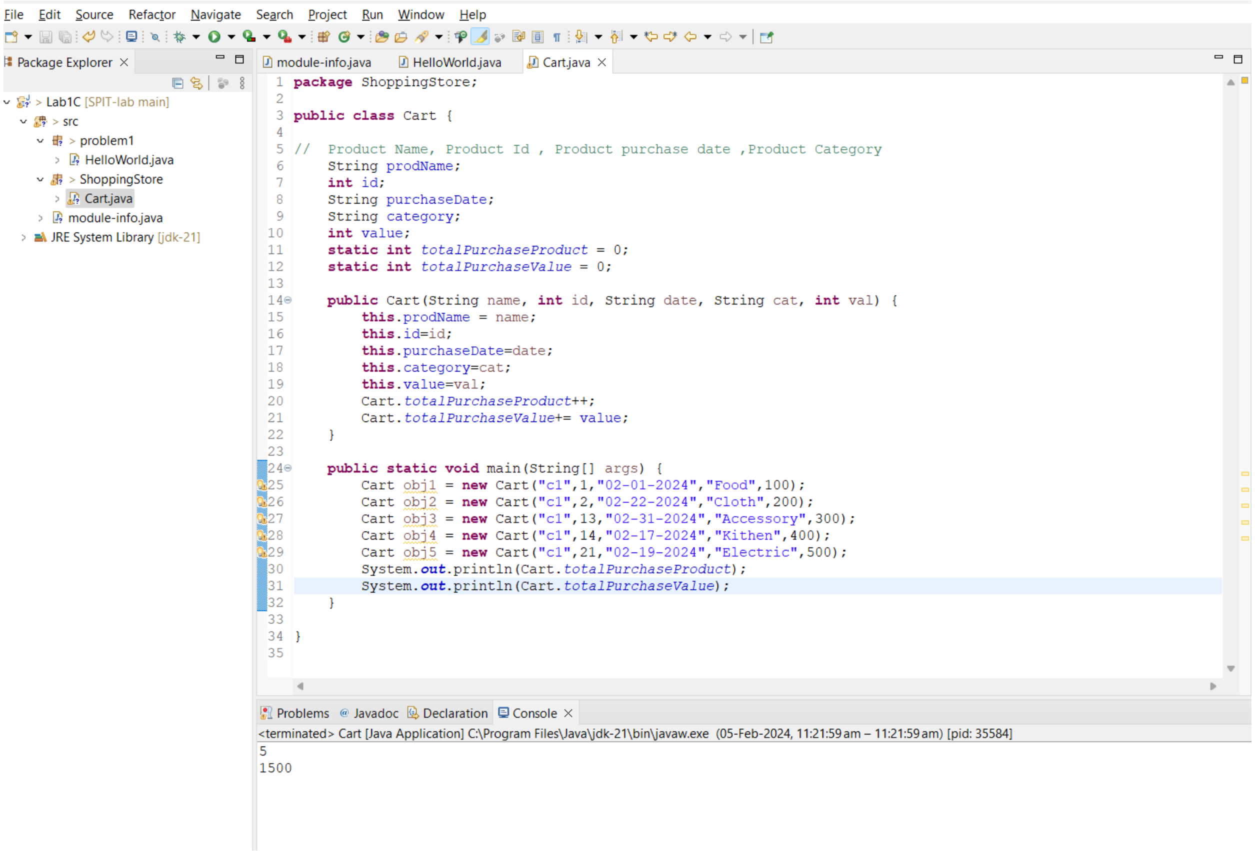
1. **W.A.P on to print " Hello World ".**


eclipse-workspace - Lab1C/src/problem1/HelloWorld.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
Lab1C [SPIT-lab main]
src
problem1
HelloWorld.java
module-info.java
JRE System Library [jdk-21]
module-info.java
HelloWorld.java
Problems
Javadoc
Declaration
Console

```java
package problem1;

public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello world");
    }
}
```

```
<terminated> HelloWorld [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe  (05-Feb-2024, 10:55:13 am – 10:55:19 ar
Hello world
```

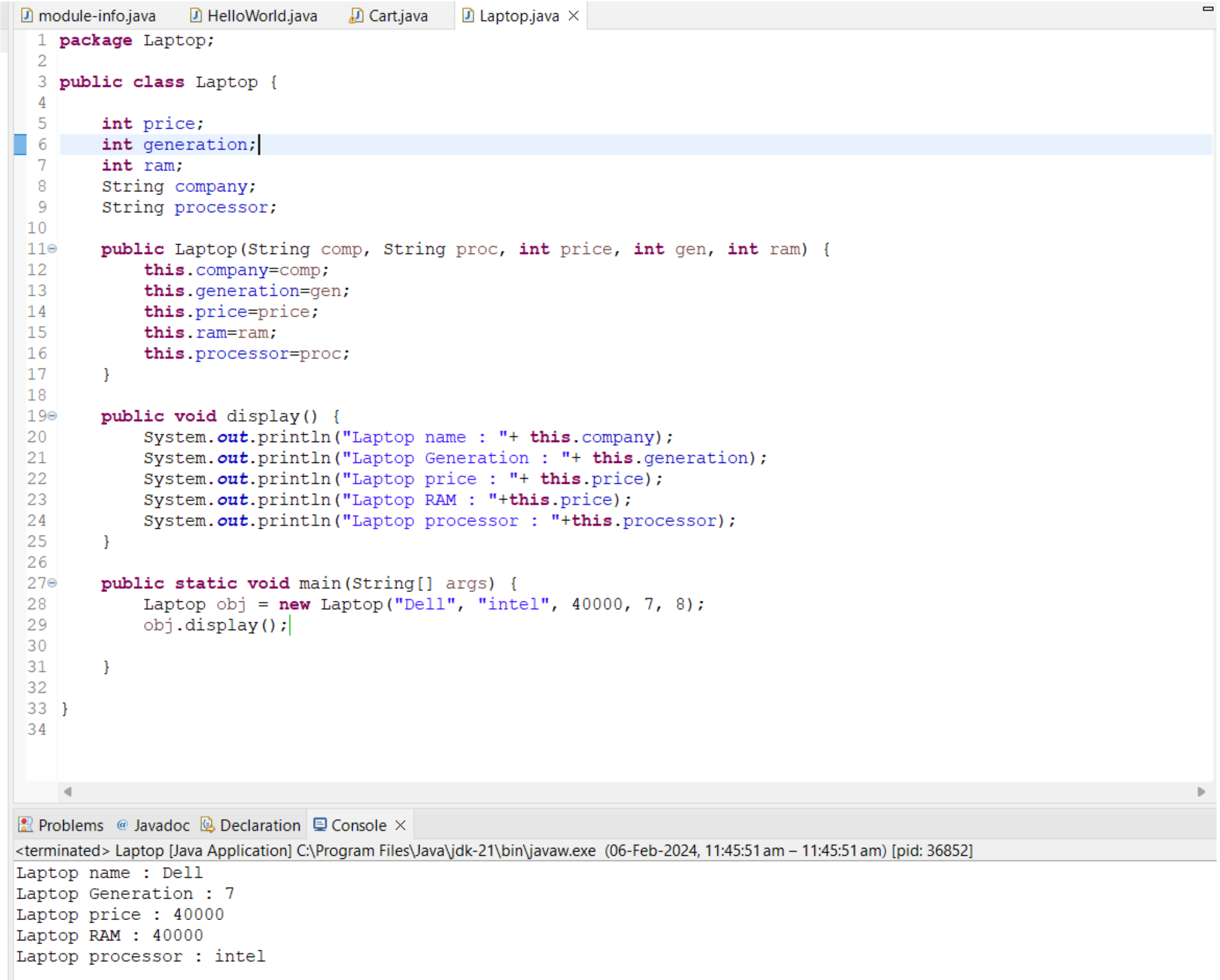
2. Create a class cart in a shopping store. The class should have Product Name, Product Id , Product purchase date ,Product Category (Apply respective data structure).Implement a method to add Total Products purchased and Final values of Each day purchase. Use appropriate Objects and Constructors wherever needed.



```java
package ShoppingStore;

public class Cart {

//  Product Name, Product Id , Product purchase date ,Product Category
    String prodName;
    int id;
    String purchaseDate;
    String category;
    int value;
    static int totalPurchaseProduct = 0;
    static int totalPurchaseValue = 0;

    public Cart(String name, int id, String date, String cat, int val) {
        this.prodName = name;
        this.id=id;
        this.purchaseDate=date;
        this.category=cat;
        this.value=val;
        Cart.totalPurchaseProduct++;
        Cart.totalPurchaseValue+= value;
    }

    public static void main(String[] args) {
        Cart obj1 = new Cart("c1",1,"02-01-2024","Food",100);
        Cart obj2 = new Cart("c1",2,"02-22-2024","Cloth",200);
        Cart obj3 = new Cart("c1",13,"02-31-2024","Accessory",300);
        Cart obj4 = new Cart("c1",14,"02-17-2024","Kithen",400);
        Cart obj5 = new Cart("c1",21,"02-19-2024","Electric",500);
        System.out.println(Cart.totalPurchaseProduct);
        System.out.println(Cart.totalPurchaseValue);
    }

}

```

```
<terminated> Cart [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe  (05-Feb-2024, 11:21:59 am – 11:21:59 am) [pid: 35584]
5
1500
```

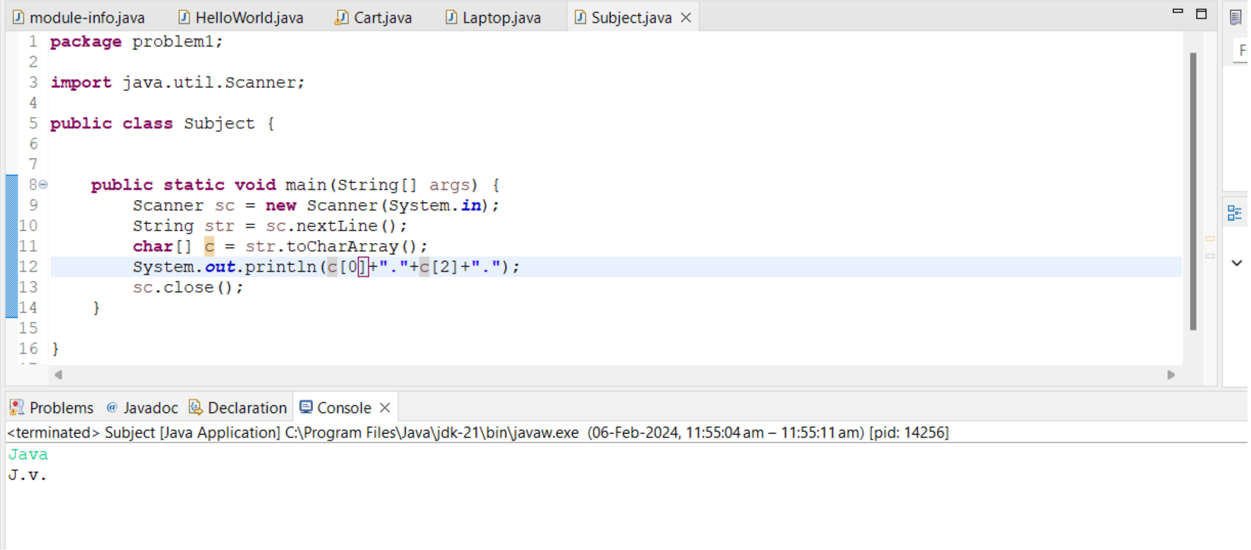
3. Create a class Laptop with specifications and assign suitable data types to its feature Make use of suitable methods and Constructors and display the output.



```java
package Laptop;

public class Laptop {

    int price;
    int generation;
    int ram;
    String company;
    String processor;

    public Laptop(String comp, String proc, int price, int gen, int ram) {
        this.company=comp;
        this.generation=gen;
        this.price=price;
        this.ram=ram;
        this.processor=proc;
    }

    public void display() {
        System.out.println("Laptop name : "+ this.company);
        System.out.println("Laptop Generation : "+ this.generation);
        System.out.println("Laptop price : "+ this.price);
        System.out.println("Laptop RAM : "+this.price);
        System.out.println("Laptop processor : "+this.processor);
    }

    public static void main(String[] args) {
        Laptop obj = new Laptop("Dell", "intel", 40000, 7, 8);
        obj.display();

    }

}
```

```
<terminated> Laptop [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe  (06-Feb-2024, 11:45:51 am – 11:45:51 am) [pid: 36852]
Laptop name : Dell
Laptop Generation : 7
Laptop price : 40000
Laptop RAM : 40000
Laptop processor : intel
```

4. Define a class subject . Take input string as name "Java " Show the o/p as J.V. using conversion to char[]. Then using CharAt() show the o/p.

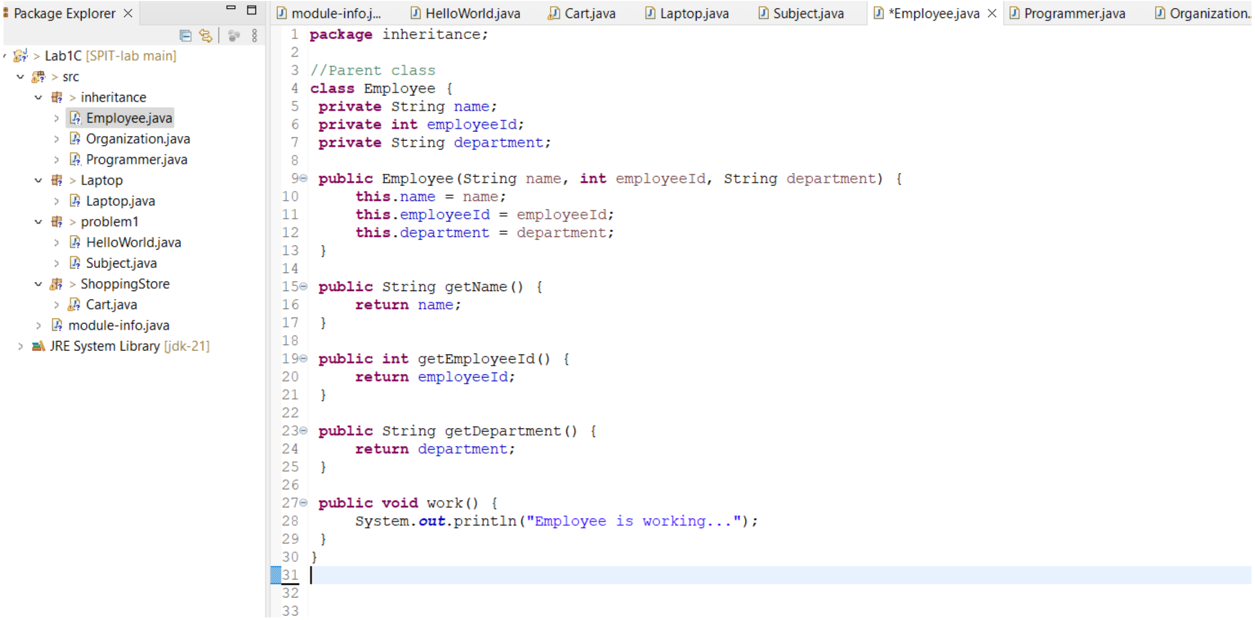


```java
package problem1;

import java.util.Scanner;

public class Subject {


    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String str = sc.nextLine();
        char[] c = str.toCharArray();
        System.out.println(c[0]+"."+c[2]+".");
        sc.close();
    }

}
```

```
<terminated> Subject [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe  (06-Feb-2024, 11:55:04 am – 11:55:11 am) [pid: 14256]
Java
J.v.
```

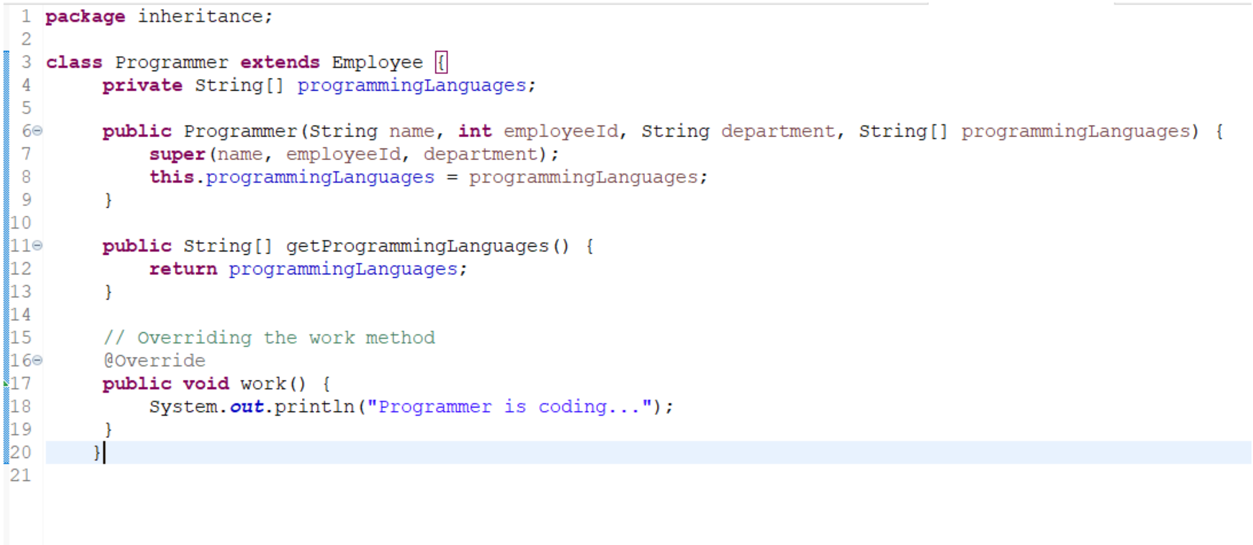
5. W.A.P based on Is- A relationship in a context of Organisation which have Employee & Programmer "Is-A" relationship.

