# **Experiment 3**

# **Program** **Based on the concept of inheritance**

1. **Write a Java program to add two numbers using single inheritance. Accept these two numbers from the user in base class and display the sum of these two numbers in derived class.**

# **Program:**

// additon using simple inheritance

import java.util.Scanner;

 class add {

    int first\_no;

    int second\_no;

   public void take\_num(){

    Scanner sc=new Scanner (System.in);

    System.out.println("Enter first no:");

    first\_no= sc.nextInt();

    System.out.println("Enter second no:");

    second\_no= sc.nextInt();

    sc.close();

   }

}

class child1 extends add{

    void display(){

        int sum= first\_no+second\_no;

        System.out.println("Additon of "+first\_no+"+"+second\_no+"="+sum);

    }

}

public class exp3\_1 {

    public  static void main (String args[]){

        child1 obj= new child1();

        obj.take\_num();

        obj.display();

}

}

# **Output:**

Enter first no:

10

Enter second no:

12

Addition of 10+12=22

1. **Write a Java program to calculate the percentage of a student using multi-level inheritance. Accept the marks of three subjects in base class. A class will derived from the above mentioned class which includes a function to find the total marks obtained and another class derived from this class which calculates and displays the percentage of student.**

# **Program:**

  /\* Multilevel Inheritance \*/

  import java.util.Scanner;

  class Student {

      int math;

      int science;

      int Eng;

      public void get\_marks() {

          Scanner sc = new Scanner(System.in);

          System.out.println("Enter marks of maths:");

          math = sc.nextInt();

          System.out.println("Enter marks of science:");

          science = sc.nextInt();

          System.out.println("Enter marks of English:");

          Eng = sc.nextInt();

      }

  }

  class total\_mark extends Student {

      double Total\_Marks; // Declaration moved here

      // Method to calculate Total\_Marks

      public void calculateTotalMarks() {

          Total\_Marks = math + science + Eng;

      }

      public void display1() {

          System.out.println("Total Marks: " + Total\_Marks);

      }

}

  class Percent extends total\_mark {

      public void calculatePercentage() {

          double percentage = Total\_Marks / 3.0;

          System.out.println("Percentage: " + percentage);

      }

  }

  public class exp3\_2 {

      public static void main(String args[]) {

          Percent obj1 = new Percent();

          obj1.get\_marks(); // Input marks

          obj1.calculateTotalMarks(); // Calculate total marks

          obj1.display1(); // Display total marks

          obj1.calculatePercentage(); // Calculate and display percentage

      }

  }

# **Output:**

Enter marks of maths:

88

Enter marks of science:

89

Enter marks of English:

90

Total Marks: 267.0

Percentage: 89.0

1. **Write a Java program to design a base class Person (name, address, phone\_no). Derive a class Employee (eno, ename) from Person. Derive a class Manager (designation, department name, basic-salary) from Employee. Write a menu driven program to:**

**a. Accept all details of 'n' managers.**

**b. Display manager having highest salary**

# **Program:**

import java.util.Scanner;

class Person {

    String name;

    String address;

    String phone\_no;

    public void get\_info() {

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter Name:");

        name = sc.nextLine();

        System.out.println("Enter Address:");

        address = sc.nextLine();

        System.out.println("Enter Phone no:");

        phone\_no = sc.nextLine();

    }

}

class Employee extends Person {

    String ename;

    String eno;

    public void get\_einfo() {

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter eName:");

        ename = sc.nextLine();

        System.out.println("Enter eno:");

        eno = sc.nextLine();

    }

}

class Manager extends Employee {

    String designation;

    String departmentName;

    double basicSalary;

    public void get\_additional\_info() {

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter Designation:");

        designation = sc.nextLine();

        System.out.println("Enter Department Name:");

        departmentName = sc.nextLine();

        System.out.println("Enter Basic Salary:");

        basicSalary = sc.nextDouble();

    }

}

public class exp3\_3 {

    public static void main(String args[]) {

        Scanner sc = new Scanner(System.in);

        int n;

        System.out.println("Enter the number of managers:");

        n = sc.nextInt();

        Manager[] managers = new Manager[n];

        for (int i = 0; i < n; i++) {

            System.out.println("Enter details of manager " + (i + 1) + ":");

            managers[i] = new Manager();

            managers[i].get\_info();

            managers[i].get\_einfo();

            managers[i].get\_additional\_info();

