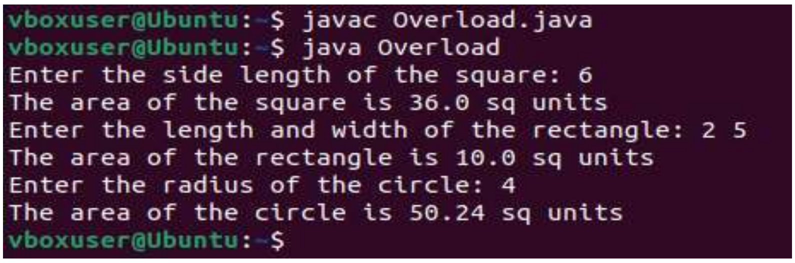
SOURCE CODE

```
import java.util.Scanner;
class OverloadDemo {
    void area(float x) {
        System.out.println("The area of the square is " + Math.pow(x, 2) + " sq units");    }
    void area(float x, float y) {
        System.out.println("The area of the rectangle is " + x * y + " sq units");    }
    void area(double x) {
        double z = 3.14 * x * x;
        System.out.println("The area of the circle is " + z + " sq units");    }   }
class Overload {
    public static void main(String args[]) {
        OverloadDemo ob = new OverloadDemo();
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the side length of the square: ");
        float squareSide = scanner.nextFloat();
        ob.area(squareSide);
        System.out.print("Enter the length and width of the rectangle: ");
        float rectangleLength = scanner.nextFloat();
        float rectangleWidth = scanner.nextFloat();
        ob.area(rectangleLength, rectangleWidth);
        System.out.print("Enter the radius of the circle: ");
        double circleRadius = scanner.nextDouble();
        ob.area(circleRadius);
        scanner.close();    }}
```

OUTPUT



```
vboxuser@Ubuntu:~$ javac Overload.java
vboxuser@Ubuntu:~$ java Overload
Enter the side length of the square: 6
The area of the square is 36.0 sq units
Enter the length and width of the rectangle: 2 5
The area of the rectangle is 10.0 sq units
Enter the radius of the circle: 4
The area of the circle is 50.24 sq units
vboxuser@Ubuntu:~$
```

Lab cycle:3 Date:17/06/2023

Experiment no:16

AREA OF SHAPES USING METHOD OVERLOADING

AIM: Using the concept of method overloading find area of different shapes Rectangle, Circle and Square.

ALGORITHM:

Step 1: Define a class named OverloadDemo with three overloaded area methods that calculate and print the areas of a square, rectangle, and circle.

Step 2: In the main method of the Overload class, create an instance of OverloadDemo named ob and a Scanner object named scanner.

Step 3: Read the side length of a square from the user and call ob.area(squareSide) to calculate and display the area.

Step 4: Read the length and width of a rectangle from the user and call ob.area(rectangleLength, rectangleWidth) to calculate and display the area.

Step 5: Read the radius of a circle from the user and call ob.area(circleRadius) to calculate and display the area.

Step 6: Close the Scanner object scanner.

RESULT: The program has been executed successfully and output obtained.

SOURCE CODE

```
import java.util.Scanner;
class Employee {
    int empid;
    String name;
    double salary;
    String address;
    Employee(int no, String na, double sal, String add) {
        this.empid = no;
        this.name = na;
        this.salary = sal;
        this.address = add;    }  }
public class Teacher extends Employee {
    String department;
    String subject;
    Teacher(int no, String na, double sal, String add, String dep, String sub) {
        super(no, na, sal, add);
        this.department = dep;
        this.subject = sub;    }
    void display() {
        System.out.println("Employee ID: " + empid);
        System.out.println("Name: " + name);
        System.out.println("Salary: " + salary);
        System.out.println("Address: " + address);
        System.out.println("Department: " + department);
        System.out.println("Subject: " + subject);       }
    public static void main(String[] args) {
        System.out.print("Enter the number of employees: ");
        Scanner sc1 = new Scanner(System.in);
        int num = sc1.nextInt();        Teacher[] arr = new Teacher[num];
            for (int i = 0; i < num; i++) {
          Scanner sc = new Scanner(System.in);
```

