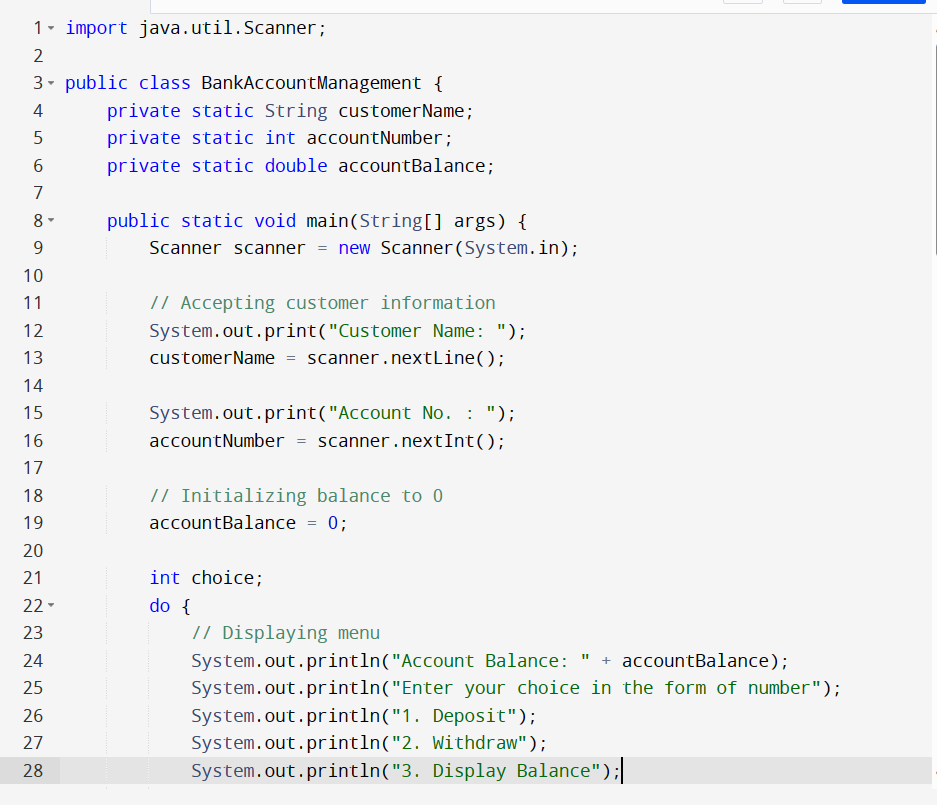
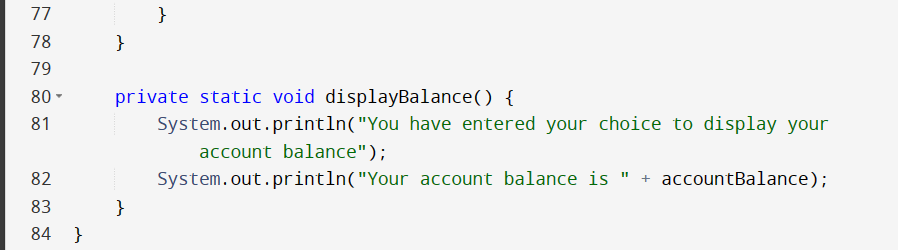
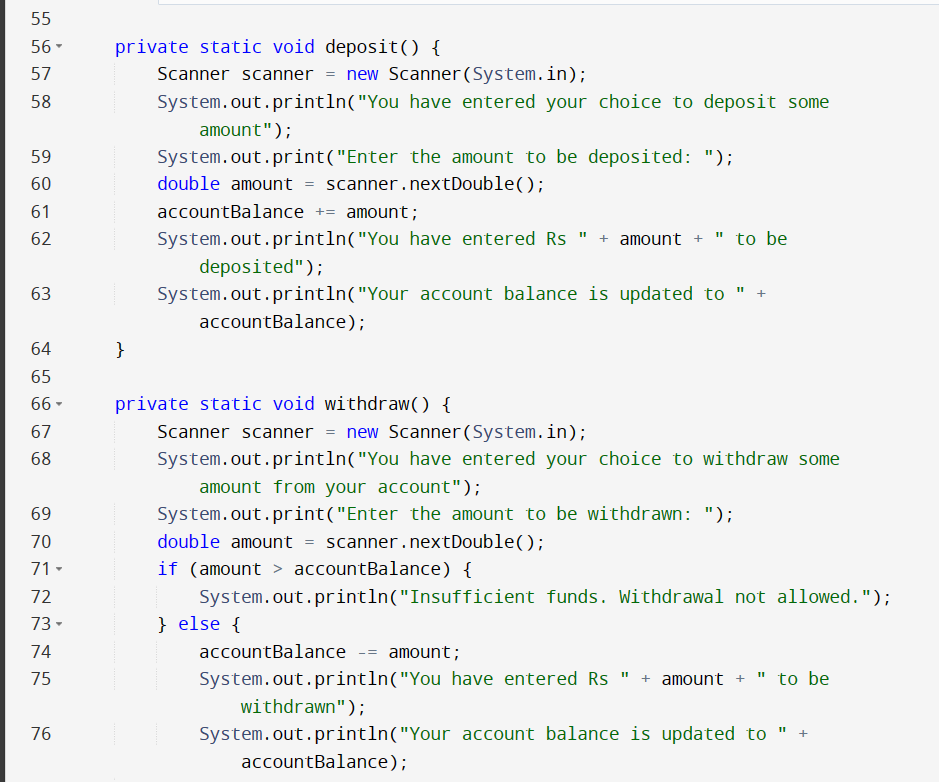
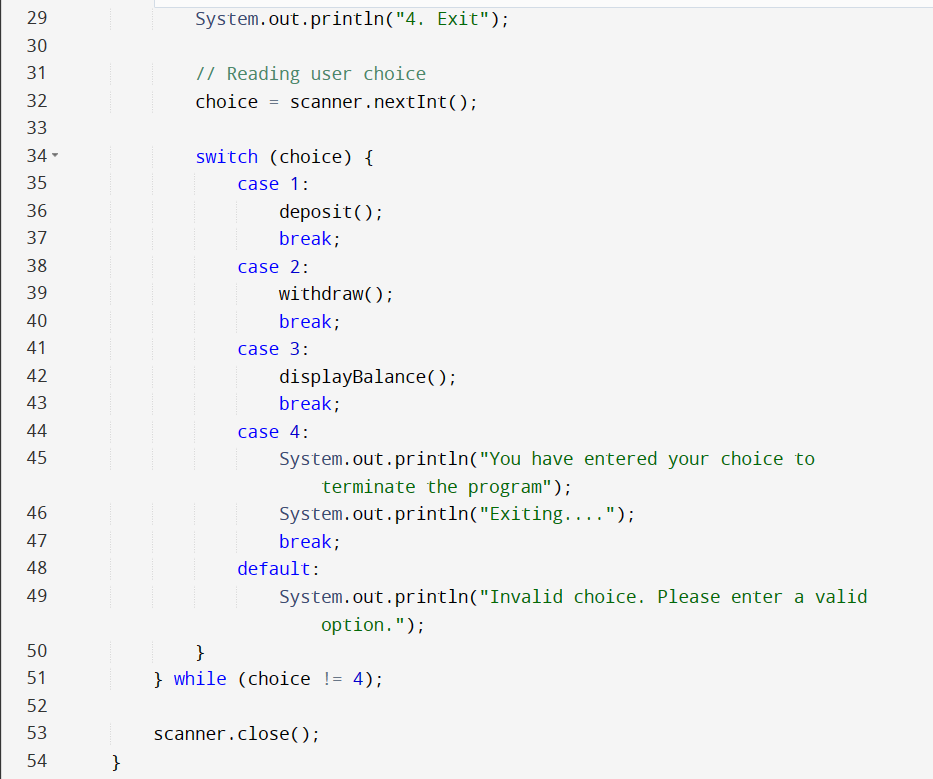
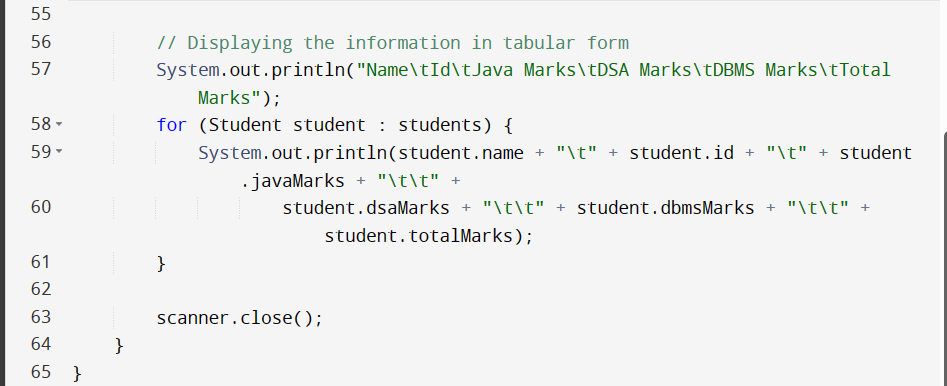
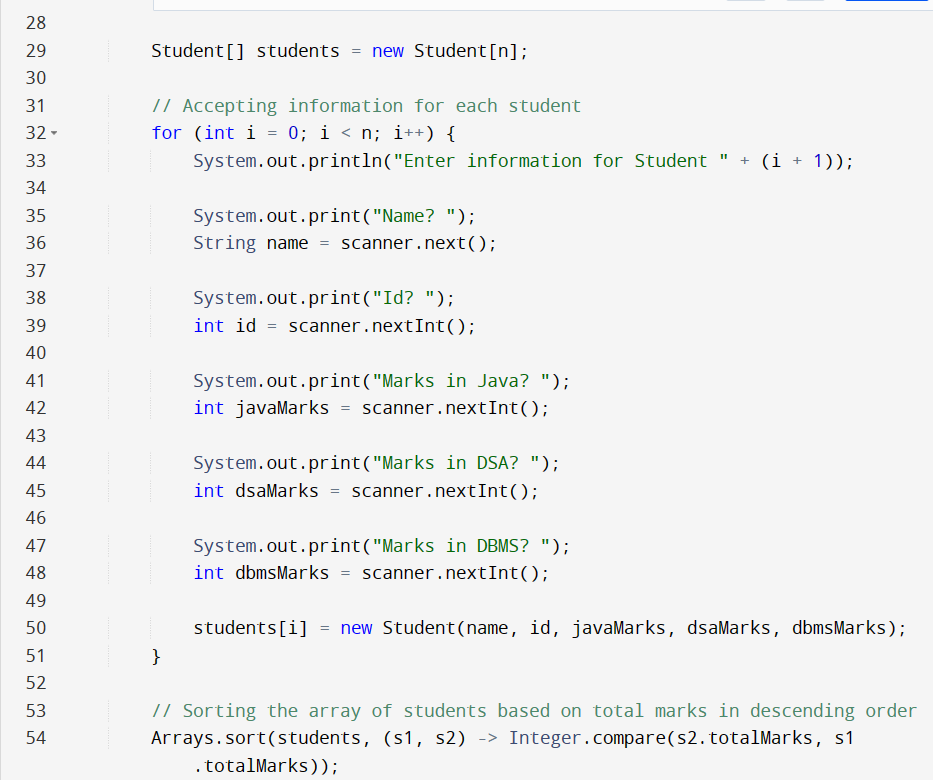
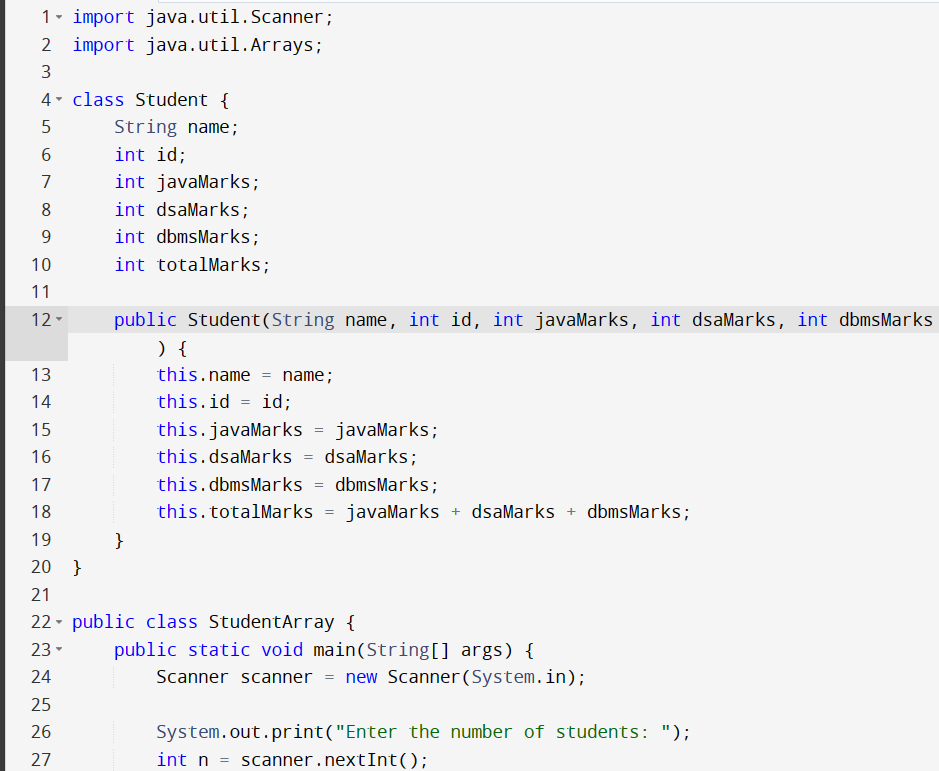
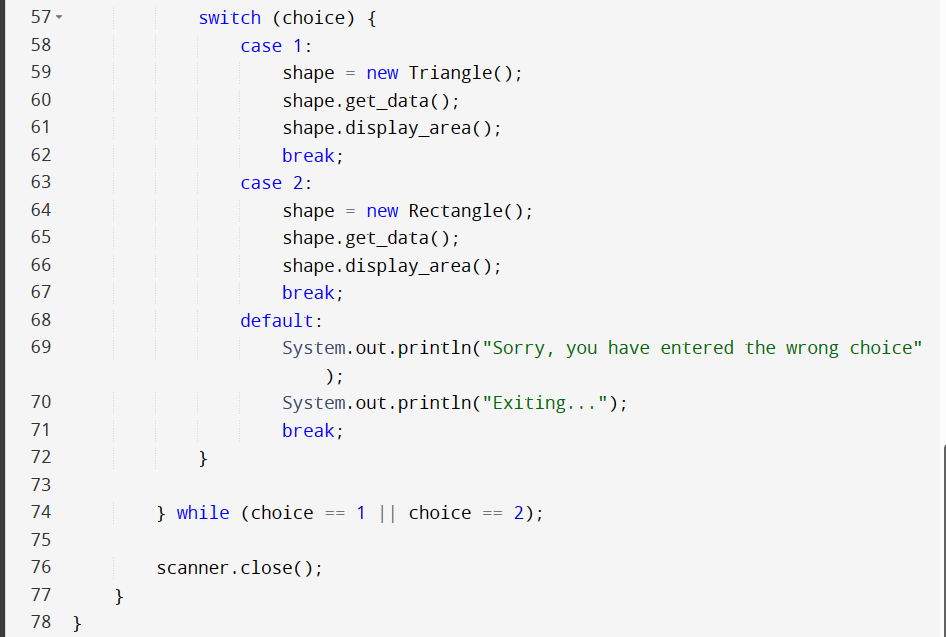
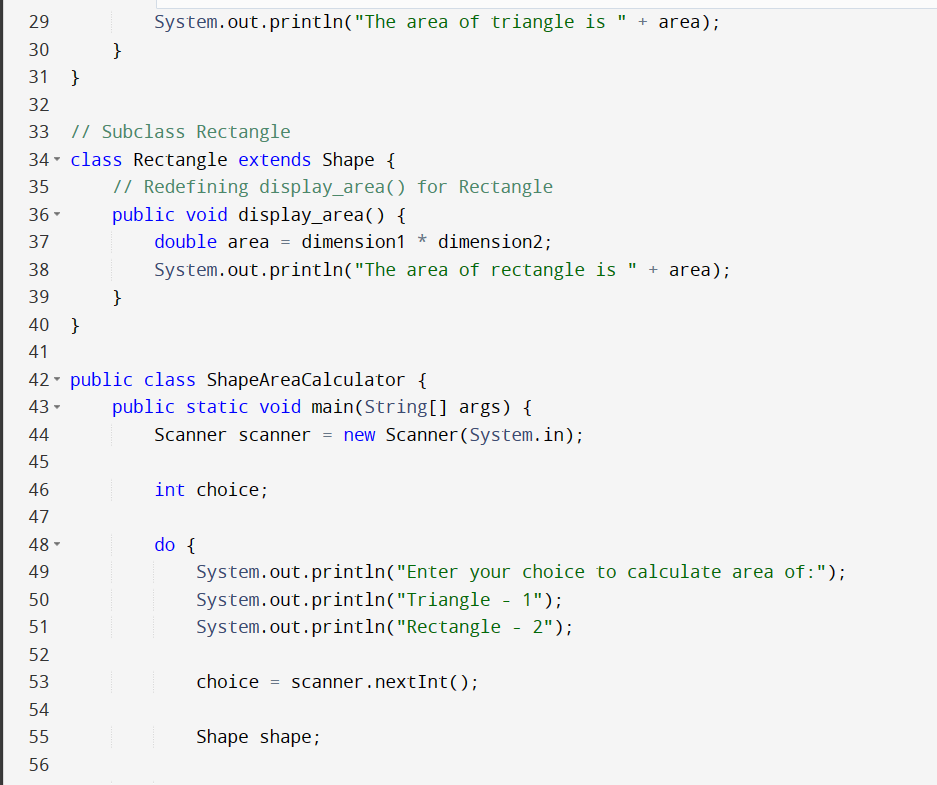
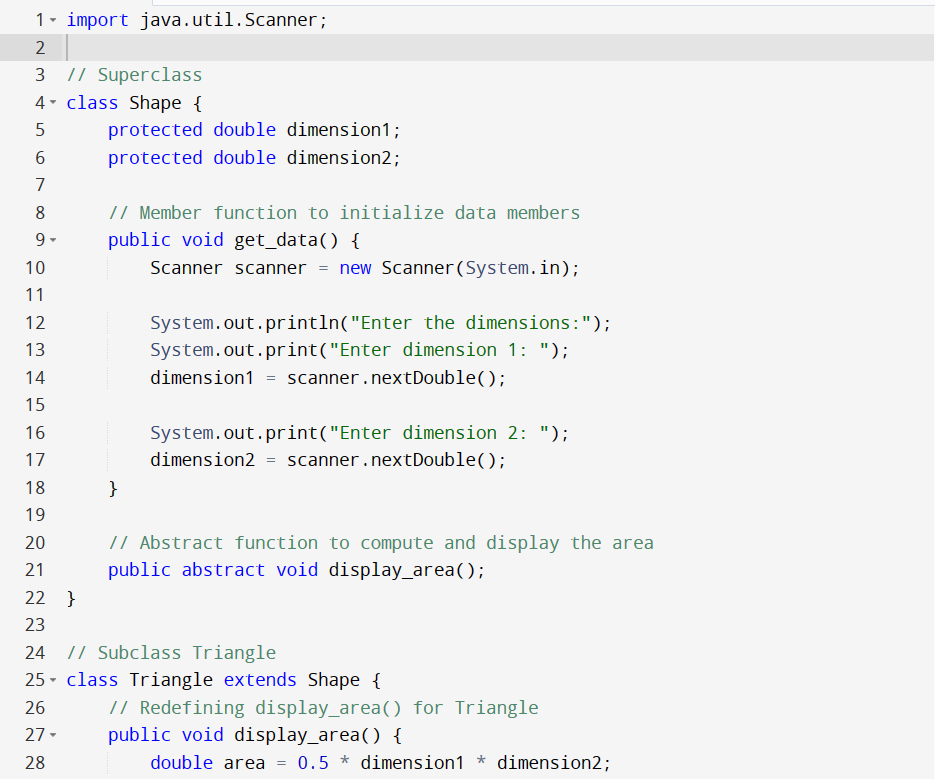
Write a program that accepts customer account information using Scanner classes 1) Customer Name 2) Account Number and provides below menu driven operations on customer account 1) Deposit 2) Withdraw 3) Display Balance 4) Exit the program Expected output: Customer Name: ABC Account No. : 123 Account Balance: 230 Enter your choice in form of number Deposit -1 Withdrawl -2 Display Balance -3 Exit -4 1 You have entered your choice to deposit some amount Enter the amount to be deposited 34 You have entered Rs 34 to be deposited Your account balance is updated to 264 Enter your choice in form of number Deposit -1 Withdrawl -2 Display Balance -3 Exit -4 2 You have entered your choice to withdraw some amount from your account Enter the amount to be withdrawn 64 You have entered Rs 64 to be withdrawn Your account balance is updated to 200 Enter your choice in form of number Deposit -1 Withdrawl -2 Display Balance -3 Exit -4 3 You have entered your choice to display your account balance Your account balance is 200 Enter your choice in form of number Deposit -1 Withdrawl -2 Display Balance -3 Exit -4 You have entered your choice to terminate the program Exiting...  
 **  
**  
  