        }

        // Finding manager with highest salary

        Manager managerWithHighestSalary = managers[0];

        for (int i = 1; i < n; i++) {

            if (managers[i].basicSalary > managerWithHighestSalary.basicSalary) {

                managerWithHighestSalary = managers[i];

            }

        }

        // Displaying manager with highest salary

        System.out.println("Manager with the highest salary:");

        System.out.println("Name: " + managerWithHighestSalary.name);

        System.out.println("Employee Number: " + managerWithHighestSalary.eno);

        System.out.println("Designation: " + managerWithHighestSalary.designation);

        System.out.println("Department Name: " + managerWithHighestSalary.departmentName);

        System.out.println("Basic Salary: " + managerWithHighestSalary.basicSalary);

    }

}

# **Output:**

Enter the number of managers:

4

Enter details of manager 1:

Enter Name:

Prathmesh Patil

Enter Address:

Ozarde

Enter Phone no:

8767173285

Enter eName:

jrPrath

Enter eno:

1022031066

Enter Designation:

Software Engineer

Enter Department Name:

CSE

Enter Basic Salary:

90000

Enter details of manager 2:

Enter Name:

Raj Shamani

Enter Address:

Mumbai

Enter Phone no:

9876858954

Enter eName:

rajsh99

Enter eno:

56464633

Enter Designation:

Full stack web developer

Enter Department Name:

IT

Enter Basic Salary:

89000

Enter details of manager 3:

Enter Name:

Rushikesh Phalle

Enter Address:

Aitavde

Enter Phone no:

7868584838

Enter eName:

Rushi6768

Enter eno:

1022031055

Enter Designation:

Cloud administrator

Enter Department Name:

CSE

Enter Basic Salary:

89999

Enter details of manager 4:

Enter Name:

Swapnil Patil

Enter Address:

Islampur

Enter Phone no:

5747374757

Enter eName:

Swap1163

Enter eno:

1929292929

Enter Designation:

graphic designer

Enter Department Name:

CSE

Enter Basic Salary:

89999

Manager with the highest salary:

Name: Prathmesh Patil

Employee Number: 1022031066

Designation: Software Engineer

Department Name: CSE

Basic Salary: 90000.0

1. **Write a Java program to define a base class Item (item-no, name, price). Derive a class Discounted-Item (discount-percent). A customer purchases 'n' items. Display the item-wise bill and total amount using appropriate format.**

# **Program:**

 import java.util.Scanner;

 class Item {

     String item\_name;

     int item\_no;

     double cost\_price;

     public Item(int item\_no, String item\_name, double cost\_price) {

         this.item\_name = item\_name;

         this.item\_no = item\_no;

         this.cost\_price = cost\_price;

     }

     public double getCostPrice() {

         return cost\_price;

     }

     public String toString() {

         return "Item-no: " + item\_no + ", Name: " + item\_name + ", Cost Price: $" + cost\_price;

     }

 }

 class DiscountedItem extends Item {

     double discount;

     double selling\_price;

     public DiscountedItem(int item\_no, String item\_name, double cost\_price, double discount) {

         super(item\_no, item\_name, cost\_price);

         this.discount = discount;

         calculateDiscountedPrice();

     }

     public void calculateDiscountedPrice() {

         selling\_price = cost\_price - (cost\_price \* (discount / 100));

     }

     public double getDiscountedPrice() {

         return selling\_price;

     }

     public String toString() {

         return super.toString() + ", Discounted Price: $" + selling\_price + " (Discount: " + discount + "%)";

     }

 }

 public class exp3\_4{

     public static void main(String args[]) {

         Scanner sc = new Scanner(System.in);

         // Number of items to purchase

         System.out.print("Enter the number of items to purchase: ");

         int n = sc.nextInt();

         // Array to store items

         Item[] items = new Item[n];

         // Input items data

         for (int i = 0; i < n; i++) {

             System.out.println("Enter details for item " + (i + 1) + ":");

             System.out.print("Item-no: ");

             int item\_no = sc.nextInt();

             sc.nextLine(); // Consume newline

             System.out.print("Name: ");

             String item\_name = sc.nextLine();

             System.out.print("Cost Price: ");

             double cost\_price = sc.nextDouble();

             System.out.print("Discount (%): ");

             double discount = sc.nextDouble();

             items[i] = new DiscountedItem(item\_no, item\_name, cost\_price, discount);