Parent class



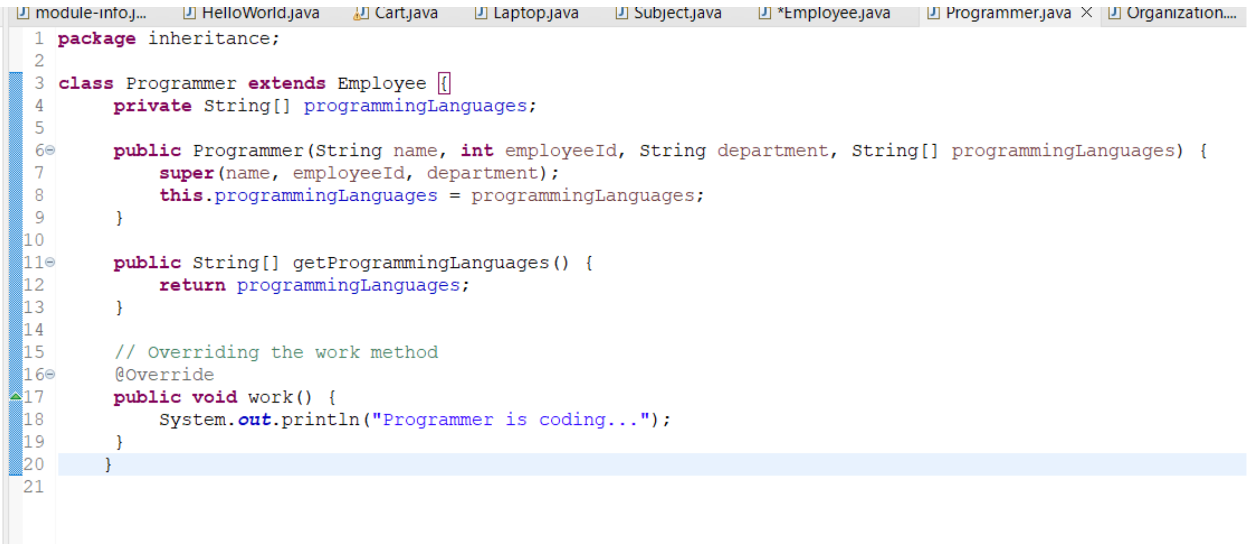
```java
package inheritance;

//Parent class
class Employee {
 private String name;
 private int employeeId;
 private String department;

 public Employee(String name, int employeeId, String department) {
     this.name = name;
     this.employeeId = employeeId;
     this.department = department;
 }

 public String getName() {
     return name;
 }

 public int getEmployeeId() {
     return employeeId;
 }

 public String getDepartment() {
     return department;
 }

 public void work() {
     System.out.println("Employee is working...");
 }
}
```

Child class



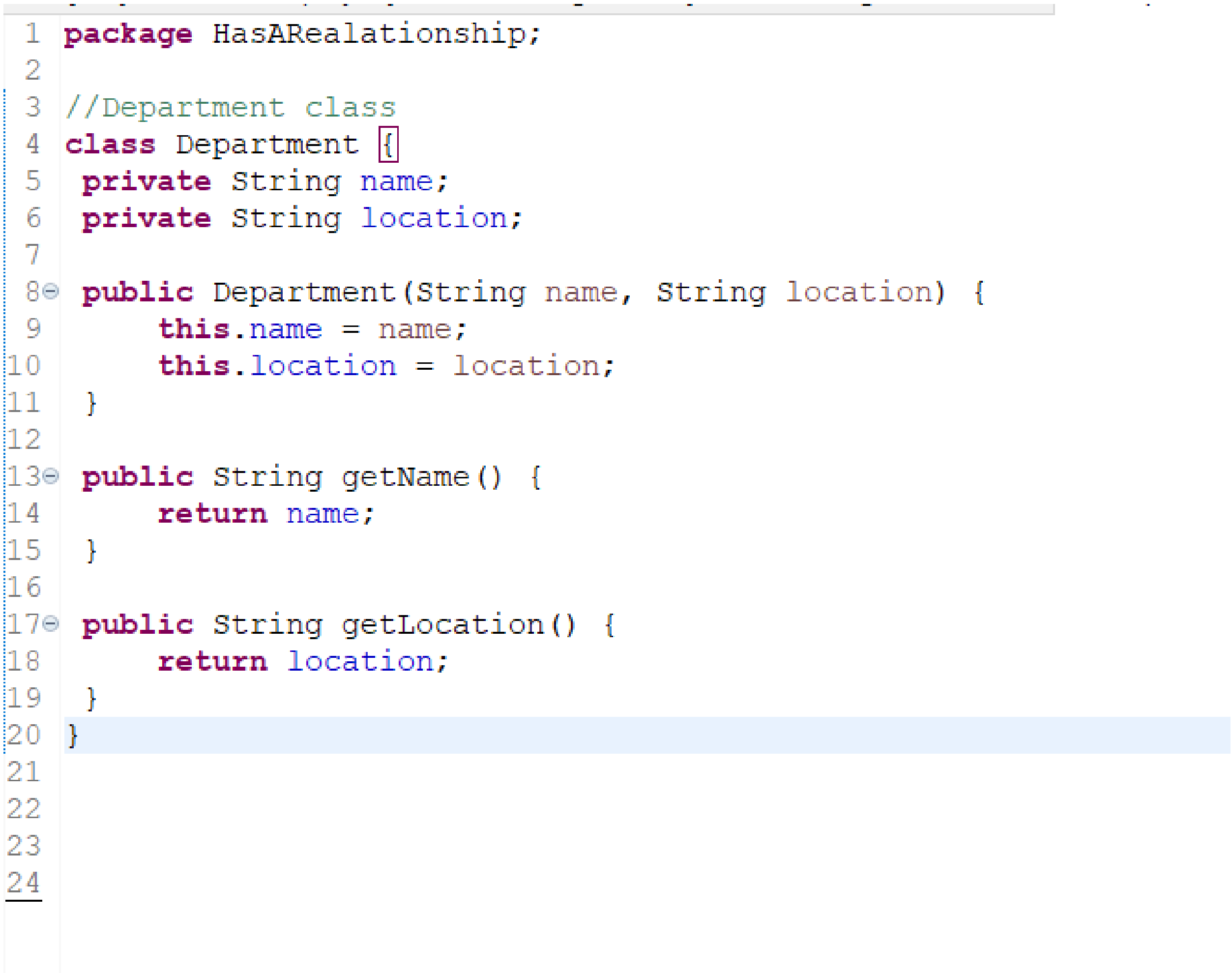
```java
package inheritance;

class Programmer extends Employee {
    private String[] programmingLanguages;

    public Programmer(String name, int employeeId, String department, String[] programmingLanguages) {
        super(name, employeeId, department);
        this.programmingLanguages = programmingLanguages;
    }

    public String[] getProgrammingLanguages() {
        return programmingLanguages;
    }

    // Overriding the work method
    @Override
    public void work() {
        System.out.println("Programmer is coding...");
    }
    }
```

Main class

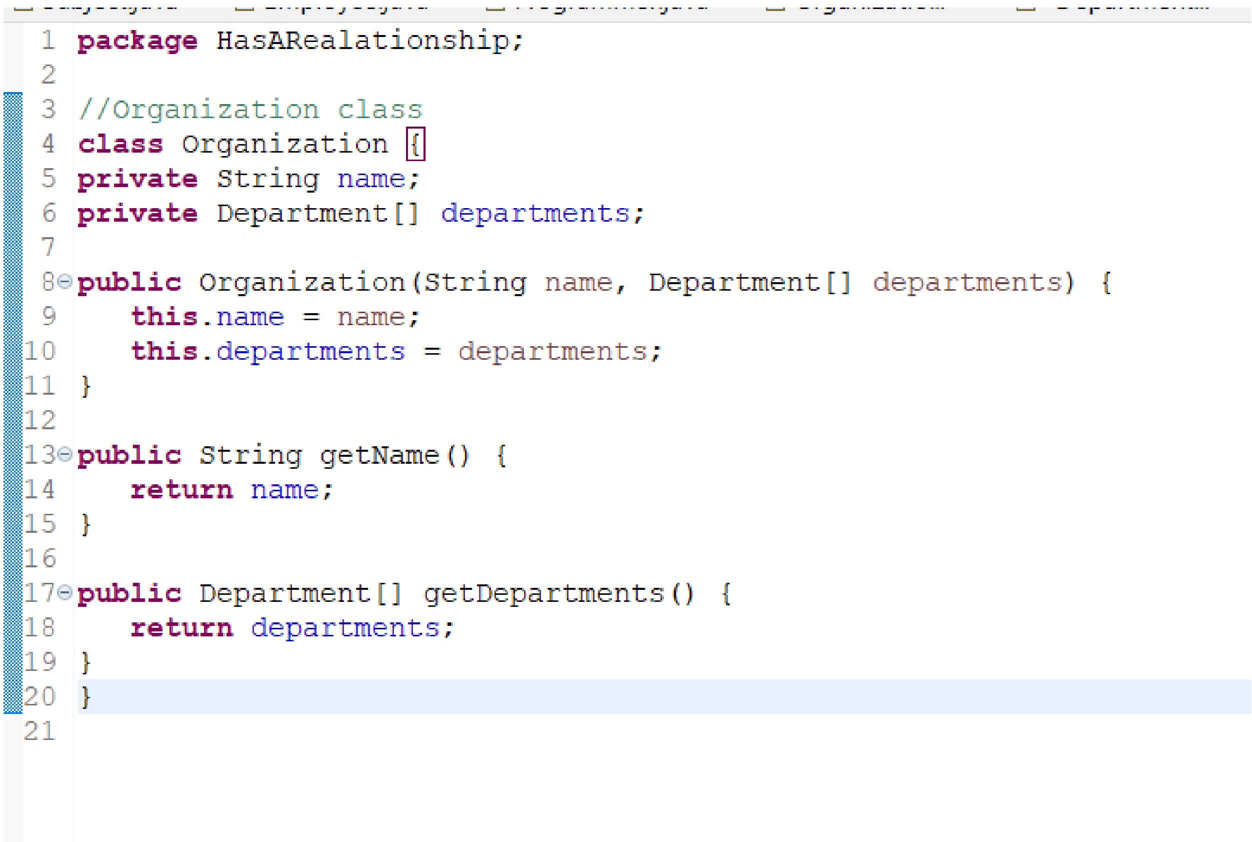


```java
package inheritance;

class Programmer extends Employee {
    private String[] programmingLanguages;

    public Programmer(String name, int employeeId, String department, String[] programmingLanguages) {
        super(name, employeeId, department);
        this.programmingLanguages = programmingLanguages;
    }

    public String[] getProgrammingLanguages() {
        return programmingLanguages;
    }

    // Overriding the work method
    @Override
    public void work() {
        System.out.println("Programmer is coding...");
    }
    }
```

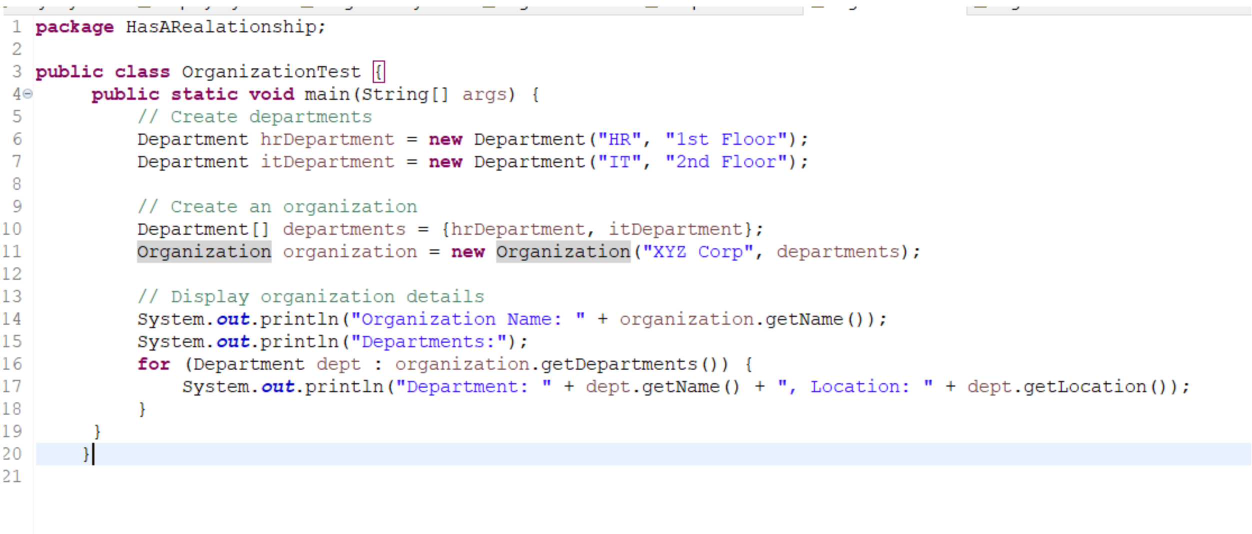
6. W.A.P and explain the "Has-A" relationship.



```java
package HasARealationship;

//Department class
class Department {
 private String name;
 private String location;

 public Department(String name, String location) {
     this.name = name;
     this.location = location;
 }

 public String getName() {
     return name;
 }

 public String getLocation() {
     return location;
 }
}
```



```java
package HasARealationship;

//Organization class
class Organization {
private String name;
private Department[] departments;

public Organization(String name, Department[] departments) {
    this.name = name;
    this.departments = departments;
}

public String getName() {
    return name;
}

public Department[] getDepartments() {
    return departments;
}
}
```