Lab cycle:3 Date:17/06/2023

Experiment no:17

EMPLOYEE DETAILS USING SINGLE INHERITANCE

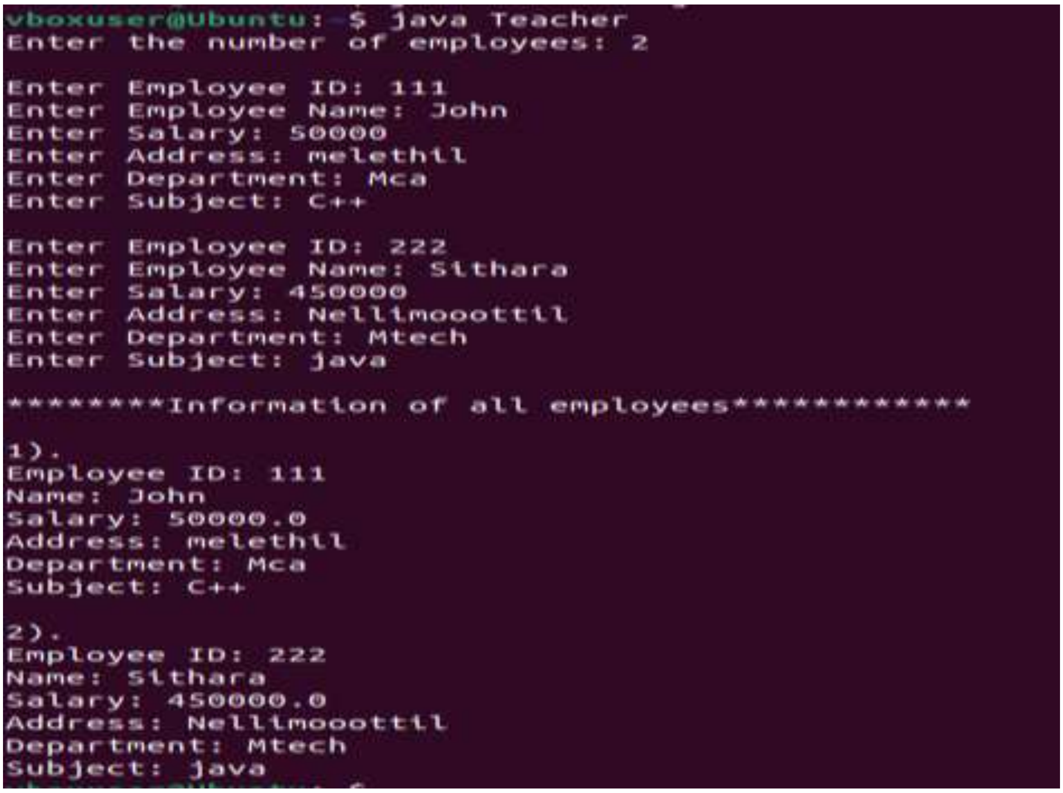
AIM: Create a class employee with data members empId, name, salary, address and constructors to initialize the data members. create another class teacher that inherit the properties of class employee and contain its on data members department, subjects thought and constructors to initialize each data members and also include display function to display all the data members, use array of objects to display details of n teachers.

ALGORITHM:

```
Step 1: Prompt the user to enter the number of employees.

Step 2: Create an empty array of Teacher objects with a size equal to the number of employees entered.

Step 3: Use a loop to iterate through each position in the array.
        3.1 Prompt the user to enter the details of the employee (employee ID, name, salary, address, department, and subject).
        3.2. Create a new Teacher object with the entered details.
        3.3. Assign the newly created Teacher object to the current position in the array.

Step 4: Display a header indicating the information of all employees.

Step 5: Use a loop to iterate through each position in the array.
        5.1. Display the index of the employee.
        5.2 Call the display() method for the Teacher object at the current position in the array, which will print the employee details.

Step 6: Close the scanner used for input.

Step 7: End the program.
```

```
    Teacher[] arr = new Teacher[num];
    for (int i = 0; i < num; i++) {          Scanner sc = new Scanner(System.in);
    System.out.print("\nEnter Employee ID: ");
    int empid = sc.nextInt();
    System.out.print("Enter Employee Name: ");
    String name = sc.next();
    System.out.print("Enter Salary: ");
    double salary = sc.nextDouble();
    System.out.print("Enter Address: ");
    String address = sc.next();
    System.out.print("Enter Department: ");
    String department = sc.next();
    System.out.print("Enter Subject: ");
    String subject = sc.next();
    arr[i] = new Teacher(empid, name, salary, address, department, subject);        }
  System.out.println("\n********Information of all employees************");
  for (int i = 0; i < num; i++) {        int j = i + 1;
    System.out.println("\n" + j + ").");        arr[i].display();        }     sc1.close();                }  }
```

OUTPUT



```
vboxuser@Ubuntu:~$ java Teacher
Enter the number of employees: 2

Enter Employee ID: 111
Enter Employee Name: John
Enter Salary: 50000
Enter Address: melethil
Enter Department: Mca
Enter Subject: C++

Enter Employee ID: 222
Enter Employee Name: Sithara
Enter Salary: 450000
Enter Address: Nellimooottil
Enter Department: Mtech
Enter Subject: java

********Information of all employees************

1).
Employee ID: 111
Name: John
Salary: 50000.0
Address: melethil
Department: Mca
Subject: C++

2).
Employee ID: 222
Name: Sithara
Salary: 450000.0
Address: Nellimooottil
Department: Mtech
Subject: java
```

RESULT: The program has been executed successfully and output obtained.