         }

         // Display item-wise bill

         System.out.println("\nItem-wise Bill:");

         System.out.println("\tItem-no\tName\tCost Price\tDiscount\tSelling Price");

         double totalAmount = 0;

         for (Item item : items) {

             totalAmount += ((DiscountedItem) item).getDiscountedPrice();

             System.out.println("\t" + item.item\_no + "\t" + item.item\_name + "\t$" + item.cost\_price + "\t\t" + ((DiscountedItem) item).discount + "%\t\t$" + ((DiscountedItem) item).getDiscountedPrice());

         }

         // Display total amount

         System.out.println("\nTotal Amount: $" + totalAmount);

         sc.close();

     }

 }

# **Output:**

Enter the number of items to purchase: 5

Enter details for item 1:

Item-no: 1

Name: Sugar

Cost Price: 40

Discount (%): 4

Enter details for item 2:

Item-no: 2

Name: Rice

Cost Price: 70

Discount (%): 0

Enter details for item 3:

Item-no: 3

Name: Wheat

Cost Price: 42

Discount (%): 1

Enter details for item 4:

Item-no: 4

Name: jaggery

Cost Price: 48

Discount (%): 0

Enter details for item 5:

Item-no: 5

Name: green lentils

Cost Price: 88

Discount (%): 1

Item-wise Bill:

Item-no Name Cost Price Discount Selling Price

1 Sugar $40.0 4.0% $38.4

2 Rice $70.0 0.0% $70.0

3 Wheat $42.0 1.0% $41.58

4 jaggery $48.0 0.0% $48.0

5 green lentils $88.0 1.0% $87.12

Total Amount: $285.1

1. **Write a Program to design a class having static member function named showcount() which has the property of displaying the number of objects created of the class.**

# **Program:**

class MyClass {

    // Static member variable to keep track of the number of objects created

    private static int count = 0;

    // Constructor to increment the count when an object is created

    public MyClass() {

        count++;

    }

    // Static method to display the number of objects created

    public static void showCount() {

        System.out.println("Number of objects created: " + count);

    }

}

public class exp3\_5 {

    public static void main(String[] args) {

        // Creating objects of MyClass

        MyClass obj1 = new MyClass();

        MyClass obj2 = new MyClass();

        MyClass obj3 = new MyClass();

        // Calling the static method showCount() to display the count

        MyClass.showCount();

    }

}

# **Output:**

Number of objects created: 3

1. **Write a Program using class to process Shopping List for a Departmental Store. The list include details such as the Code No and Price of each item and perform the operations like Adding, Deleting Items to the list and Printing the Total value of a Order.**

# **Program:**

import java.util.ArrayList;

import java.util.Scanner;

class ShoppingItem {

    private int codeNo;

    private String name;

    private double price;

    public ShoppingItem(int codeNo, String name, double price) {

        this.codeNo = codeNo;

        this.name = name;

        this.price = price;

    }

    public int getCodeNo() {

        return codeNo;

    }

    public String getName() {

        return name;

    }

    public double getPrice() {

        return price;

    }

}

class ShoppingList {

    private ArrayList<ShoppingItem> items;

    public ShoppingList() {

        items = new ArrayList<>();

    }

    public void addItem(ShoppingItem item) {

        items.add(item);

    }

    public void deleteItem(int codeNo) {

        for (int i = 0; i < items.size(); i++) {

            if (items.get(i).getCodeNo() == codeNo) {

                items.remove(i);

                System.out.println("Item with code no " + codeNo + " deleted successfully.");

                return;

            }

        }

        System.out.println("Item with code no " + codeNo + " not found.");

    }

    public void printList() {

        System.out.println("\tCode No\tName\tPrice");

        for (ShoppingItem item : items) {

            System.out.println("\t" + item.getCodeNo() + "\t" + item.getName() + "\t$" + item.getPrice());

        }

    }

    public double calculateTotal() {

        double total = 0;

        for (ShoppingItem item : items) {

            total += item.getPrice();

        }

        return total;

    }

}

public class exp3\_6 {

    public static void main(String args[]) {

        Scanner sc = new Scanner(System.in);

        ShoppingList shoppingList = new ShoppingList();

        int choice;

        do {

            System.out.println("\nEnter Choice:");

            System.out.println("\t1. Add item\n\t2. Delete item\n\t3. Print list\n\t4. Print total value\n\t5. Exit\n");

            choice = sc.nextInt();

            switch (choice) {

                case 1:

                    System.out.println("Enter item details:");

                    System.out.print("Code No: ");

                    int codeNo = sc.nextInt();

                    sc.nextLine(); // Consume newline

                    System.out.print("Name: ");

                    String name = sc.nextLine();

                    System.out.print("Price: ");

                    double price = sc.nextDouble();

                    shoppingList.addItem(new ShoppingItem(codeNo, name, price));

                    break;

                case 2:

                    System.out.print("Enter code no of item to delete: ");

                    int deleteCodeNo = sc.nextInt();

                    shoppingList.deleteItem(deleteCodeNo);

                    break;

                case 3:

                    shoppingList.printList();

                    break;

                case 4:

                    System.out.println("Total value of the order: $" + shoppingList.calculateTotal());

                    break;

                case 5:

                    System.out.println("Exiting...");

                    break;

                default:

                    System.out.println("Enter valid choice.");

            }

        } while (choice != 5);

    }

}

# **Output:**

Enter Choice:

1. Add item

2. Delete item

3. Print list

4. Print total value

5. Exit

1

Enter item details:

Code No: 123

Name: Sugar

Price: 40

Enter Choice:

1. Add item

2. Delete item

3. Print list

4. Print total value

5. Exit

1

Enter item details:

Code No: 143

Name: Wheat

Price: 42

Enter Choice:

1. Add item

2. Delete item

3. Print list

4. Print total value

5. Exit

3

Code No Name Price

123 Sugar $40.0

143 Wheat $42.0

Enter Choice:

1. Add item

2. Delete item

3. Print list

4. Print total value

5. Exit

4

Total value of the order: $82.0

Enter Choice:

1. Add item

2. Delete item

3. Print list

4. Print total value

5. Exit

2

Enter code no of item to delete: 123

Item with code no 123 deleted successfully.

Enter Choice:

1. Add item

2. Delete item

3. Print list

4. Print total value

5. Exit

3

Code No Name Price

143 Wheat $42.0

Enter Choice:

1. Add item

2. Delete item

3. Print list

4. Print total value

5. Exit

5

Exiting...

1. **Write a Program to design a student class representing student roll no. and a test class (derived class of student) representing the scores of the student in various subjects and sports class representing the score in sports. The sports and test class should be inherited by a result class having the functionality to add the scores and display the final result for a student.**

# **Program:**

 class Student {

    private int rollNo;

    public Student(int rollNo) {

        this.rollNo = rollNo;

    }

    public int getRollNo() {

        return rollNo;

    }

}

class Test extends Student {

    private int[] testScores;

    public Test(int rollNo, int[] testScores) {

        super(rollNo);

        this.testScores = testScores;

    }

    public int[] getTestScores() {

        return testScores;

    }

}

class Sports extends Student {

    private int sportsScore;

    public Sports(int rollNo, int sportsScore) {

        super(rollNo);

        this.sportsScore = sportsScore;

    }

    public int getSportsScore() {

        return sportsScore;

    }

}

class Result extends Test {

    private int totalScore;

    public Result(int rollNo, int[] testScores, int sportsScore) {

        super(rollNo, testScores);

        this.totalScore = calculateTotalScore(testScores, sportsScore);

    }

    private int calculateTotalScore(int[] testScores, int sportsScore) {

        int total = sportsScore;

        for (int score : testScores) {

            total += score;

        }

        return total;

    }

    public int getTotalScore() {

        return totalScore;

    }

    public void displayResult() {

        System.out.println("Roll No: " + getRollNo());

        System.out.println("Total Score: " + totalScore);

    }

}

public class exp3\_7 {

    public static void main(String[] args) {

        // Example usage

        int[] testScores = {80, 85, 90};

        int sportsScore = 95;

        int rollNo = 2078;

        Result result = new Result(rollNo, testScores, sportsScore);

        result.displayResult();

    }

}

# **Output:**

Roll No: 2078

Total Score: 350

1. **Write a program to design a class representing theinformation regarding digital library (books, tape: book & tape should be separate classes having the base class as media).The class should have the functionality for adding new item, issuing, deposit etc.**

# **Program:**

import java.util.ArrayList;

class Media {

    private String title;

    private String author;

    private boolean available;

    public Media(String title, String author) {

        this.title = title;

        this.author = author;

        this.available = true; // Initially available

    }

    public String getTitle() {

        return title;