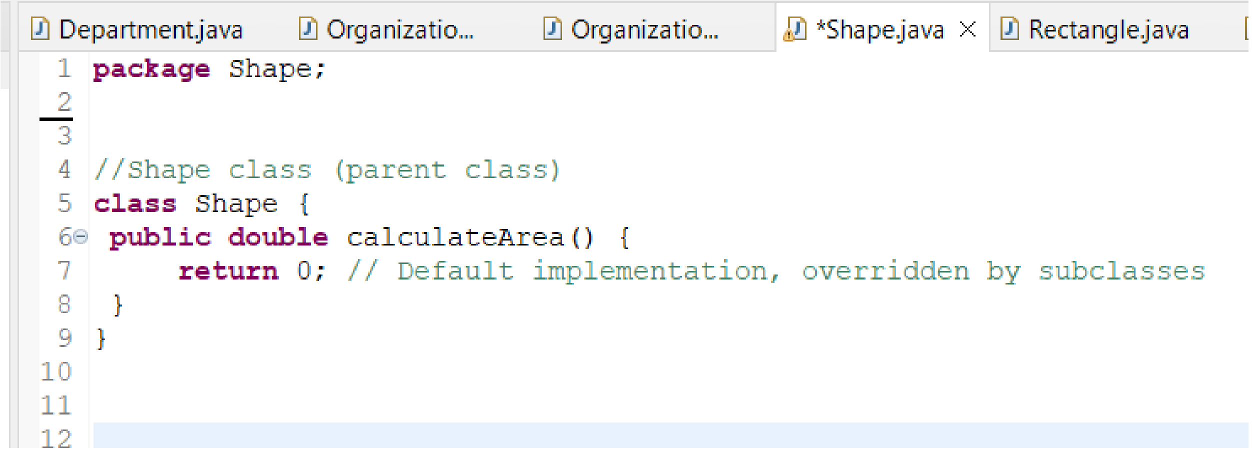


```java
package HasARealationship;

public class OrganizationTest {
    public static void main(String[] args) {
        // Create departments
        Department hrDepartment = new Department("HR", "1st Floor");
        Department itDepartment = new Department("IT", "2nd Floor");

        // Create an organization
        Department[] departments = {hrDepartment, itDepartment};
        Organization organization = new Organization("XYZ Corp", departments);

        // Display organization details
        System.out.println("Organization Name: " + organization.getName());
        System.out.println("Departments:");
        for (Department dept : organization.getDepartments()) {
            System.out.println("Department: " + dept.getName() + ", Location: " + dept.getLocation());
        }
    }
   }
```

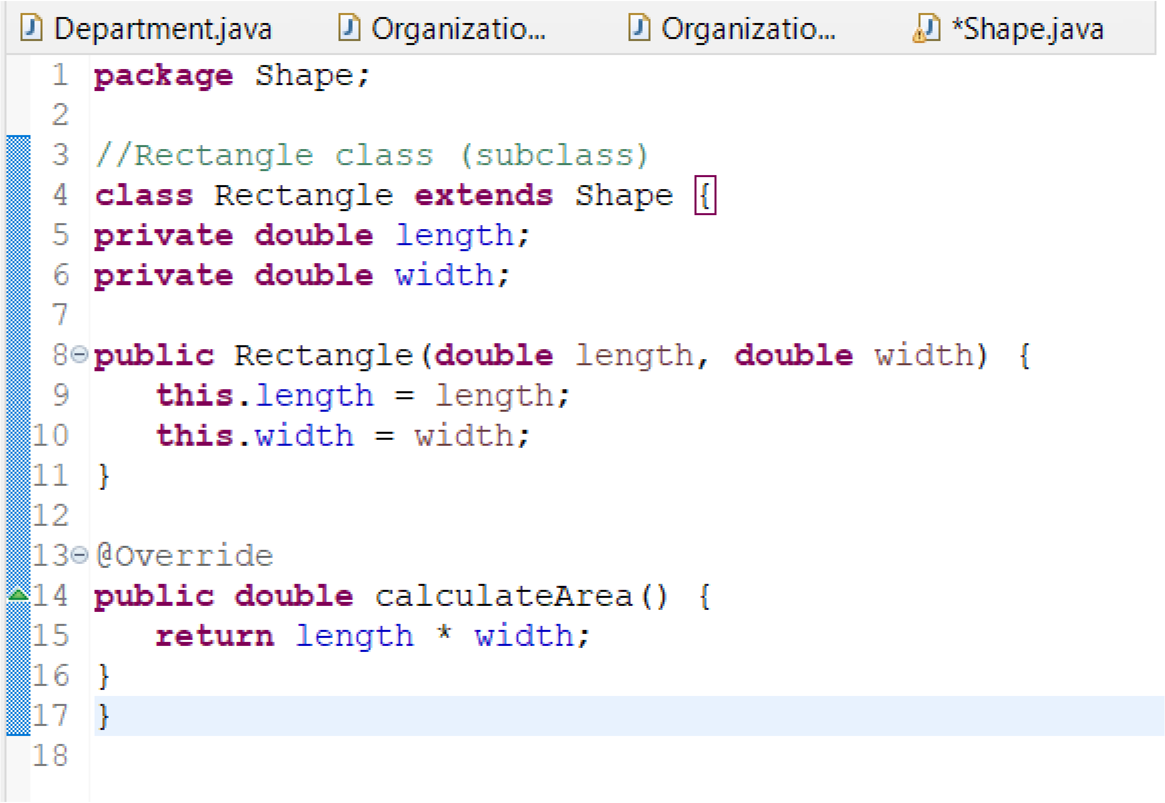
Has-A relationship : In a "has-a" relationship, one class has an instance of another class as one of its members. This is often represented by composition, where one class contains an instance of another class as a field.

In the above relation, class Organization has a instance of Department as its class member.

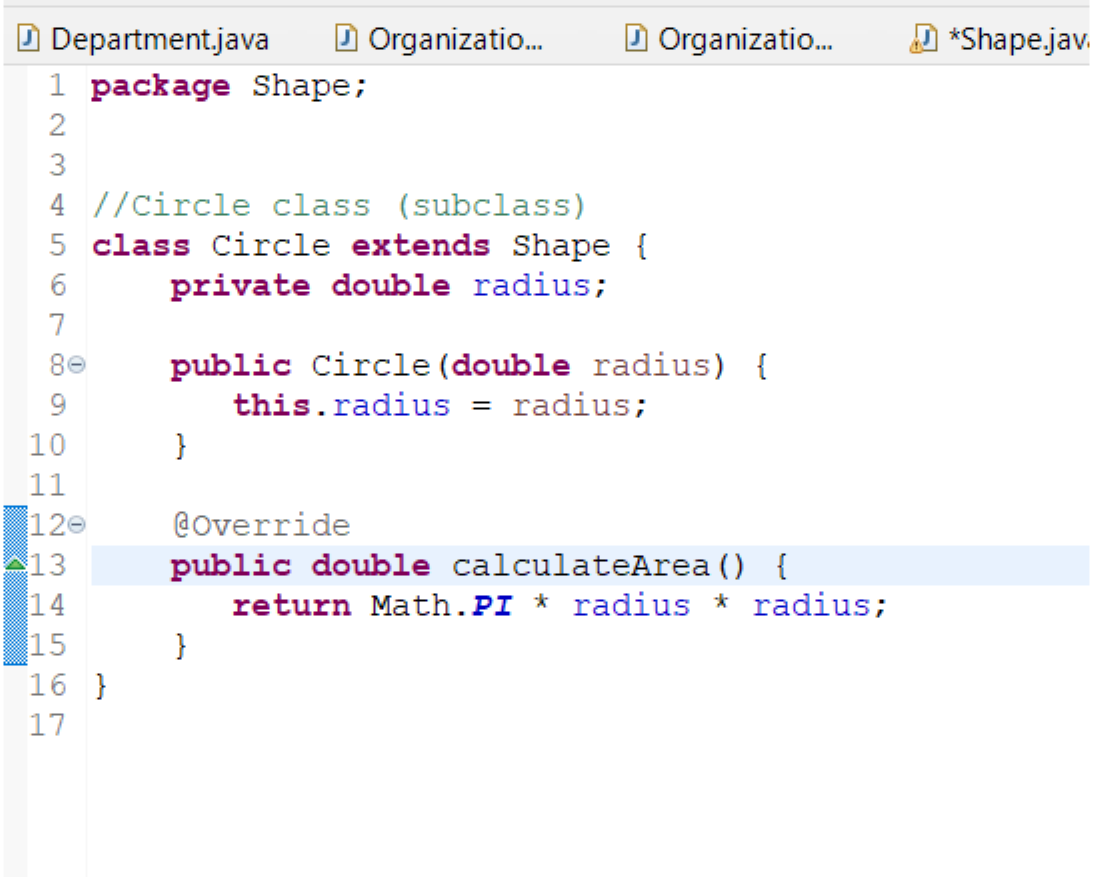
7. W.A.P based on the concept of Inheritance using Shape class of various geometrical figures.Calculate Area for different shapes



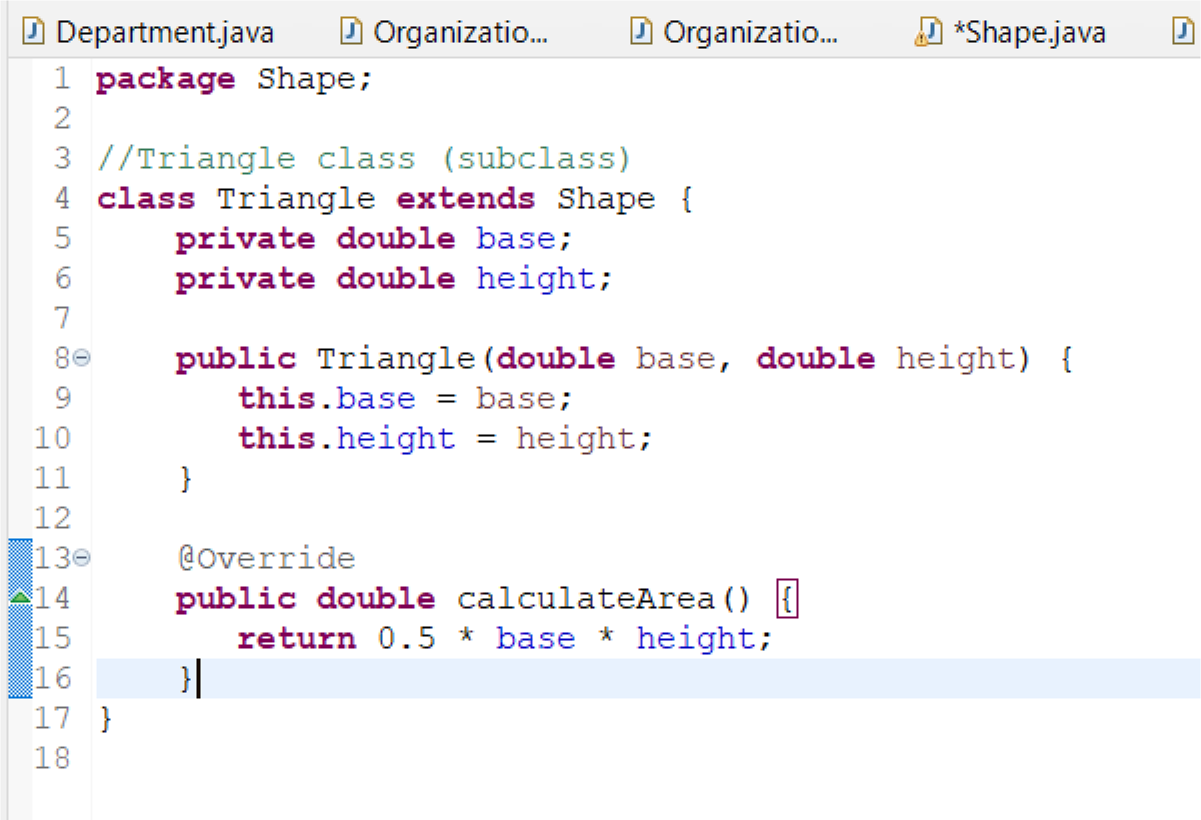
```
package Shape;


//Shape class (parent class)
class Shape {
 public double calculateArea() {
     return 0; // Default implementation, overridden by subclasses
 }
}



```



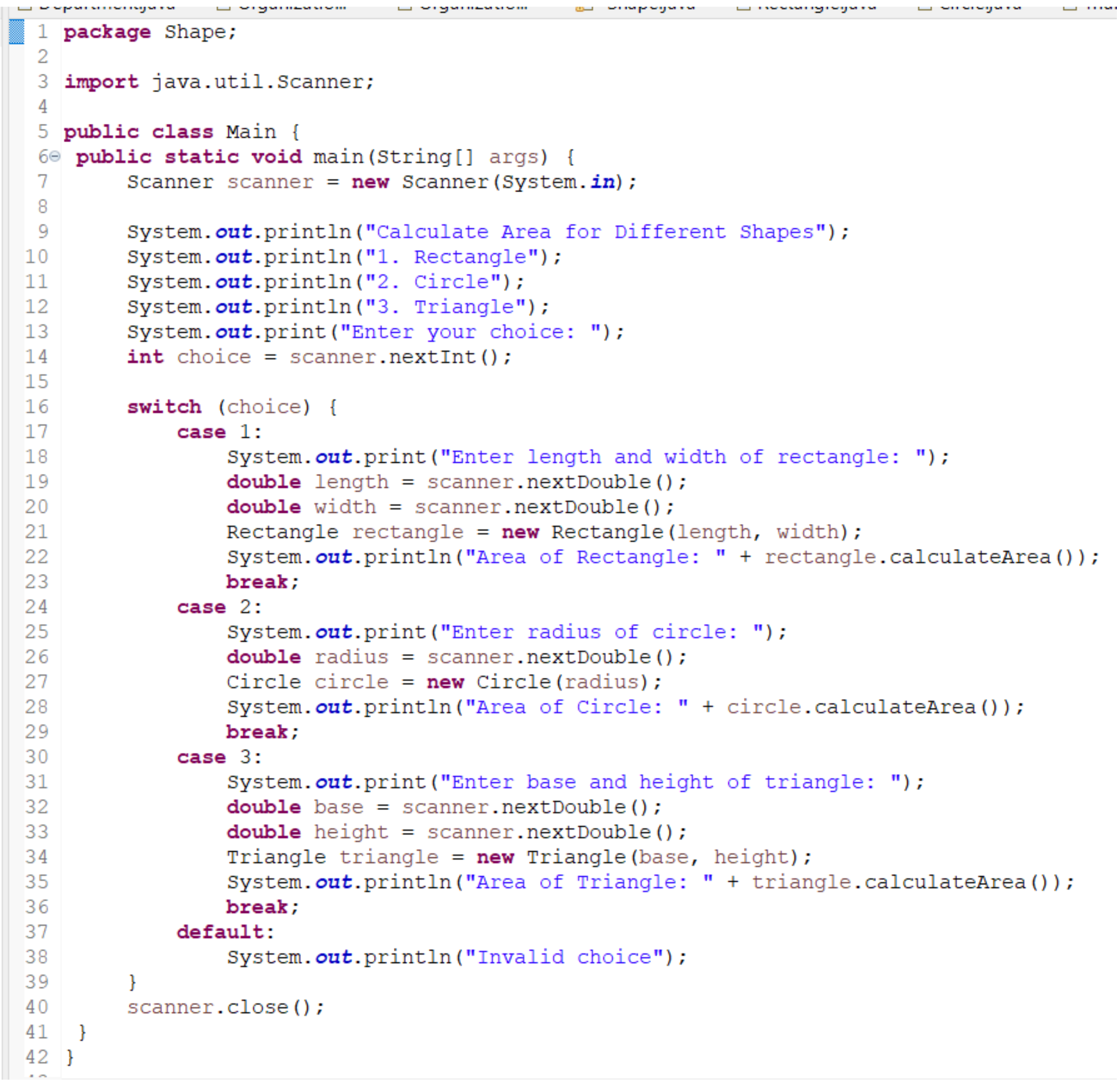
```
package Shape;

//Rectangle class (subclass)
class Rectangle extends Shape {
private double length;
private double width;

public Rectangle(double length, double width) {
    this.length = length;
    this.width = width;
}

@Override
public double calculateArea() {
    return length * width;
}
}