SOURCE CODE

```
import java.util.Scanner;
class Person    {
            Scanner in=new Scanner(System.in);
            String name,gender,address;
            int age;
            Person()            {
                        System.out.println("\nEnter name");
                        name=in.nextLine();
                        System.out.println("Enter gender if male M if female F others O");
                        gender=in.nextLine();
                        System.out.println("Enter address");
                        address=in.nextLine();
                        System.out.println("Enter age");
                        age=in.nextInt();            }
            void display()    {
                         System.out.println("\n********Information of all Persons************");
                        System.out.println("Name:"+name);
                        System.out.println("Gender:"+gender);
                        System.out.println("Address:"+address);
                        System.out.println("Age:"+age);}     }
class Employee extends Person   {
            int empid;
            float salary;s
            String cname,quali;
            Employee()       {
                        System.out.println("\nEnter Employee id");
                        empid=in.nextInt();
                        System.out.println("Enter Company Name");
                        cname=in.next();
                        System.out.println("Enter Education Qualification");
                        in.skip("[\r\n]+");
```

Lab cycle:3 Date: 19/06/2023
Experiment no:18

PERSON DETAILS USING MULTI LEVEL INHERITANCE

AIM: Create a class person with data members name, gender, address, age and constructor to initialize the data members and another class Employee that inherits the properties of class person and also contains its own data members like empid, company name, qualification, salary and its own constructor. Create another class Teacher that inherits the properties of class employee and contains its own data members like subject, department, teacherId and also contain constructors and methods to display the data members. Use array of objects to display details of N Teachers.

ALGORITHM:

Step 1: Define three classes: Person, Employee, and Teacher.

Step 2: The Person class has attributes for name, gender, address, and age. It has a constructor to prompt the user for these details and a display() method to print the information.

Step 3: The Employee class extends Person and adds attributes for employee ID, company name, education qualification, and salary. It has its own constructor to prompt for these details and overrides the display() method to print all the information.

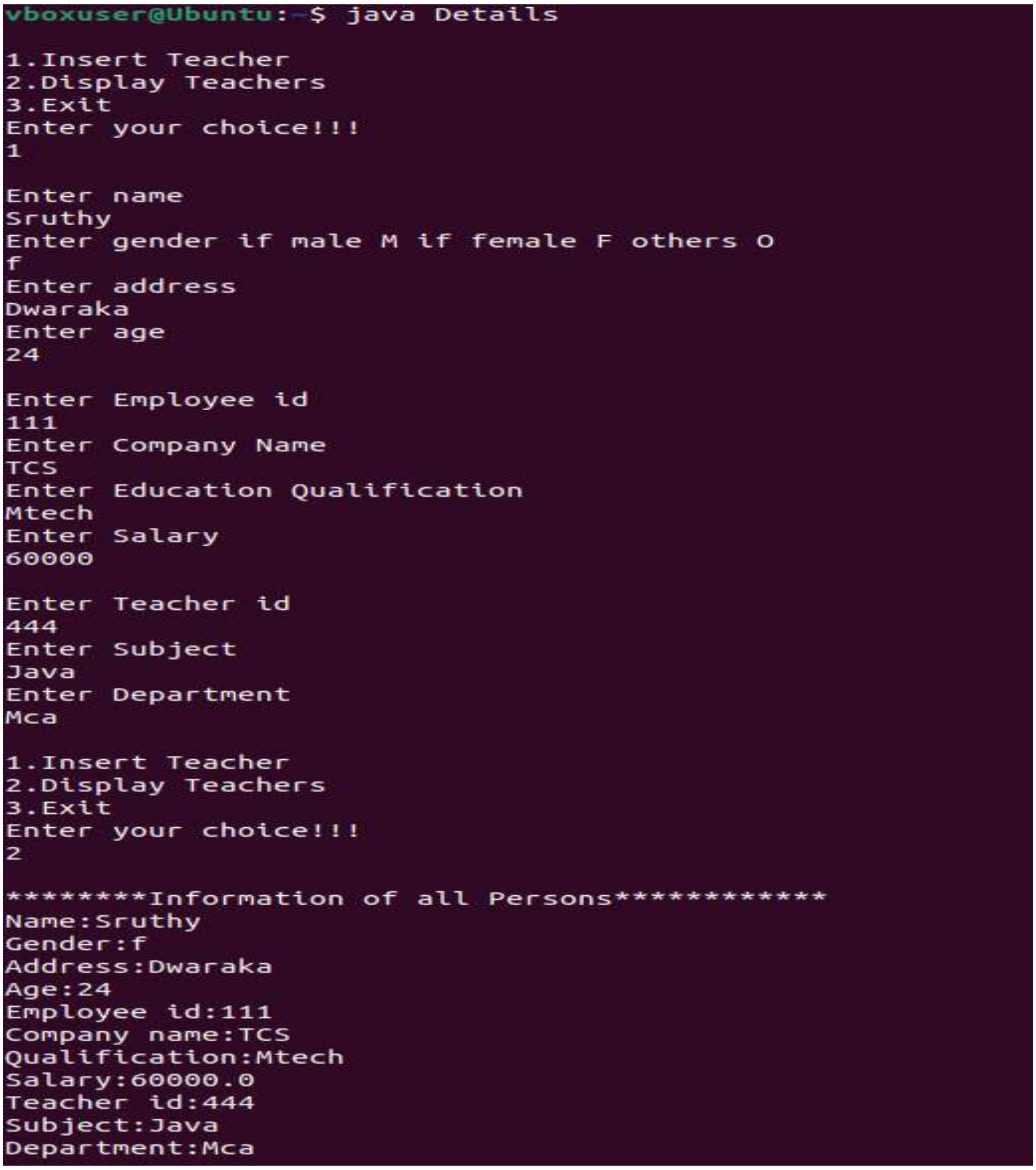
Step 4: The Teacher class extends Employee and adds attributes for teacher ID, subject, and department. It has its own constructor and overrides the display() method to print all the information.