    }

    public String getAuthor() {

        return author;

    }

    public boolean isAvailable() {

        return available;

    }

    public void setAvailable(boolean available) {

        this.available = available;

    }

}

class Book extends Media {

    private int pages;

    public Book(String title, String author, int pages) {

        super(title, author);

        this.pages = pages;

    }

    public int getPages() {

        return pages;

    }

}

class Tape extends Media {

    private int duration; // Duration in minutes

    public Tape(String title, String author, int duration) {

        super(title, author);

        this.duration = duration;

    }

    public int getDuration() {

        return duration;

    }

}

public class exp3\_8 {

    public static void main(String[] args) {

        ArrayList<Media> library = new ArrayList<>();

        // Adding books and tapes to the library

        library.add(new Book("Can't Hurt Me", "David Goggins", 400));

        library.add(new Tape("Atomic Habits", "James Clear", 320));

        // Displaying available items in the library

        System.out.println("Available Items in the Library:");

        for (Media item : library) {

            if (item.isAvailable()) {

                if (item instanceof Book) {

                    Book book = (Book) item;

                    System.out.println("Book: " + book.getTitle() + " by " + book.getAuthor() + " (" + book.getPages() + " pages)");

                } else if (item instanceof Tape) {

                    Tape tape = (Tape) item;

                    System.out.println("Tape: " + tape.getTitle() + " by " + tape.getAuthor() + " (" + tape.getDuration() + " minutes)");

                }

            }

        }

        // Simulating issuing and depositing items

        Media book = library.get(0);

        book.setAvailable(false); // Book is issued

        System.out.println("Book '" + book.getTitle() + "' has been issued.");

        Media tape = library.get(1);

        tape.setAvailable(false); // Tape is issued

        System.out.println("Tape '" + tape.getTitle() + "' has been issued.");

        // Displaying updated availability

        System.out.println("\nUpdated Availability:");

        for (Media item : library) {

            System.out.println(item.getTitle() + ": " + (item.isAvailable() ? "Available" : "Not Available"));

        }

        // Simulating depositing

        book.setAvailable(true); // Book is deposited

        System.out.println("\nBook '" + book.getTitle() + "' has been deposited.");

        // Displaying updated availability

        System.out.println("\nUpdated Availability:");

        for (Media item : library) {

            System.out.println(item.getTitle() + ": " + (item.isAvailable() ? "Available" : "Not Available"));

        }

    }

}

# **Output:**

Available Items in the Library:

Book: Can't Hurt Me by David Goggins (400 pages)

Tape: Atomic Habits by James Clear (320 minutes)

Book 'Can't Hurt Me' has been issued.

Tape 'Atomic Habits' has been issued.

Updated Availability:

Can't Hurt Me: Not Available

Atomic Habits: Not Available

Book 'Can't Hurt Me' has been deposited.

Updated Availability:

Can't Hurt Me: Available

Atomic Habits: Not Available

1. **Create class student having member variable mark and name. there is a derived class named as markname which will take input name and marks of 10 student as a input from the user and display the name and mark of 10 student.**

# **Program:**

import java.util.Scanner;

class Student {

    int marks[];

    String name;

}

class Markname extends Student {

    public void take\_input() {

        Scanner scanner = new Scanner(System.in);

        marks = new int[2]; // Assuming 2 subjects

        for (int i = 0; i < 2; i++) {

            System.out.print("Enter name for student " + (i + 1) + ": ");

            name = scanner.nextLine();

            for (int j = 0; j < marks.length; j++) {

                System.out.print("Enter marks of subject " + (j + 1) + " for student " + (i + 1) + ": ");

                marks[j] = scanner.nextInt();

            }

            scanner.nextLine(); // Consume newline character

            displayMarks(); // Display marks after input for each student

        }

    }

    public void displayMarks() {

        // Displaying names and marks of the current student

        System.out.println("\nName\tSubject 1\tSubject 2");

        System.out.print(name + "\t");

        for (int mark : marks) {

            System.out.print(mark + "\t\t");

        }

        System.out.println();