```



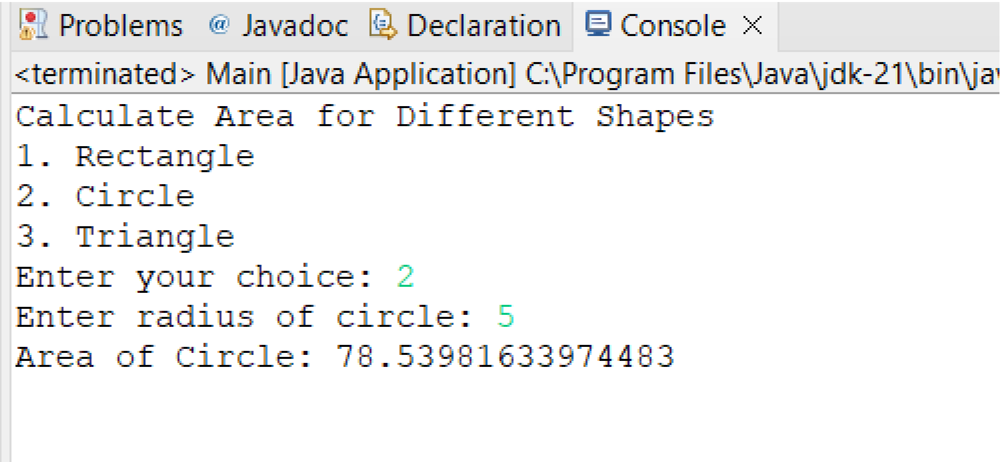
```java
package Shape;


//Circle class (subclass)
class Circle extends Shape {
    private double radius;

    public Circle(double radius) {
        this.radius = radius;
    }

    @Override
    public double calculateArea() {
        return Math.PI * radius * radius;
    }
}

```



```java
package Shape;

//Triangle class (subclass)
class Triangle extends Shape {
    private double base;
    private double height;

    public Triangle(double base, double height) {
        this.base = base;
        this.height = height;
    }

    @Override
    public double calculateArea() {
        return 0.5 * base * height;
    }
}

```



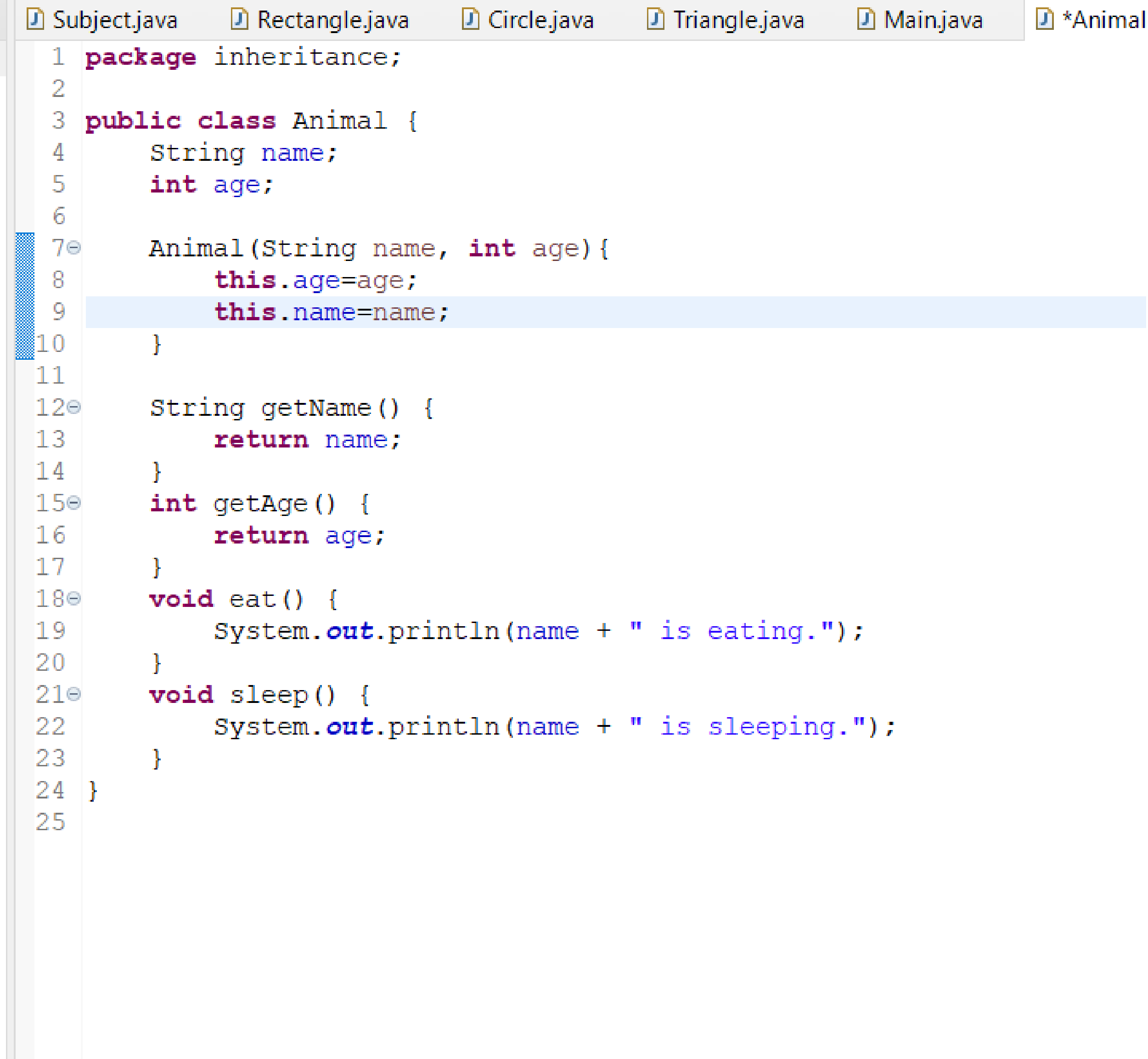
```java
package Shape;

import java.util.Scanner;

public class Main {
 public static void main(String[] args) {
     Scanner scanner = new Scanner(System.in);

     System.out.println("Calculate Area for Different Shapes");
     System.out.println("1. Rectangle");
     System.out.println("2. Circle");
     System.out.println("3. Triangle");
     System.out.print("Enter your choice: ");
     int choice = scanner.nextInt();

     switch (choice) {
         case 1:
             System.out.print("Enter length and width of rectangle: ");
             double length = scanner.nextDouble();
             double width = scanner.nextDouble();
             Rectangle rectangle = new Rectangle(length, width);
             System.out.println("Area of Rectangle: " + rectangle.calculateArea());
             break;
         case 2:
             System.out.print("Enter radius of circle: ");
             double radius = scanner.nextDouble();
             Circle circle = new Circle(radius);
             System.out.println("Area of Circle: " + circle.calculateArea());
             break;
         case 3:
             System.out.print("Enter base and height of triangle: ");
             double base = scanner.nextDouble();
             double height = scanner.nextDouble();
             Triangle triangle = new Triangle(base, height);
             System.out.println("Area of Triangle: " + triangle.calculateArea());
             break;
         default:
             System.out.println("Invalid choice");
     }
     scanner.close();
 }
}
```



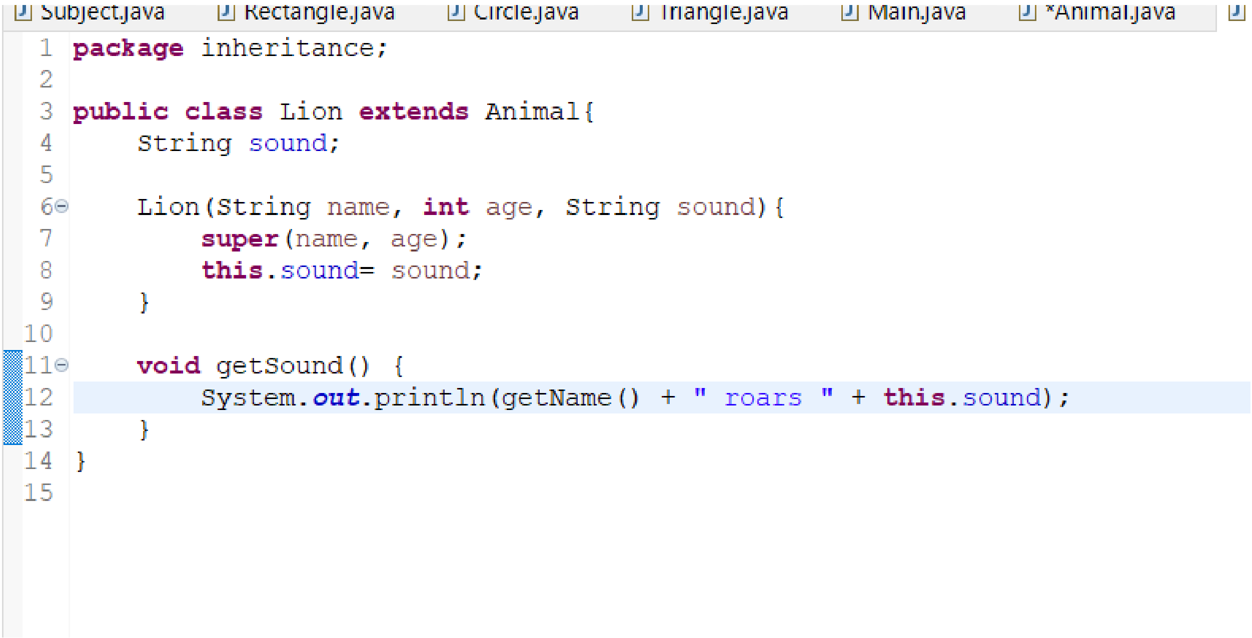
<terminated> Main [Java Application] C:\Program Files\Java\jdk-21\bin\ja

```
Calculate Area for Different Shapes
1. Rectangle
2. Circle
3. Triangle
Enter your choice: 2
Enter radius of circle: 5
Area of Circle: 78.53981633974483
```

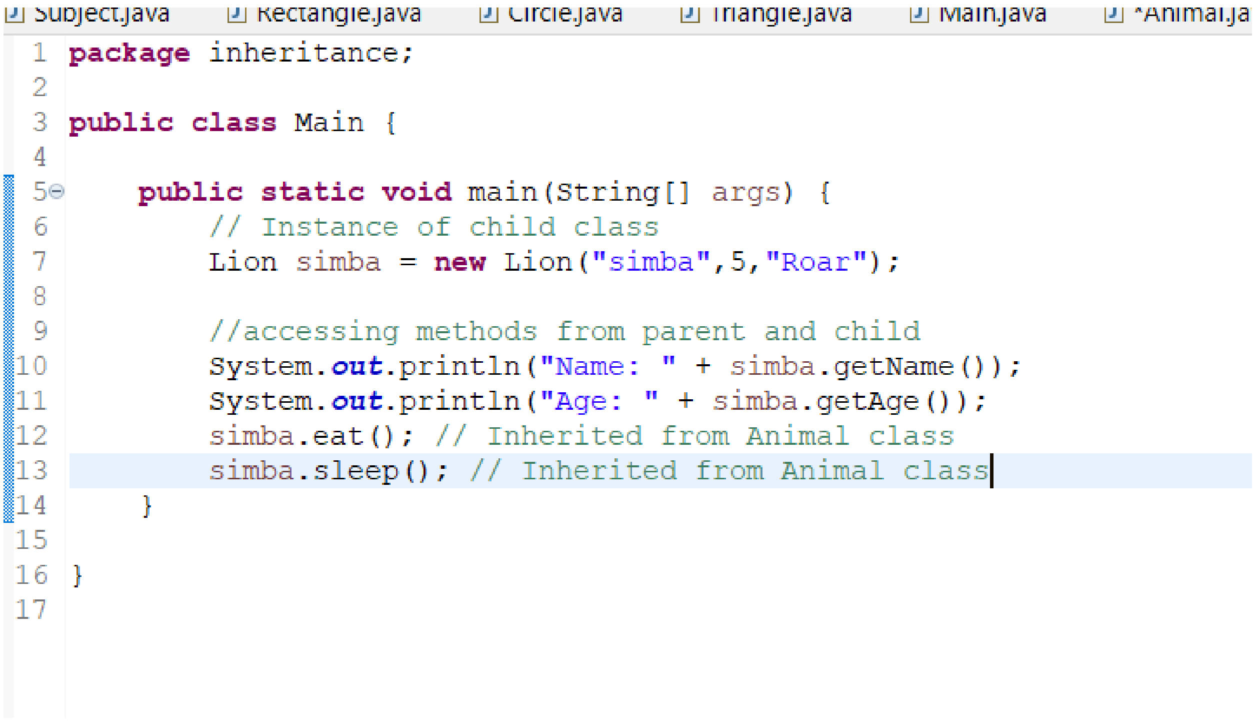
8. **W.A.P to define a Class "Animal" and a subclass "Lion"**



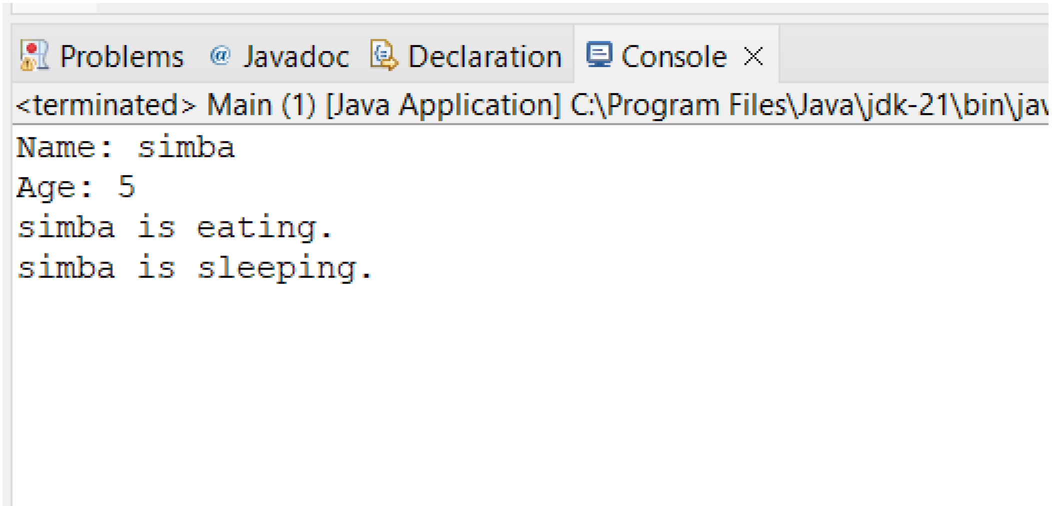
```java
package inheritance;

public class Animal {
    String name;
    int age;

    Animal(String name, int age){
        this.age=age;
        this.name=name;
    }

    String getName() {
        return name;
    }
    int getAge() {
        return age;
    }
    void eat() {
        System.out.println(name + " is eating.");
    }
    void sleep() {
        System.out.println(name + " is sleeping.");
    }
}

```



```java
package inheritance;

public class Lion extends Animal{
    String sound;

    Lion(String name, int age, String sound){
        super(name, age);
        this.sound= sound;
    }

    void getSound() {
        System.out.println(getName() + " roars " + this.sound);
    }
}

```



```java
package inheritance;

public class Main {

    public static void main(String[] args) {
        // Instance of child class
        Lion simba = new Lion("simba",5,"Roar");

        //accessing methods from parent and child
        System.out.println("Name: " + simba.getName());
        System.out.println("Age: " + simba.getAge());
        simba.eat(); // Inherited from Animal class
        simba.sleep(); // Inherited from Animal class
    }

}
```