  
  
  
Write a program to create an array of the objects of class Student. The class should have field members name, id, marks obtained in three subjects namely JAVA, DSA, DBMS and calculate student's total marks. Accept information of ‘n’ students using Scanner class and display them in tabular form in descending order of total marks obtained by student. Sample Output: Enter the no of students Enter information for Student 1 Name? A Id? 1 Marks in Java? 3 Marks in DSA? 9 Marks in DBMS? 12 Name? B Id? 2.................enter this information using Scanner Class Name Id Java Marks DSA Marks DBMS Marks Total Marks A 1 3 9 12 24 B 2 3 8 12 23 C 3 3 8 10 21 D 4 2 9 9 20 E 5 2 8 6 16  
  


**Import Statements: These lines import the necessary classes from the java.util package - Scanner for input and Arrays for array manipulation.  
  
Student Class: Defines a class named Student to represent student information. It has fields for name, id, javaMarks, dsaMarks, dbmsMarks, and totalMarks. The totalMarks field is calculated as the sum of javaMarks, dsaMarks, and dbmsMarks in the constructor.**

**StudentArray Class: Defines the main class named StudentArray.  
  
Main Method: The main method of the StudentArray class, where the program execution starts.**

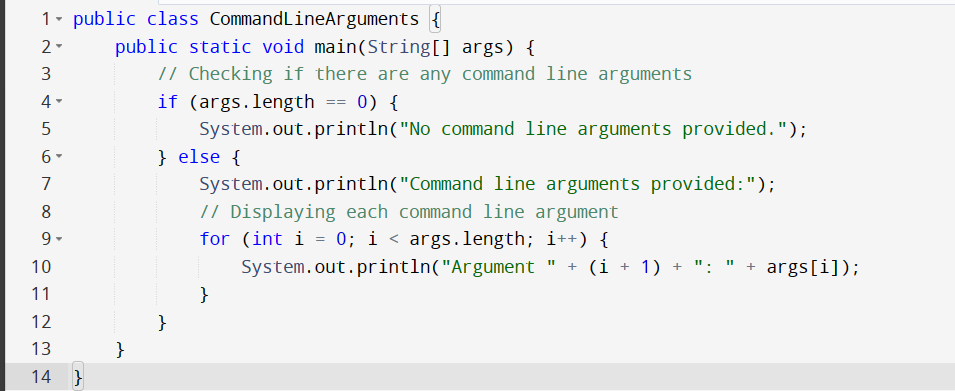
1. **Scanner Initialization: Creates a Scanner object named scanner to read input from the console.**
2. **Number of Students: Asks the user to input the number of students (n).**
3. **Array Initialization: Creates an array students of Student objects with a size of n.**
4. **Input Loop: Uses a for loop to iterate through each student and accepts information for each student using the Scanner.**
5. **Student Object Creation: Creates a new Student object for each student and adds it to the students array.**
6. **Sorting: Uses Arrays.sort to sort the students array based on the totalMarks field in descending order.**
7. **Display Loop: Uses a for-each loop to iterate through the sorted array and prints the information for each student in a tabular format.**
8. **Scanner Closing: Closes the Scanner to release resources.**

Create a superclass called Shape. This class is used to store 2 double type of values that could be used to compute the area. Two subclasses called triangle and rectangle are inherited from superclass shape. In a superclass Shape, a member function get\_data() is used to initialize data members and another member function display\_area() is used to compute and display the area. The display\_area() is used as an abstract function and redefined in derived classes to suit their requirements. Program must accept the choice from the user. Choice 1 for triangle and choice 2 for rectangle. Using scanner class the dimension values must be read and calculate area and display it back to the user. Sample output: Enter the dimensions: Enter dimension 1 10 Enter dimension 2 20 Enter the choice to calculate area of: Triangle - 1 Rectangle- 2 1 You have chosen to calculate the area of triangle The area of triangle is 100 Enter your choice to calculate area of: Triangle - 1 Rectangle- 2 2 You have chosen to calculate the area of rectangle The area of triangle is 200 Enter your choice to calculate area of: Triangle - 1 Rectangle- 2 3 Sorry you have entered wrong choice Exiting...(**in the below program it would be abstract class shape**)

****

**This declares a superclass named Shape. It has two protected double variables (dimension1 and dimension2) to store dimensions that could be used to compute the area.  
  
This method (get\_data()) is used to accept dimensions from the user using the Scanner class.  
  