    }

}

public class Exp3\_9 {

    public static void main(String args[]) {

        Markname markname = new Markname();

        markname.take\_input(); // Take input for names and marks

    }

}

# **Output:**

Enter name for student 1: Prathmesh

Enter marks of subject 1 for student 1: 89

Enter marks of subject 2 for student 1: 78

Name Subject 1 Subject 2

Prathmesh 89 78

Enter name for student 2: Junior

Enter marks of subject 1 for student 2: 09

Enter marks of subject 2 for student 2: 98

Name Subject 1 Subject 2

Junior 9 98

1. **Write a program that illustrates interface inheritance. Interface P12 inherits from both P1 and P2. Each interface declares one constant and one method. The class Q implements P12 . Instantiate Q and invoke each of its methods. Each method displays one of the constants.**

# **Program:**

interface P1 {

    int CONSTANT\_1 = 10; // Constant

    void methodP1(); // Method

}

// Interface P2

interface P2 {

    int CONSTANT\_2 = 20; // Constant

    void methodP2(); // Method

}

// Interface P12 inherits from P1 and P2

interface P12 extends P1, P2 {

}

// Class Q implements P12 interface

class Q implements P12 {

    // Implementing methodP1 from interface P1

    public void methodP1() {

        System.out.println("Constant from P1: " + CONSTANT\_1);

    }

    // Implementing methodP2 from interface P2

    public void methodP2() {

        System.out.println("Constant from P2: " + CONSTANT\_2);

    }

}

public class exp3\_10  {

    public static void main(String[] args) {

        // Instantiate Q

        Q q = new Q();

        // Invoke methods of Q

        q.methodP1();

        q.methodP2();

    }

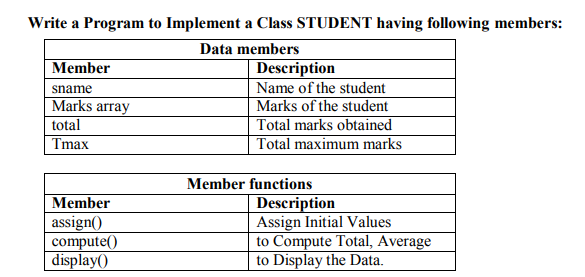
}

# **Output:**

Constant from P1: 10

Constant from P2: 20

**Q.11**



# **Program:**

import java.util.Scanner;

class Student {

    private String sname;

    private int[] marks;

    private int total;

    private int Tmax;

    // Constructor to initialize the data members

    public Student() {

        sname = "";

        marks = new int[5]; // Assuming 5 subjects

        total = 0;

        Tmax = 0;

    }

    // Method to assign initial values

    public void assign(String name, int[] studentMarks, int maxMarks) {

        sname = name;

        marks = studentMarks;

        Tmax = maxMarks;

    }

    // Method to compute total marks obtained and display data

    public void compute() {

        for (int mark : marks) {

            total += mark;

        }

        System.out.println("Student Name: " + sname);

        System.out.println("Marks:");

        for (int i = 0; i < marks.length; i++) {

            System.out.println("Subject " + (i + 1) + ": " + marks[i]);

        }

        System.out.println("Total Marks Obtained: " + total);

        System.out.println("Total Maximum Marks: " + Tmax);

        System.out.println("Average Marks: " + ((double) total / marks.length));

    }

}

public class  exp3\_11  {

    public static void main(String[] args) {

        // Create a Scanner object for input

        Scanner scanner = new Scanner(System.in);

        // Create an object of class Student

        Student student = new Student();

        // Input for student name

        System.out.print("Enter student name: ");

        String name = scanner.nextLine();

        // Input for marks of 5 subjects

        int[] marks = new int[5];

        for (int i = 0; i < 5; i++) {

            System.out.print("Enter marks for subject " + (i + 1) + ": ");

            marks[i] = scanner.nextInt();

        }

        // Input for total maximum marks

        System.out.print("Enter total maximum marks: ");

        int maxMarks = scanner.nextInt();

        // Call assign() method to assign initial values

        student.assign(name, marks, maxMarks);

        // Call compute() method to compute total marks obtained and display data

        student.compute();

    }

}

# **Output:**

Enter student name: Prathmesh

Enter marks for subject 1: 89

Enter marks for subject 2: 78

Enter marks for subject 3: 90

Enter marks for subject 4: 87

Enter marks for subject 5: 67

Enter total maximum marks: 90

Student Name: Prathmesh

Marks:

Subject 1: 89

Subject 2: 78

Subject 3: 90

Subject 4: 87

Subject 5: 67

Total Marks Obtained: 411

Total Maximum Marks: 90

Average Marks: 82.2