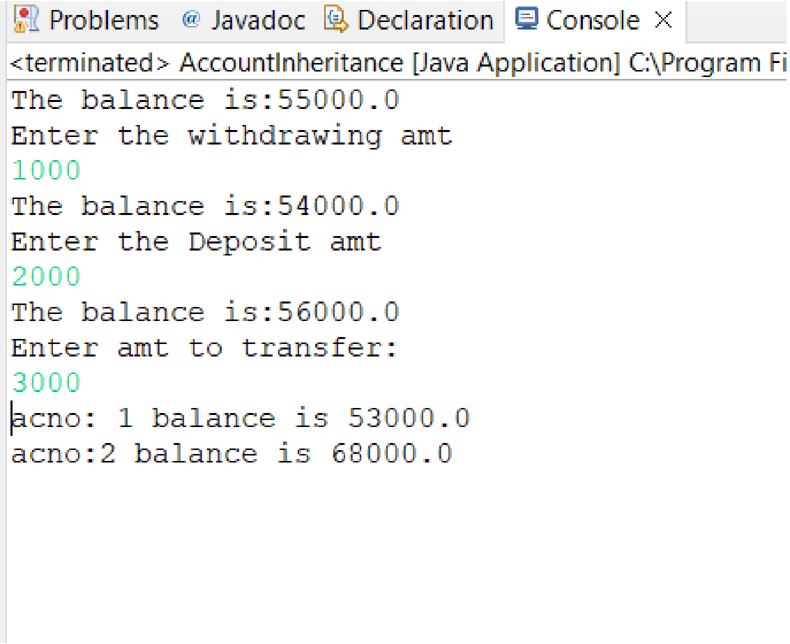
```
<terminated> Main (1) [Java Application] C:\Program Files\Java\jdk-21\bin\jav
Name: simba
Age: 5
simba is eating.
simba is sleeping.
```

9. **Find out the Error in the following program- Inheritance using Bank Class ,rectify if any.**

Code:

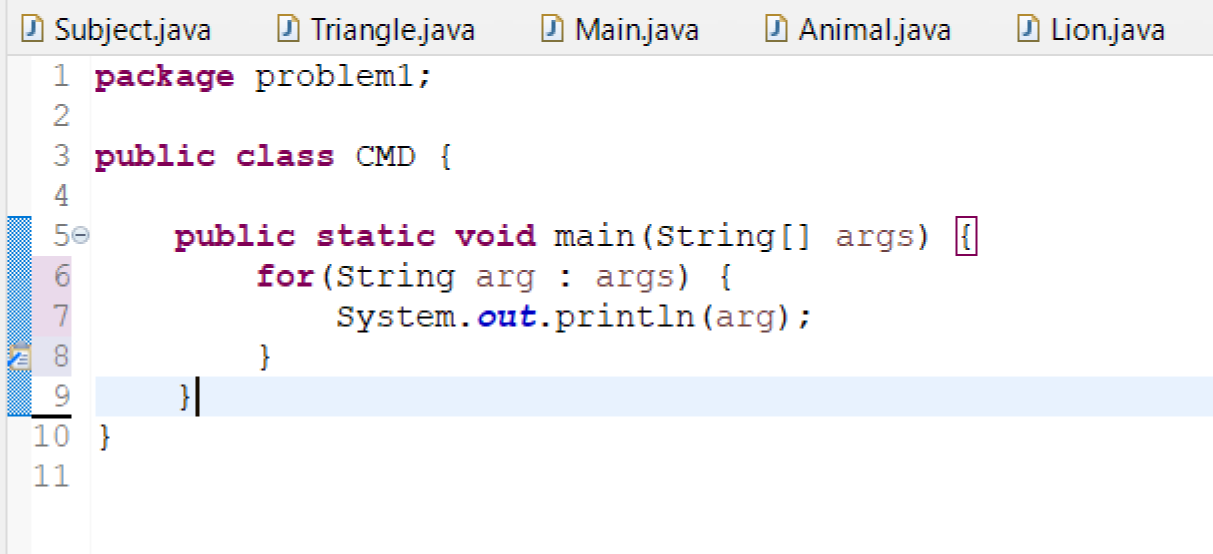
```java
package Refractor;
```

```
import java.util.Scanner;
class Account{
    String Name;
    int acno;
    double balance;

    //rectification
    Account(String name,int acno,double bal){
        this.Name=name;
        this.acno=acno;
        this.balance=bal;
    }

    double checkbal(){
        return balance;
    }
}
class AccountEx extends Account{
    AccountEx (String Name,int a,double b){
        super(Name,a,b);
    }

    void withdraw(double amt){
        balance=balance-amt;
        if(balance < 500){
            System.out.println("Can't Withdraw minimum balance should be greater than 500");
        }
    }
    void deposit(double amt){
        balance=balance+amt;
    }
    void transfer(Account b, double k){
        balance=balance-k;
        if(balance<500.00){
            System.out.println("You don't have sufficient balance transfer");
        }else{
            b.balance=b.balance+k;
            System.out.println("acno: "+acno+" "+"balance is"+" "+balance);
            System.out.println("acno:"+b.acno+" balance is "+b.balance);
        }
    }
}
class AccountInheritance{
    public static void main(String args[]){
    AccountEx s = new AccountEx("Pradnya",1,55000.00);
    System.out.println("The balance is:"+s.checkbal());
    System.out.println("Enter the withdrawing amt");
    Scanner sc = new Scanner(System.in);
    double i = sc.nextDouble();
    s.withdraw(i);
    System.out.println("The balance is:"+s.checkbal());
    System.out.println("Enter the Deposit amt");
    Scanner sc1 = new Scanner(System.in);
    double j = sc1.nextDouble();
    s.deposit(j);
    System.out.println("The balance is:"+s.checkbal());
    AccountEx b = new AccountEx("Nameeta",2,65000.00);
    System.out.println("Enter amt to transfer:");
    Scanner sc2 = new Scanner(System.in);
    double k = sc2.nextDouble();
    s.transfer(b,k);
    sc.close();
    }
}
```

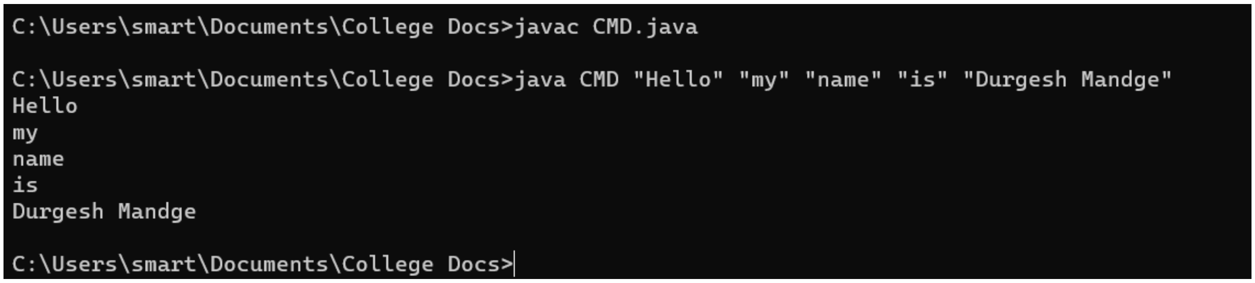


```
<terminated> AccountInheritance [Java Application] C:\Program Fi
The balance is:55000.0
Enter the withdrawing amt
1000
The balance is:54000.0
Enter the Deposit amt
2000
The balance is:56000.0
Enter amt to transfer:
3000
acno: 1 balance is 53000.0
acno:2 balance is 68000.0
```

10. W.A.P to print command line arguments using for loop

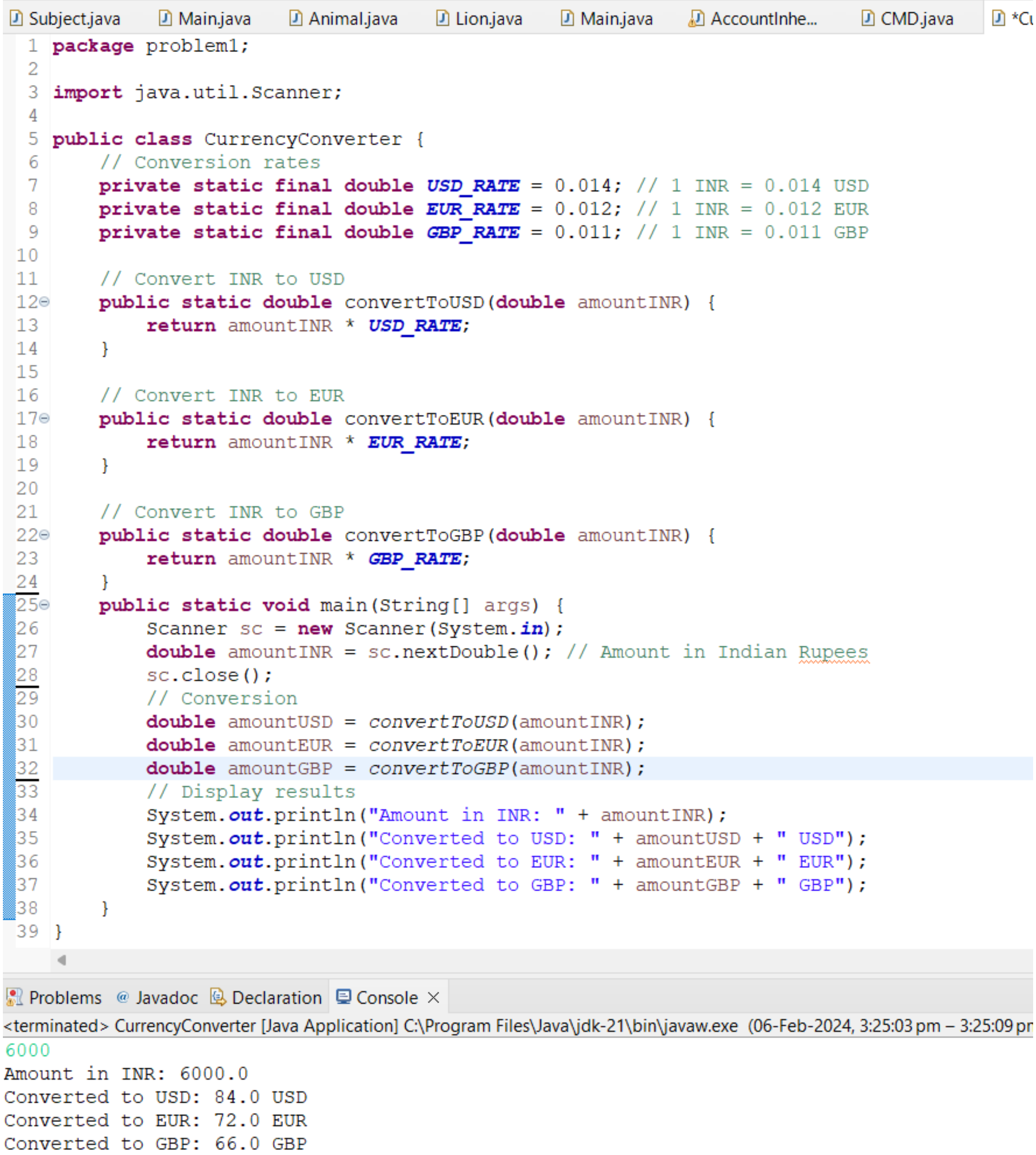


```java
package problem1;

public class CMD {

    public static void main(String[] args) {
        for(String arg : args) {
            System.out.println(arg);
        }
    }
}

```



```
C:\Users\smart\Documents\College Docs>javac CMD.java

C:\Users\smart\Documents\College Docs>java CMD "Hello" "my" "name" "is" "Durgesh Mandge"
Hello
my
name
is
Durgesh Mandge

C:\Users\smart\Documents\College Docs>
```

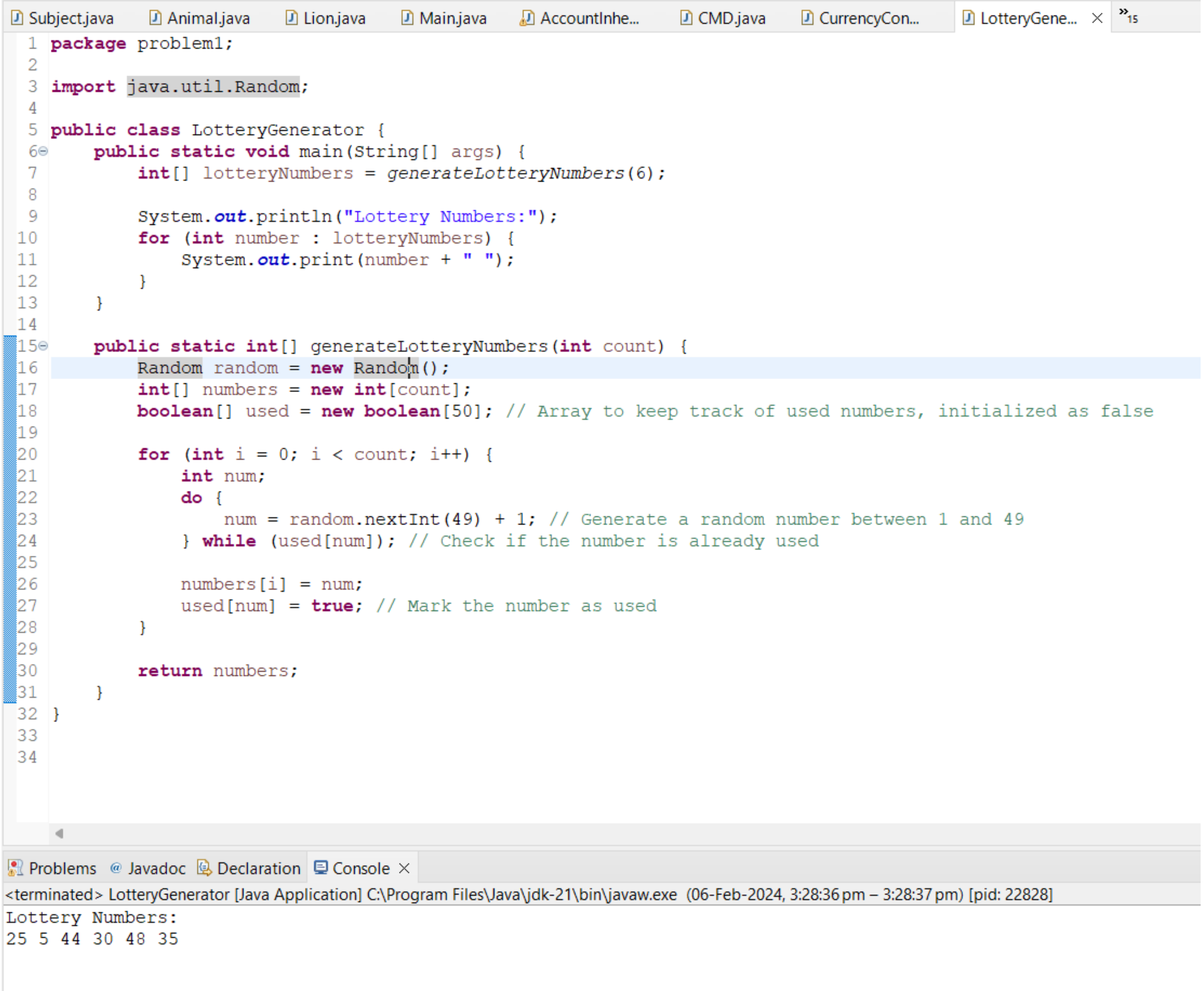
11. W.A.P to create a Class “Currency Convertor” to convertor” to convert Rupees into Different Currencies.



```java
package problem1;

import java.util.Scanner;

public class CurrencyConverter {
    // Conversion rates
    private static final double USD_RATE = 0.014; // 1 INR = 0.014 USD
    private static final double EUR_RATE = 0.012; // 1 INR = 0.012 EUR
    private static final double GBP_RATE = 0.011; // 1 INR = 0.011 GBP

    // Convert INR to USD
    public static double convertToUSD(double amountINR) {
        return amountINR * USD_RATE;
    }

    // Convert INR to EUR
    public static double convertToEUR(double amountINR) {
        return amountINR * EUR_RATE;
    }

    // Convert INR to GBP
    public static double convertToGBP(double amountINR) {
        return amountINR * GBP_RATE;
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        double amountINR = sc.nextDouble(); // Amount in Indian Rupees
        sc.close();
        // Conversion
        double amountUSD = convertToUSD(amountINR);
        double amountEUR = convertToEUR(amountINR);
        double amountGBP = convertToGBP(amountINR);
        // Display results
        System.out.println("Amount in INR: " + amountINR);
        System.out.println("Converted to USD: " + amountUSD + " USD");
        System.out.println("Converted to EUR: " + amountEUR + " EUR");
        System.out.println("Converted to GBP: " + amountGBP + " GBP");
    }
}
```

```
<terminated> CurrencyConverter [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe  (06-Feb-2024, 3:25:03 pm – 3:25:09 pr
6000
Amount in INR: 6000.0
Converted to USD: 84.0 USD
Converted to EUR: 72.0 EUR
Converted to GBP: 66.0 GBP
```