This declares an abstract method (display\_area()) which is meant to be implemented by subclasses. It will be used to compute and display the area.  
  
This defines a subclass Triangle that extends the Shape superclass. It overrides the display\_area() method to compute and display the area of a triangle.  
  
This defines another subclass Rectangle that also extends the Shape superclass. It overrides the display\_area() method to compute and display the area of a rectangle.  
  
This is the main class ShapeAreaCalculator. It contains the main method, which serves as the entry point of the program. The program uses a do-while loop to repeatedly prompt the user for their choice to calculate the area of a triangle or rectangle. Depending on the choice, it creates an instance of the corresponding subclass (Triangle or Rectangle), calls the get\_data() method to input dimensions, and then calls the display\_area() method to compute and display the area. The loop continues as long as the user enters 1 or 2. If the user enters any other choice, the program prints an error message and exits. Finally, the Scanner is closed to release resources**.

Write a program to accept command line arguments from the user.

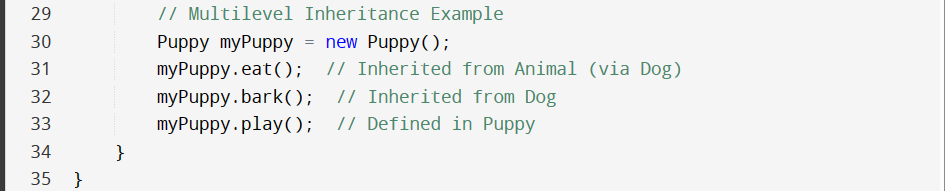
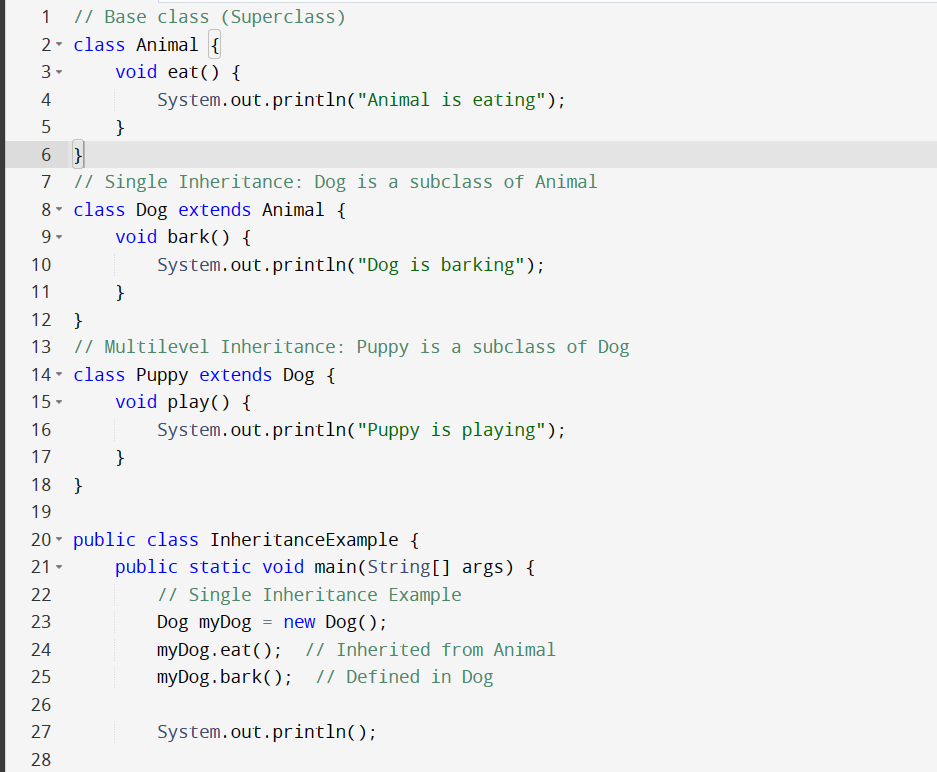
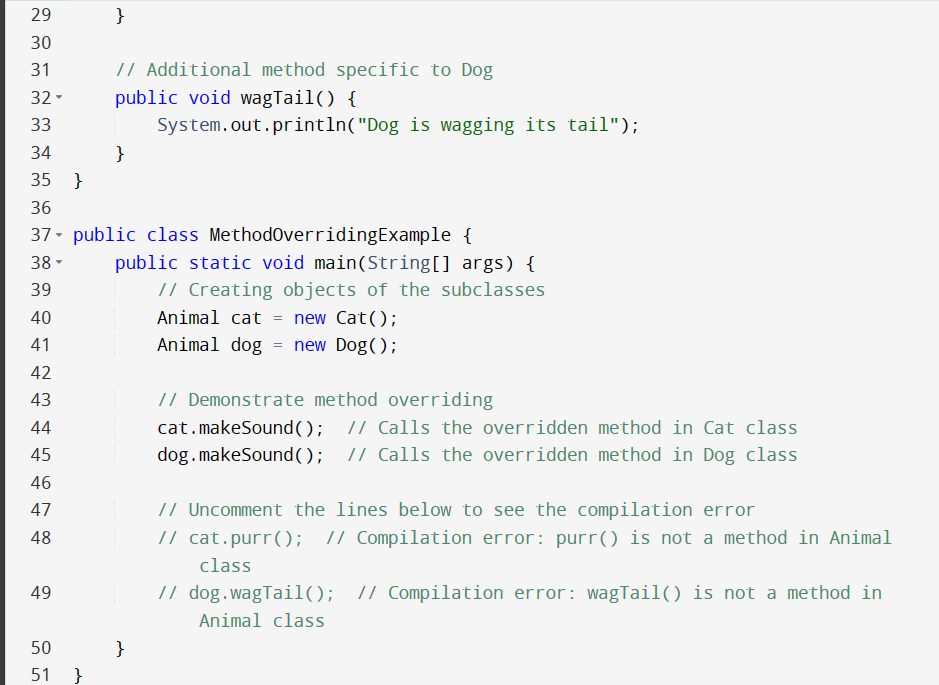
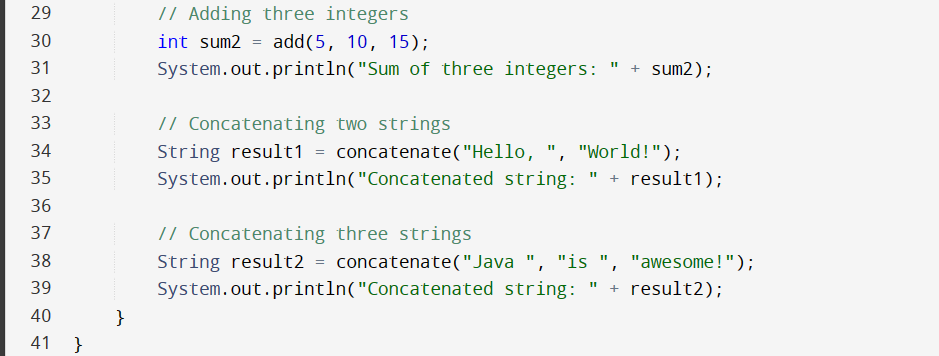
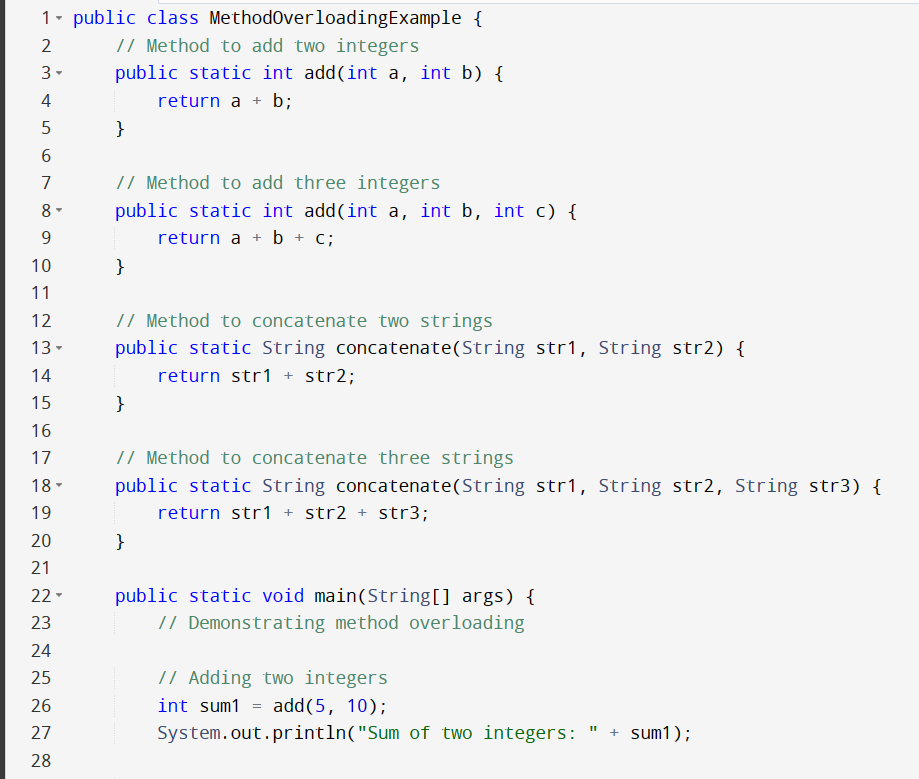
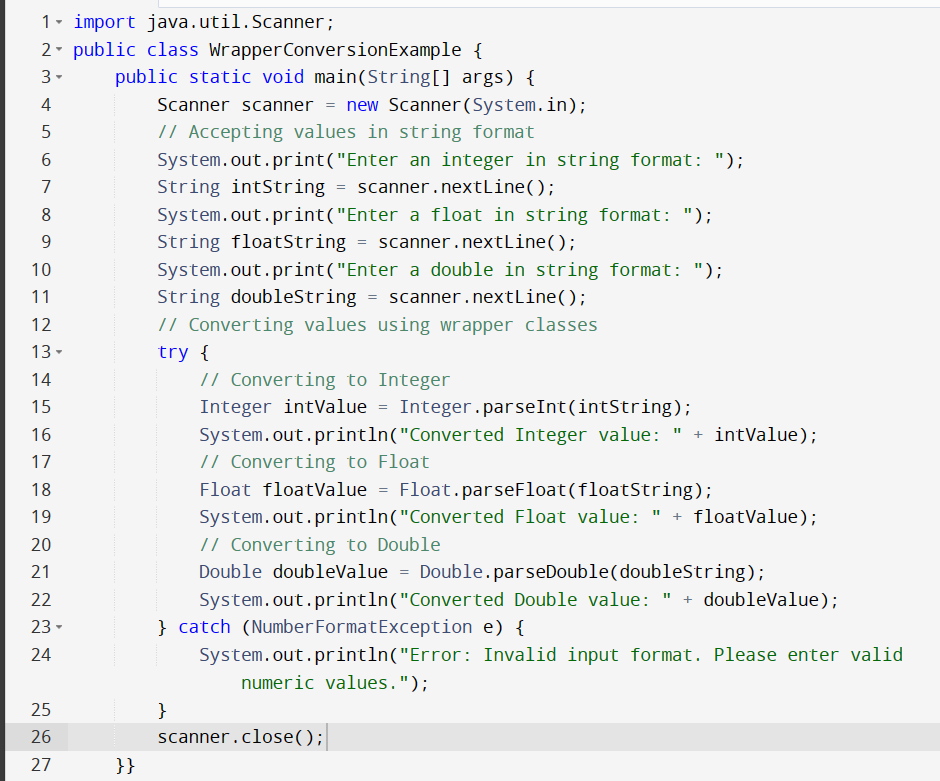
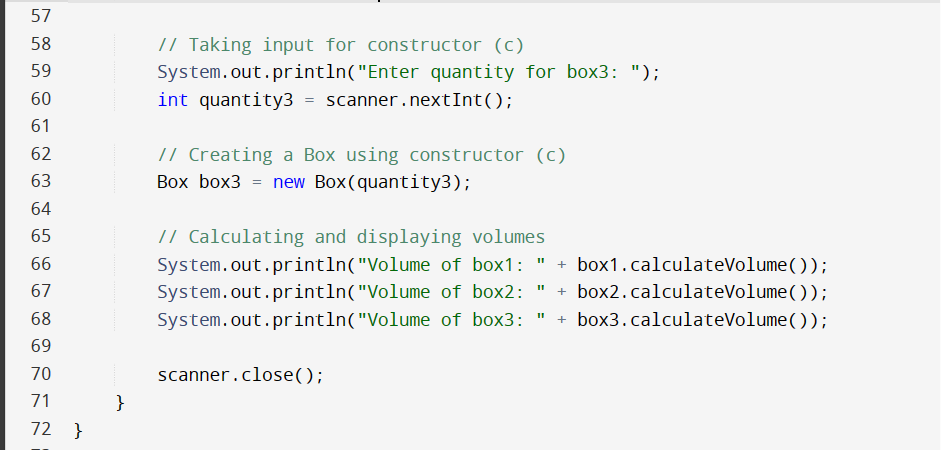
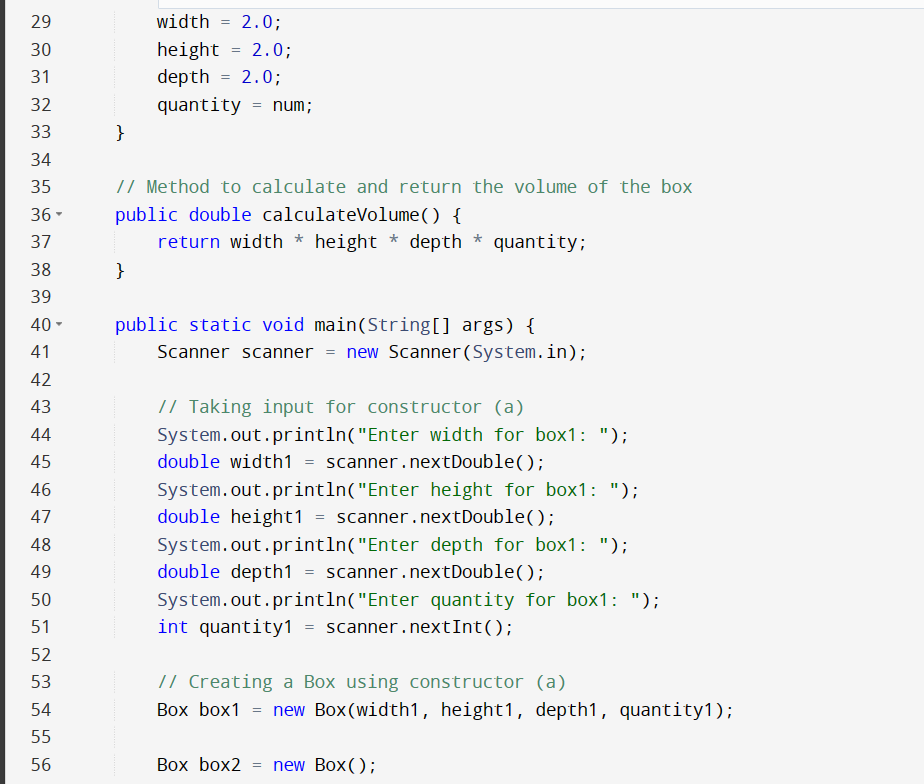
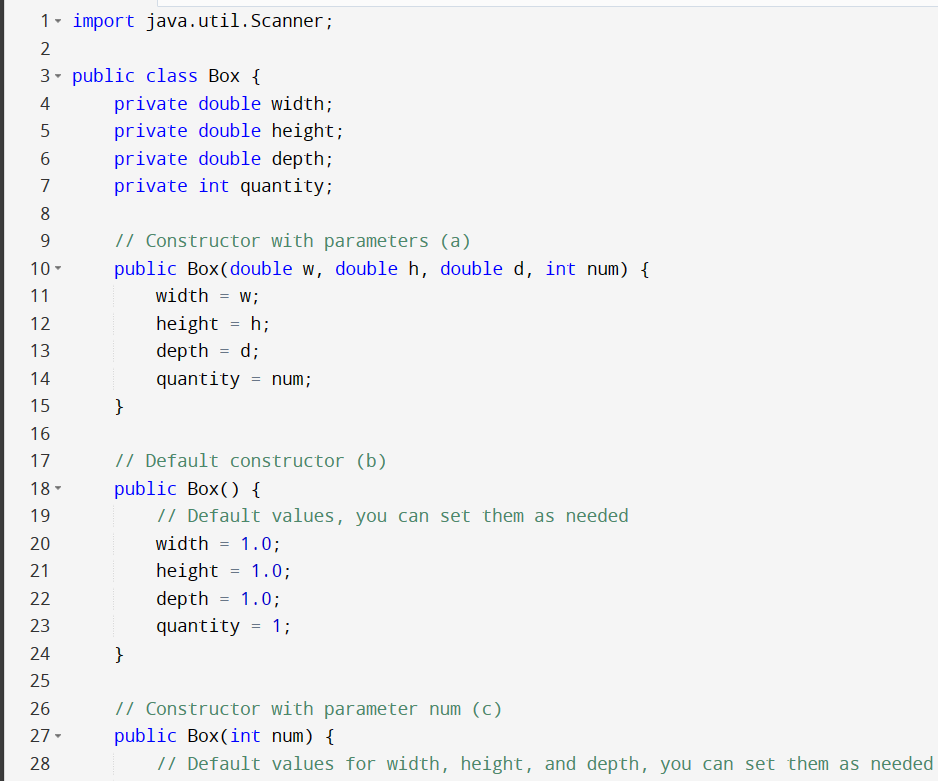
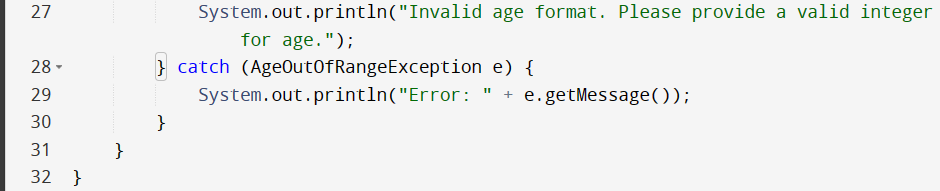
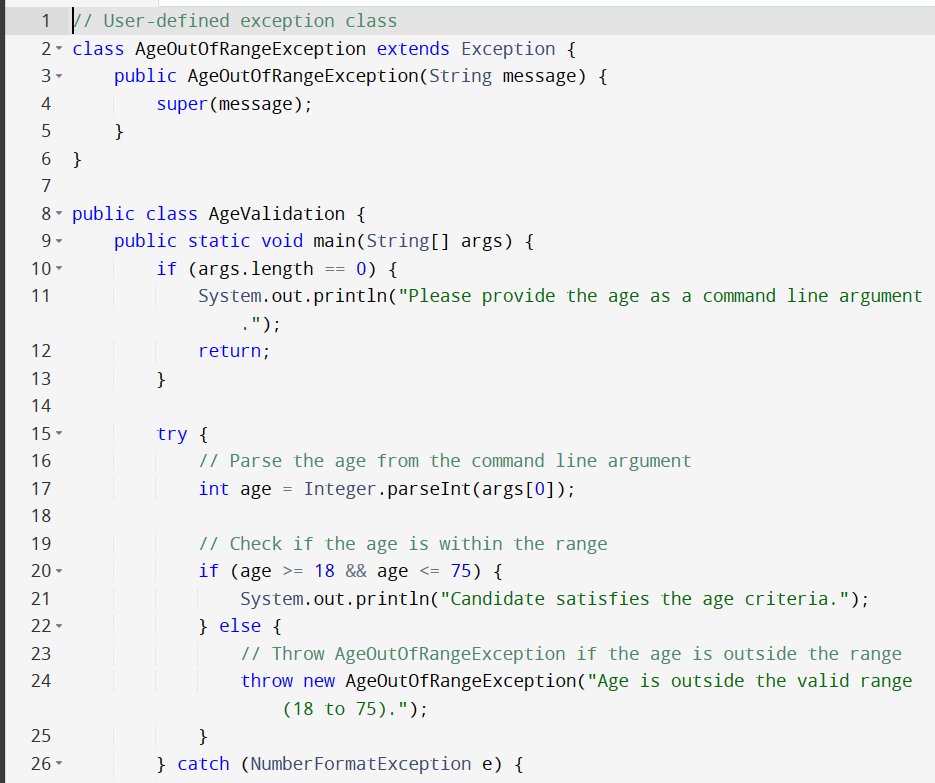