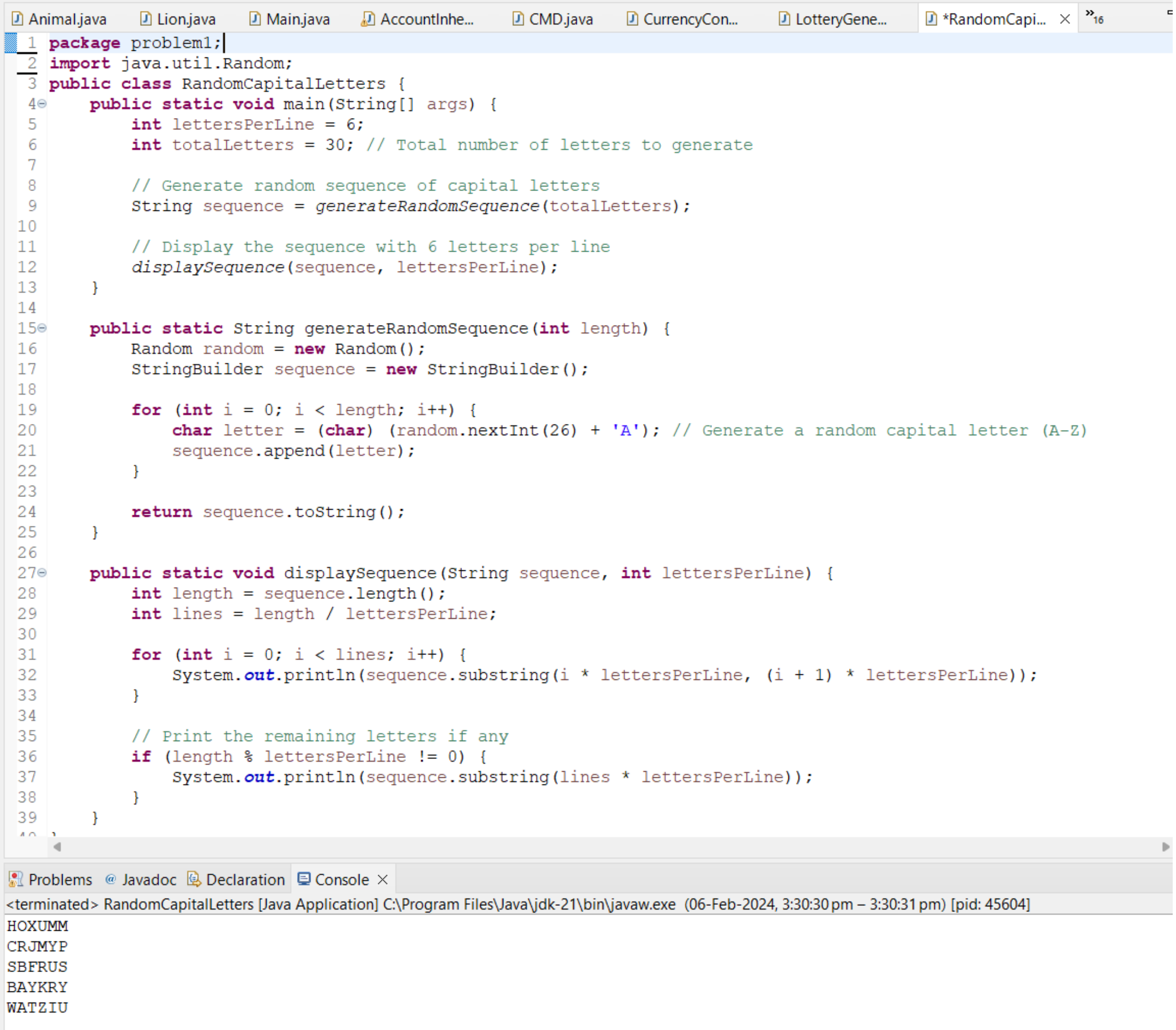
12. W.A.P to generate Lottery Number from (1 to 49)



```java
package problem1;

import java.util.Random;

public class LotteryGenerator {
    public static void main(String[] args) {
        int[] lotteryNumbers = generateLotteryNumbers(6);

        System.out.println("Lottery Numbers:");
        for (int number : lotteryNumbers) {
            System.out.print(number + " ");
        }
    }

    public static int[] generateLotteryNumbers(int count) {
        Random random = new Random();
        int[] numbers = new int[count];
        boolean[] used = new boolean[50]; // Array to keep track of used numbers, initialized as false

        for (int i = 0; i < count; i++) {
            int num;
            do {
                num = random.nextInt(49) + 1; // Generate a random number between 1 and 49
            } while (used[num]); // Check if the number is already used

            numbers[i] = num;
            used[num] = true; // Mark the number as used
        }

        return numbers;
    }
}
```

```
<terminated> LotteryGenerator [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe  (06-Feb-2024, 3:28:36 pm – 3:28:37 pm) [pid: 22828]
Lottery Numbers:
25 5 44 30 48 35
```

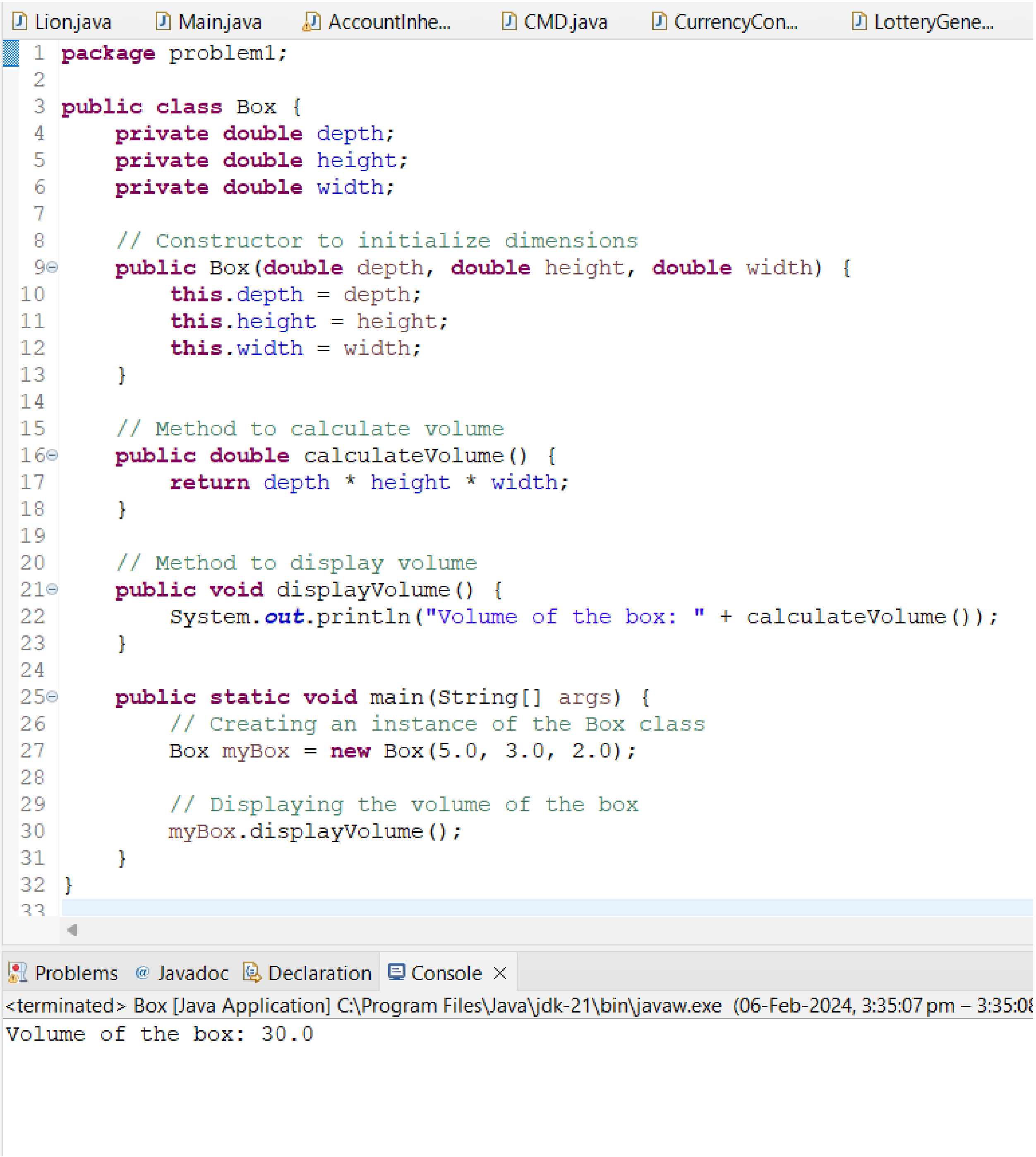
13. W.A.P to generate a random sequence of capital letters such that in one line only 6 letters can be seen.



```java
package problem1;
import java.util.Random;
public class RandomCapitalLetters {
    public static void main(String[] args) {
        int lettersPerLine = 6;
        int totalLetters = 30; // Total number of letters to generate

        // Generate random sequence of capital letters
        String sequence = generateRandomSequence(totalLetters);

        // Display the sequence with 6 letters per line
        displaySequence(sequence, lettersPerLine);
    }

    public static String generateRandomSequence(int length) {
        Random random = new Random();
        StringBuilder sequence = new StringBuilder();

        for (int i = 0; i < length; i++) {
            char letter = (char) (random.nextInt(26) + 'A'); // Generate a random capital letter (A-Z)
            sequence.append(letter);
        }

        return sequence.toString();
    }

    public static void displaySequence(String sequence, int lettersPerLine) {
        int length = sequence.length();
        int lines = length / lettersPerLine;

        for (int i = 0; i < lines; i++) {
            System.out.println(sequence.substring(i * lettersPerLine, (i + 1) * lettersPerLine));
        }

        // Print the remaining letters if any
        if (length % lettersPerLine != 0) {
            System.out.println(sequence.substring(lines * lettersPerLine));
        }
    }
```

```
<terminated> RandomCapitalLetters [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe  (06-Feb-2024, 3:30:30 pm – 3:30:31 pm) [pid: 45604]
HOXUMM
CRJMYP
SBFRUS
BAYKRY
WATZIU
```

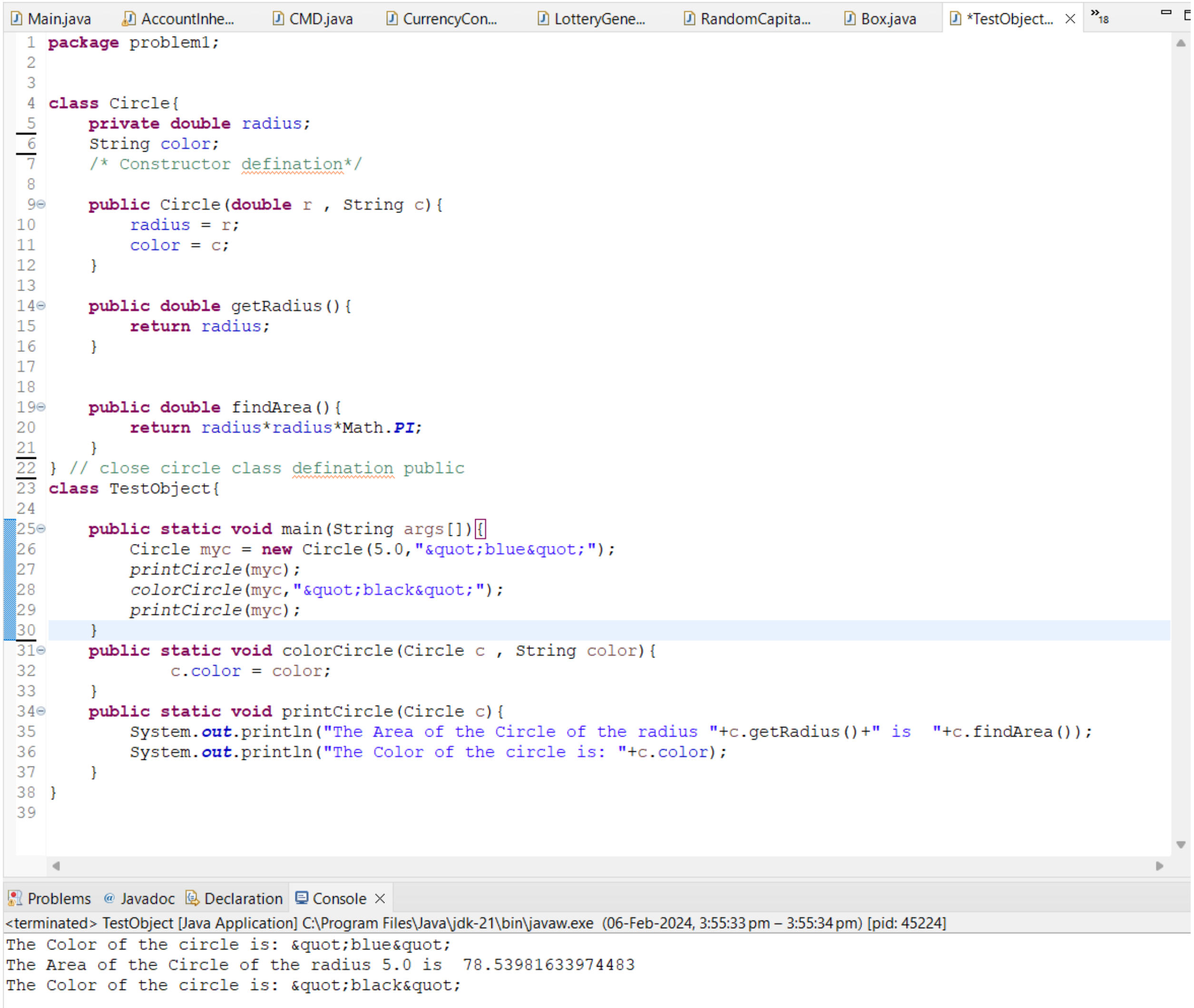
14.W.A.P using Constructor and Destructor for a Class " Box " using depth, height and width as parameters for Volume.



```java
package problem1;

public class Box {
    private double depth;
    private double height;
    private double width;

    // Constructor to initialize dimensions
    public Box(double depth, double height, double width) {
        this.depth = depth;
        this.height = height;
        this.width = width;
    }

    // Method to calculate volume
    public double calculateVolume() {
        return depth * height * width;
    }

    // Method to display volume
    public void displayVolume() {
        System.out.println("Volume of the box: " + calculateVolume());
    }

    public static void main(String[] args) {
        // Creating an instance of the Box class
        Box myBox = new Box(5.0, 3.0, 2.0);

        // Displaying the volume of the box
        myBox.displayVolume();
    }
}
```

```
<terminated> Box [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe  (06-Feb-2024, 3:35:07 pm – 3:35:0
Volume of the box: 30.0
```

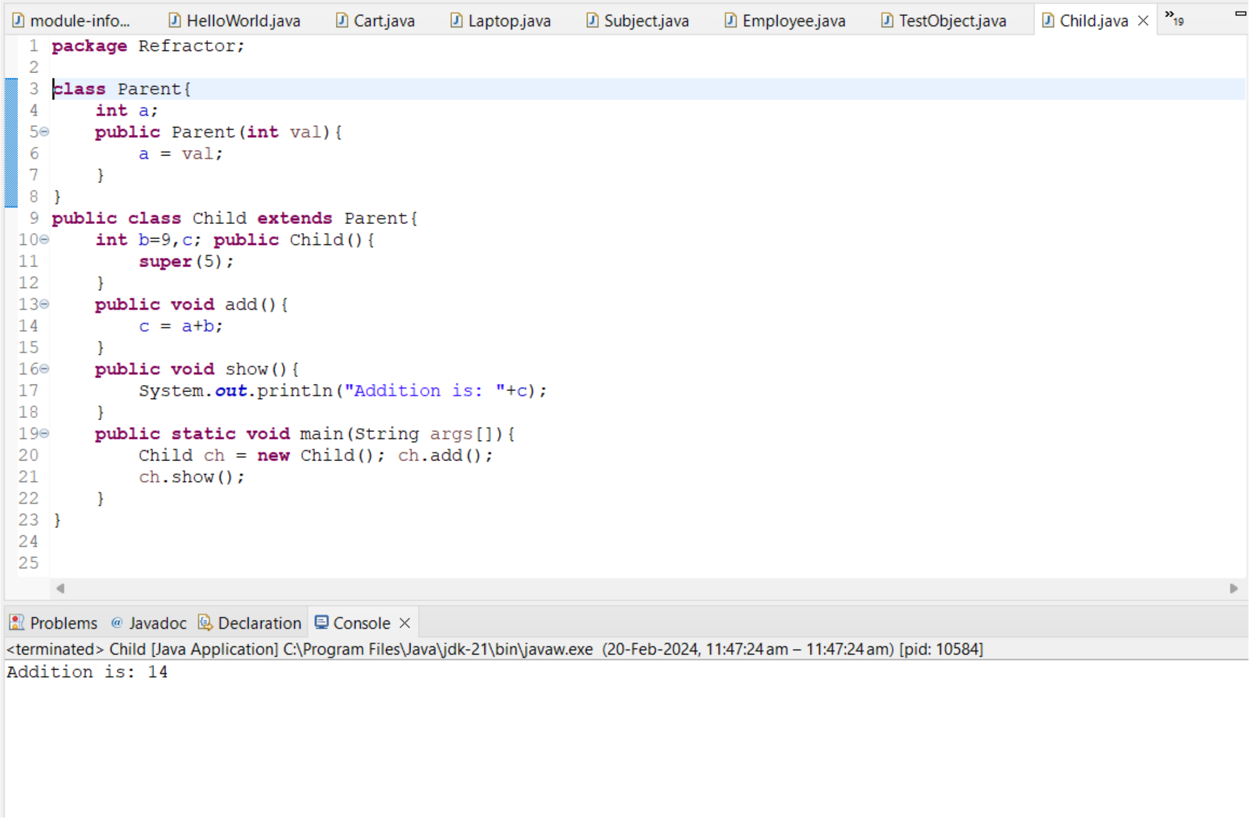
15.Identify the error in the give program class Circle



```java
package problem1;


class Circle{
    private double radius;
    String color;
    /* Constructor defination*/

    public Circle(double r , String c){
        radius = r;
        color = c;
    }

    public double getRadius(){
        return radius;
    }


    public double findArea(){
        return radius*radius*Math.PI;
    }
} // close circle class defination public
class TestObject{

    public static void main(String args[]){
        Circle myc = new Circle(5.0,"&quot;blue&quot;");
        printCircle(myc);
        colorCircle(myc,"&quot;black&quot;");
        printCircle(myc);
    }
    public static void colorCircle(Circle c , String color){
            c.color = color;
    }
    public static void printCircle(Circle c){
        System.out.println("The Area of the Circle of the radius "+c.getRadius()+" is  "+c.findArea());
        System.out.println("The Color of the circle is: "+c.color);
    }
}

```

```
<terminated> TestObject [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe  (06-Feb-2024, 3:55:33 pm – 3:55:34 pm) [pid: 45224]
The Color of the circle is: &quot;blue&quot;
The Area of the Circle of the radius 5.0 is  78.53981633974483
The Color of the circle is: &quot;black&quot;
```

16.Execute the code and Demonstrate the significance of how Super Keyword is used in the Program.



```java
package Refractor;

class Parent{
    int a;
    public Parent(int val){
        a = val;
    }
}
public class Child extends Parent{
    int b=9,c; public Child(){
        super(5);
    }
    public void add(){
        c = a+b;
    }
    public void show(){
        System.out.println("Addition is: "+c);
    }
    public static void main(String args[]){
        Child ch = new Child(); ch.add();
        ch.show();
    }
}
```

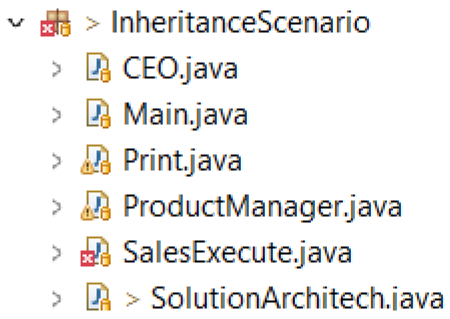
```
<terminated> Child [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe  (20-Feb-2024, 11:47:24 am – 11:47:24 am) [pid: 10584]
Addition is: 14
```

The Super keyword invoked the parent class constructor and passed the val=5, which is set to int a in the default constructor. So when we invoke the add() method, value of c becomes the 9+5=14.

Hence it is printed in show() method.

22. Derive a scenario which has a presence of all levels of Inheritance to demonstrate your understanding about the concept. (note - each student will work on exclusive case study)

Scenario:-



- InheritanceScenario
  - CEO.java
  - Main.java
  - Print.java
  - ProductManager.java
  - SalesExecute.java
  - SolutionArchitech.java

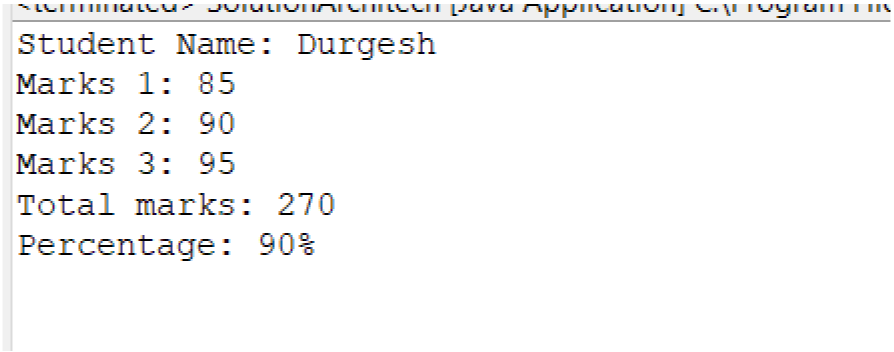
In the above example CEO is the main class which has 3 child classes :- ProductManager, SalesExecute, SolutionArchitech

In the ProductManager class, we can

23. Write a Package MCA which has one class Student. Accept student details through parameterized constructor. Write display () method to display details. Create a main class which will use package and calculate total marks and percentage.

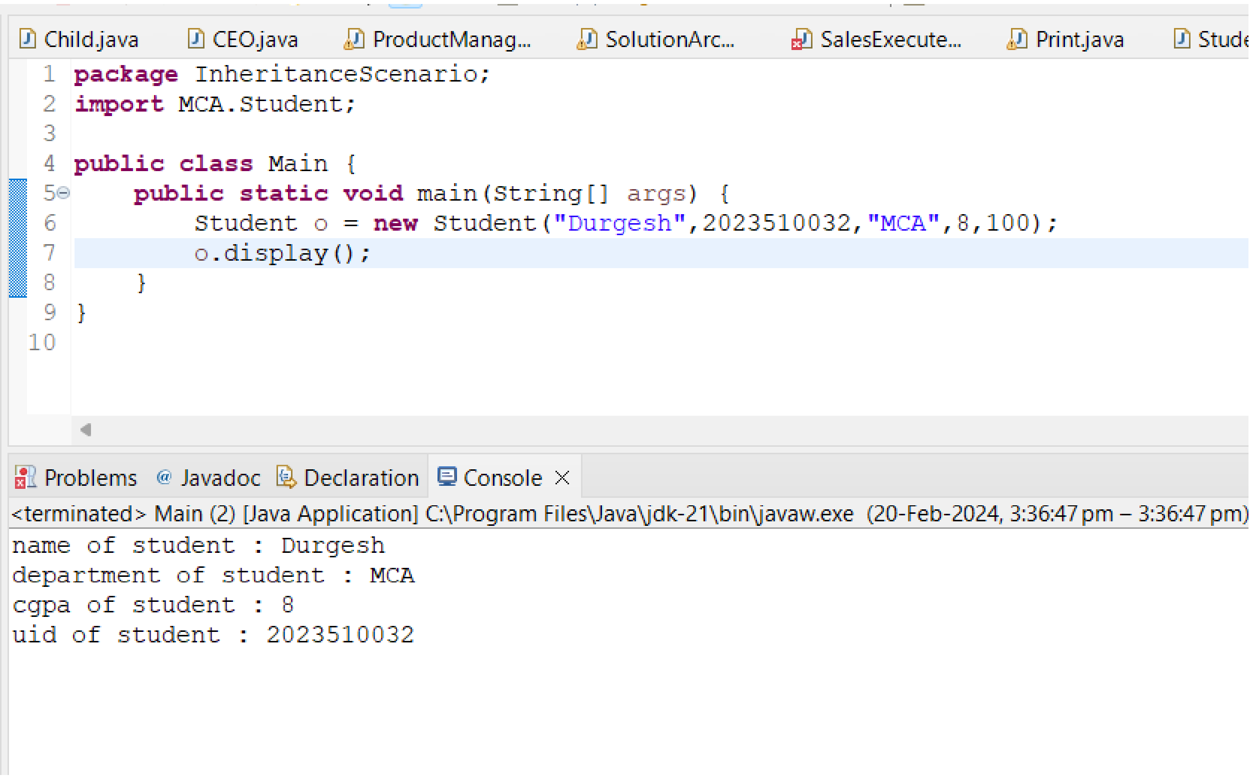
```
        package MCA;

 public class Student {
private String name;
private int marks1; private
int marks2; private int
marks3;

        public Student(String name, int marks1, int marks2, int marks3) {
this.name = name;            this.marks1 = marks1;            this.marks2 = marks2;
        this.marks3 = marks3;
        }
        public void display() {
        System.out.println("Student Name: " + name);
        System.out.println("Marks 1: " + marks1);
        System.out.println("Marks 2: " + marks2);
        System.out.println("Marks 3: " + marks3);
        }
        public int getTotalMarks() {        return
marks1 + marks2 + marks3;
        }     public double getPercentage() {
return (getTotalMarks() / 3.0);
        } } Main
package lab1c;
import
MCA.Student;


public class Main4 {
        public static void main(String[] args) {
        Student student = new Student("Dharmesh", 85, 90, 95);          student.display();
        int totalMarks = student.getTotalMarks();          double
percentage = student.getPercentage();
```

```
        System.out.println("Total Marks: " + totalMarks);
        System.out.println("Percentage: " + percentage + "%");
        } }
```



```
<terminated> SolutionArchitech [Java Application] C:\Program Fil
Student Name: Durgesh
Marks 1: 85
Marks 2: 90
Marks 3: 95
Total marks: 270
Percentage: 90%
```

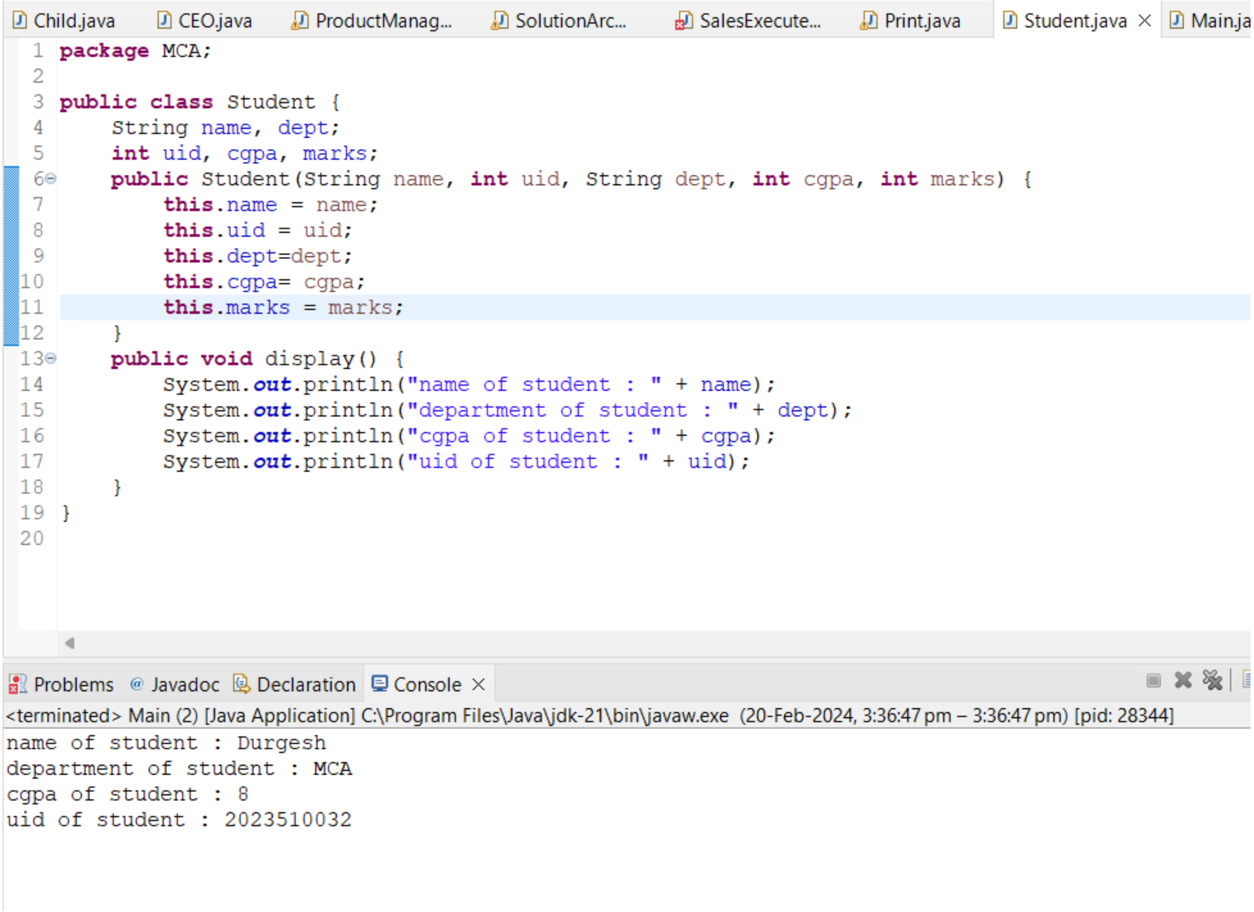
22. Write a program to create a user defined package in Java.



```java
package InheritanceScenario;
import MCA.Student;

public class Main {
    public static void main(String[] args) {
        Student o = new Student("Durgesh",2023510032,"MCA",8,100);
        o.display();
    }
}
```

<terminated> Main (2) [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (20-Feb-2024, 3:36:47 pm – 3:36:47 pm)

```
name of student : Durgesh
department of student : MCA
cgpa of student : 8
uid of student : 2023510032
```



```java
package MCA;

public class Student {
    String name, dept;
    int uid, cgpa, marks;
    public Student(String name, int uid, String dept, int cgpa, int marks) {
        this.name = name;
        this.uid = uid;
        this.dept=dept;
        this.cgpa= cgpa;
        this.marks = marks;
    }
    public void display() {
        System.out.println("name of student : " + name);
        System.out.println("department of student : " + dept);
        System.out.println("cgpa of student : " + cgpa);
        System.out.println("uid of student : " + uid);
    }
}
```