**Here's how you can run this program with command line arguments:**

1. **Open a text editor and save the above code in a file named CommandLineArguments.java.**
2. **Open a command prompt or terminal and navigate to the directory where you saved the file.**
3. **Compile the program using the command: javac CommandLineArguments.java**
4. **Run the program with command line arguments: java CommandLineArguments arg1 arg2 arg3**

**Replace arg1, arg2, and arg3 with the actual command line arguments you want to provide.**

Write a program to "throw" an exception if age is below 18 (print "Access denied"). If age is 18 or older, print "Access granted" to the user.  
  
  
  
  
Write a program to demonstrate single and multilevel inheritance in Java.

  
  
  
Write a program to demonstrate method overriding in Java  
  
  
  
  
  
  
  
  
Write a program to demonstrate method overloading in Java.  
  
  
  
  
  
  
  
  
Write a program to accept values in String format and convert them into Interger, Float etc using wrapper class.  
   
  
  
Write a program to demonstrate constructor overloading for a class Box which has parameters defined as: a. Box(double w, double h, double d, int num) b. Box() c.Box(int num) and calculate volume of the box on the given parameters.  
  
  
  
Write a program which accepts age of the candidate as a command line argument and checks whether it is within the range (between 18 to 75) or not. If it is within the range then it displays “Candidate satisfy the age criteria”, if not then it throws the exception of user defined class “AgeOutOfRangeException”  
  
  
  
  
Write a program to accept values and display it in 2D jagged arrray.  
  
  
  
Create a class Student having method setRollnumber() and displayRollnumber() to accept roll no using Scanner class and display roll number respectively. Derive a sub-class Test having methods getMarks() and displayMarks() to accept and display marks for any three subjects respectively. Create an interface Marks having member bonusMarks=5 and displayBonus() method declaration. Derive a class Result from Test and implement Marks interface. Class Result should have method display() which should calculate the sum of marks for all three subjects, add bonusMarks and display all the details of student. Result should override displayBonus() method to display the bonus marks awarded. Expected Output: Enter your roll number: 1 You have entered your roll number as 1 Please enter your marks for subject A 20 Please enter your marks for subject B 50 Please enter your marks for subject C 30 You entered marks for subject A, B, C as 10,50,30 How much bonus marks can be awarded to a student? 10 You have entered that total of 10 bonus marks can be awarded to the student Student with Roll No 1 have secured total marks as 110 out of which 10 marks are bonus marks awarded to the student  
  
**import java.util.Scanner;**

**// Interface Marks**

**interface Marks {**

**int bonusMarks = 5;**

**void displayBonus();**

**}**

**// Class Student**

**class Student {**

**private int rollNumber;**

**// Method to set roll number**

**void setRollNumber() {**

**Scanner scanner = new Scanner(System.in);**

**System.out.print("Enter your roll number: ");**

**rollNumber = scanner.nextInt();**

**System.out.println("You have entered your roll number as " + rollNumber);**

**}**

**// Method to display roll number**

**void displayRollNumber() {**

**System.out.println("Student with Roll No " + rollNumber);**

**}**

**}**

**// Subclass Test**

**class Test extends Student implements Marks {**

**private int marksA;**

**private int marksB;**

**private int marksC;**

**// Method to get marks for three subjects**

**void getMarks() {**

**Scanner scanner = new Scanner(System.in);**

**System.out.print("Please enter your marks for subject A: ");**

**marksA = scanner.nextInt();**

**System.out.print("Please enter your marks for subject B: ");**

**marksB = scanner.nextInt();**

**System.out.print("Please enter your marks for subject C: ");**

**marksC = scanner.nextInt();**

**System.out.println("You entered marks for subject A, B, C as " + marksA + ", " + marksB + ", " + marksC);**

**}**

**// Method to display marks for three subjects**

**void displayMarks() {**

**System.out.println("Student with Roll No " + rollNumber + " have secured marks as " +**

**marksA + ", " + marksB + ", " + marksC);**

**}**

**// Implementation of displayBonus() from Marks interface**

**@Override**

**public void displayBonus() {**

**Scanner scanner = new Scanner(System.in);**

**System.out.print("How much bonus marks can be awarded to a student? ");**

**int bonus = scanner.nextInt();**

**System.out.println("You have entered that a total of " + bonus + " bonus marks can be awarded to the student");**

**}**

**// Method to calculate and display total marks with bonus**

**void display() {**

**int totalMarks = marksA + marksB + marksC;**

**int totalMarksWithBonus = totalMarks + Marks.bonusMarks;**

**System.out.println("Student with Roll No " + rollNumber + " have secured total marks as " +**

**totalMarks + " out of which " + Marks.bonusMarks + " marks are bonus marks awarded to the student");**

**}**

**}**

**// Result class deriving from Test and implementing Marks interface**

**class Result extends Test implements Marks {**

**// Empty for now, no additional methods needed**

**}**

**public class StudentResultDemo {**

**public static void main(String[] args) {**

**Result result = new Result();**

**// Set roll number**

**result.setRollNumber();**

**// Get and display marks**

**result.getMarks();**

**result.displayMarks();**

**// Display bonus marks**

**result.displayBonus();**

**// Calculate and display total marks with bonus**

**result.display();**

**}**

**}**