<terminated> Main (2) [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (20-Feb-2024, 3:36:47 pm – 3:36:47 pm) [pid: 28344]

```
name of student : Durgesh
department of student : MCA
cgpa of student : 8
uid of student : 2023510032
```

23. **Write a class Dept with a final keyword as class. Assign the class value as “SYMCA”. Restrict object of Dept from overwriting the value of the class to any other value as “TYMCA” or “FYMCA**

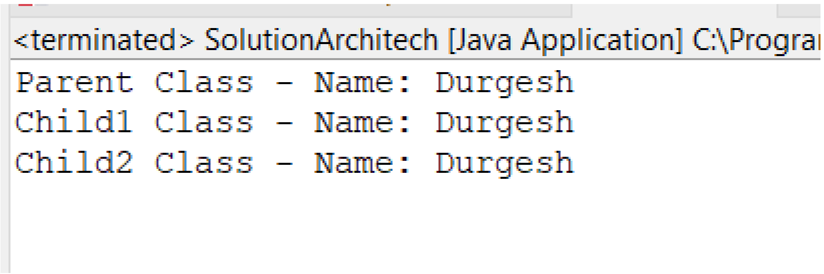
```
package MCA;
     public class Student {
    private String name;
    private int marks1; private
    int marks2; private int
    marks3;

        public Student(String name, int marks1, int marks2, int marks3) {
    this.name = name;              this.marks1 = marks1;              this.marks2 = marks2;
        this.marks3 = marks3;
        }
        public void display() {
        System.out.println("Student Name: " + name);
        System.out.println("Marks 1: " + marks1);
        System.out.println("Marks 2: " + marks2);
        System.out.println("Marks 3: " + marks3);
        }
        public int getTotalMarks() {       return
    marks1 + marks2 + marks3;
        }     public double getPercentage() {
    return (getTotalMarks() / 3.0);
        } } Main
```

```
    package lab1c;
    import
    MCA.Student;


    public class Main4 {
        public static void main(String[] args) {
        Student student = new Student("Dharmesh", 85, 90, 95);            student.display();
        int totalMarks = student.getTotalMarks();           double
    percentage = student.getPercentage();

        System.out.println("Total Marks: " + totalMarks);
        System.out.println("Percentage: " + percentage + "%");
        } }
```

After uncommenting

24. **Implement class parent inheriting class child 1. In a simple way extend class child 1 to child 2. write the same function name in all the classes as showdata and write a same variable name "name". using super keyword show the call of parent class function or variable from the respective child class.**

```
package lab1c;

class Parent {  protected
String name;
  public Parent(String name) {
      this.name = name;
 }
 public void showData() {
      System.out.println("Parent Class - Name: " + name);
 }
}
class Child1 extends Parent {  public
Child1(String name) {        super(name);
 }

 @Override
 public void showData() {  super.showData();
      System.out.println("Child1 Class - Name: " + name);
 }
}
class Child2 extends Child1 {  public
Child2(String name) {        super(name);
 }
  @Override  public void
showData() {
super.showData();
      System.out.println("Child2 Class - Name: " + name);
 }
}
public class Main6 {
 public static void main(String[] args) {     Child2
child2 = new Child2(" Durgesh ");
child2.showData();
 } }
```

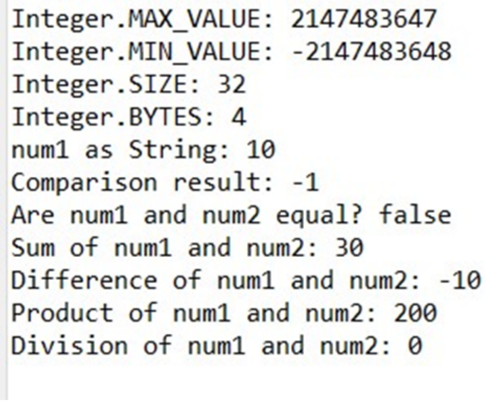


```
<terminated> SolutionArchitech [Java Application] C:\Prograı
Parent Class - Name: Durgesh
Child1 Class - Name: Durgesh
Child2 Class - Name: Durgesh
```

25. Demonstrate wrapper class for Integer class using all function

```
package lab1c;

public class WrapperDemo {
      public static void main(String[] args) {
```

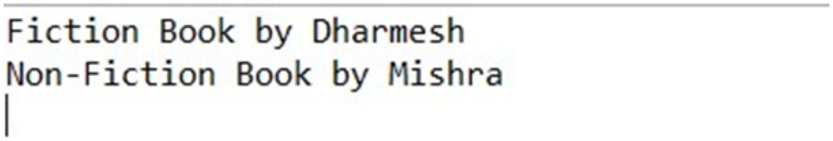
```
      Integer num1 = Integer.valueOf(10);                  Integer
num2 = Integer.valueOf(20);
      int x = 30;          Integer
num3 = x;


      int y = num3.intValue();
      System.out.println("Integer.MAX_VALUE: " + Integer.MAX_VALUE);
      System.out.println("Integer.MIN_VALUE: " + Integer.MIN_VALUE);
      System.out.println("Integer.SIZE: " + Integer.SIZE);
      System.out.println("Integer.BYTES: " + Integer.BYTES);

      String num1Str = num1.toString();
      System.out.println("num1 as String: " + num1Str);

      int compareResult = num1.compareTo(num2);
      System.out.println("Comparison result: " + compareResult);

      boolean isEqual = num1.equals(num2);
      System.out.println("Are num1 and num2 equal? " + isEqual);
      System.out.println("Sum of num1 and num2: " + Integer.sum(num1,
num2));
      System.out.println("Difference of num1 and num2: " + (num1 - num2));
      System.out.println("Product of num1 and num2: " + (num1 * num2));
      System.out.println("Division of num1 and num2: " + (num1 / num2));
      }
      }
```



```
Integer.MAX_VALUE: 2147483647
Integer.MIN_VALUE: -2147483648
Integer.SIZE: 32
Integer.BYTES: 4
num1 as String: 10
Comparison result: -1
Are num1 and num2 equal? false
Sum of num1 and num2: 30
Difference of num1 and num2: -10
Product of num1 and num2: 200
Division of num1 and num2: 0
```

26. Write a program for writing book with author class with different type of books as per domain. Demonstrate the multilevel Inheritance. Demonstrate Super and Final keyword with appropriate assumed data and functions.

```
package lab1c;
 class Book {
 private String author;
  public Book(String author) {
      this.author = author;
 }   public String getAuthor() {
      return author;
 }   public void setAuthor(String author) {
      this.author = author;
 }
}
class FictionBook extends Book {  private
final String domain = "Fiction";

 public FictionBook(String author) {          super(author);
 }
 public final void displayBookDetails() {
      System.out.println("Fiction Book by " + getAuthor());
 }
}
class NonFictionBook extends Book {  private final
String domain = "Non-Fiction";

 public NonFictionBook(String author) {   super(author);
 }
 public final void displayBookDetails() {
      System.out.println("Non-Fiction Book by " + getAuthor());
 }
}
public class Main7 {
 public static void main(String[] args) {
      FictionBook fictionBook = new FictionBook("Durgesh");
      NonFictionBook nonFictionBook = new NonFictionBook("Mandge");
      fictionBook.displayBookDetails();
nonFictionBook.displayBookDetails();
 }
}
```

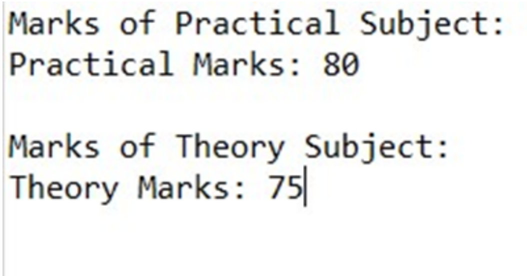


```
Fiction Book by Dharmesh
Non-Fiction Book by Mishra
```

27. Create an interface for the class Subject. Display the marks of the subject derived from practical and theory class

```
package lab1c;
 interface Subject1 {
void displayMarks();
}
class Practical implements Subject1 {  private int
practicalMarks;
  public Practical(int marks) {
this.practicalMarks = marks;
 }
 public void displayMarks() {
      System.out.println("Practical Marks: " + practicalMarks);
 }
}
```

```
    class Theory implements Subject1 {  private int
    theoryMarks;
      public Theory(int marks) {
          this.theoryMarks = marks;
     }
     public void displayMarks() {
          System.out.println("Theory Marks: " + theoryMarks);
     }
    }
    public class Main8 {
     public static void main(String[] args) {

          Practical practicalSubject = new Practical(80);
          Theory theorySubject = new Theory(75);


          System.out.println("Marks of Practical Subject:");      practicalSubject.displayMarks();

          System.out.println("\nMarks of Theory Subject:");      theorySubject.displayMarks();
} }
```



```
Marks of Practical Subject:
Practical Marks: 80

Marks of Theory Subject:
Theory Marks: 75
```

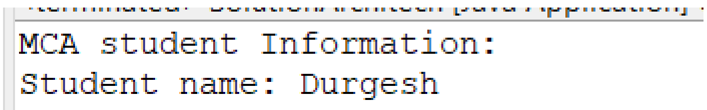
28. Write a student admission process like student information, Roll no. allocation process, Exam process(To take marks from the user and display marks) in a package "Student". Create the object for MCA and Comp. Engg student by importing package using fully qualified way.

StudentInfo

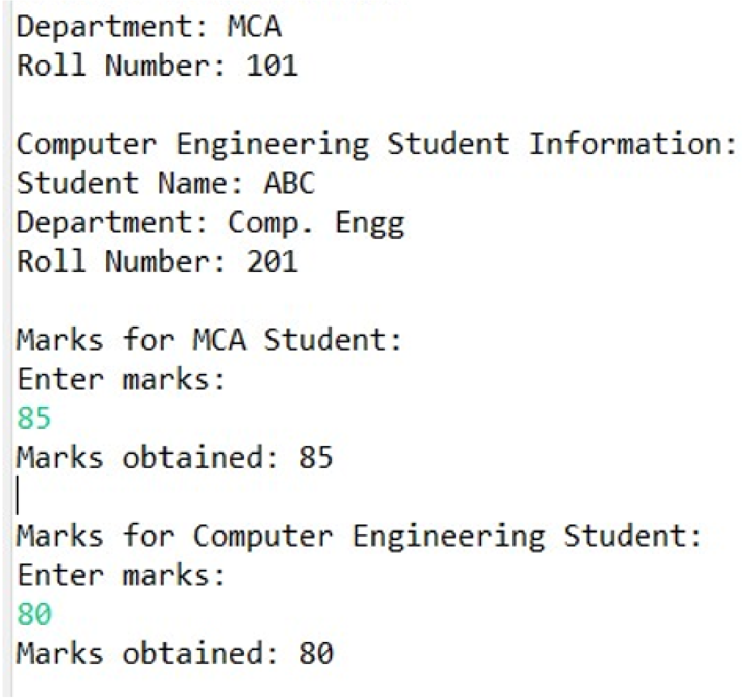
```
package Student;

import java.util.Scanner;
 public class StudentInfo {
      private String name;    private
String department;            private
int rollNumber;

      public StudentInfo(String name, String department, int rollNumber) {
this.name = name;             this.department = department;           this.rollNumber =
rollNumber;
      }
      public void displayInfo() {
      System.out.println("Student Name: " + name);
      System.out.println("Department: " + department);
      System.out.println("Roll Number: " + rollNumber);
      }
}
```

ExamProcess

```
package Student;

import java.util.Scanner;

public class ExamProcess {          public
static void takeMarks() {
      Scanner scanner = new Scanner(System.in);
      System.out.println("Enter marks:");               int
marks = scanner.nextInt();
      System.out.println("Marks obtained: " + marks);
      } }
Main
package lab1c;
 import Student.StudentInfo;
import Student.ExamProcess;

public class Main9 {
      public static void main(String[] args) {
      StudentInfo mcaStudent = new StudentInfo("Dharmesh", "MCA", 101);
      StudentInfo compEnggStudent = new StudentInfo("ABC", "Comp. Engg", 201);
      System.out.println("MCA Student Information:");       mcaStudent.displayInfo();

      System.out.println("\nComputer Engineering Student Information:");
compEnggStudent.displayInfo();

      System.out.println("\nMarks for MCA Student:");
      ExamProcess.takeMarks();

      System.out.println("\nMarks for Computer Engineering Student:");
      ExamProcess.takeMarks();
      } }
```



```
MCA student Information:
Student name: Durgesh
```

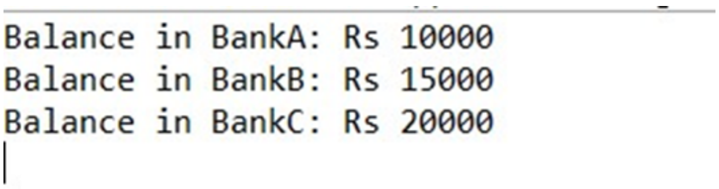


```
Department: MCA
Roll Number: 101

Computer Engineering Student Information:
Student Name: ABC
Department: Comp. Engg
Roll Number: 201

Marks for MCA Student:
Enter marks:
85
Marks obtained: 85

Marks for Computer Engineering Student:
Enter marks:
80
Marks obtained: 80
```

29. **Create an abstract class 'Bank' with an abstract method 'getBalance'. $100, $150 and $200 are deposited in banks A, B and C respectively. 'BankA', 'BankB' and 'BankC' are subclasses of class 'Bank', each having a method named 'getBalance'. Call this method by creating an object of each of the three classes**

```
package lab1c;

abstract class Bank {
abstract int getBalance();
}
class BankA extends Bank {
private int balance = 10000;

    @Override       int
getBalance() {      return
balance;
    }
}  class BankB extends Bank {
private int balance = 15000;

    @Override       int
getBalance() {      return
balance;
    }
}
```

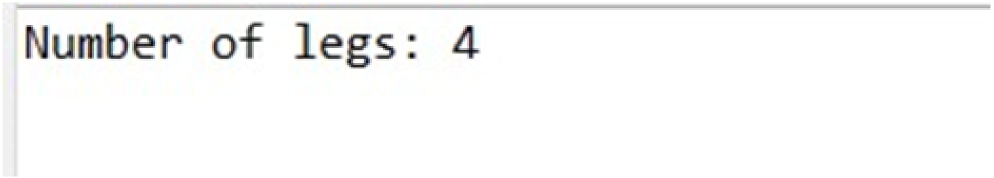
```
    class BankC extends Bank {
    private int balance = 20000;

        @Override       int
    getBalance() {       return
    balance;
        }
    }
    public class Main10 {
        public static void main(String[] args) {
        BankA bankA = new BankA();
        BankB bankB = new BankB();
        BankC bankC = new BankC();

        System.out.println("Balance in BankA: Rs " + bankA.getBalance());

                System.out.println("Balance in BankB: Rs " + bankB.getBalance());
                System.out.println("Balance in BankC: Rs " + bankC.getBalance());
        }
}
```



```
Balance in BankA: Rs 10000
Balance in BankB: Rs 15000
Balance in BankC: Rs 20000
```

30. Declare a protected integer attribute called legs, which records the number of legs for this animal.

```
    package lab1c;

    public class Animal1 {
    protected int legs;

        public Animal1(int legs) {       this.legs =
    legs;
        }     public int getLegs() {
        return legs;
        }
        public void setLegs(int legs) {           this.legs
    = legs;
        }
        public static void main(String[] args) {

        Animal1 animal = new Animal1(4);

        System.out.println("Number of legs: " + animal.getLegs());    }
    }
```



```
Number of legs: 4
```

Observation:

In this lab practical I lerned how implement packages , interfaces , inheritance and all Object Oriented Principles with hands on. By troubleshooting errors and experimenting with different solutions, I've gained practical experience in writing efficient and effective Java code.