Step 5: In the main class Details, create an array to store Teacher objects.

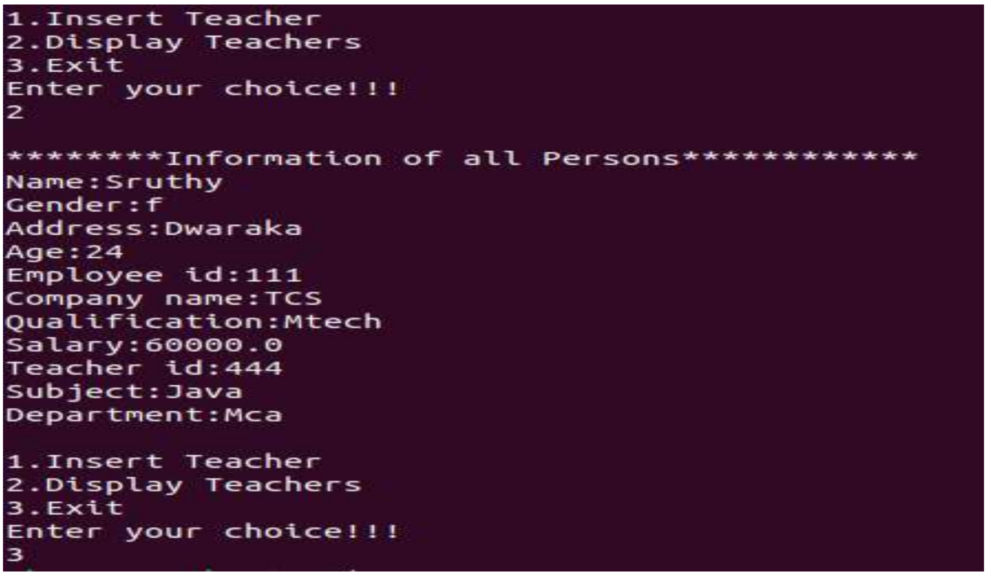
Step 6: Continue the loop until the user chooses to exit the program.

```
        quali=in.next();     System.out.println("Enter Salary");   salary=in.nextFloat();       }
        void display()    {
        super.display();
        System.out.println("Employee id:"+empid+"\nCompany name:"+cname+
        “\nQualification:"+quali+"\nSalary:"+salary);     }           }
class Teacher extends Employee{
    int tid;    String sub,dept;
    Teacher()          {
        System.out.println("\nEnter Teacher id");
        tid=in.nextInt();
        System.out.println("Enter Subject");
        sub=in.next();
        System.out.println("Enter Department");
        in.skip("[\r\n]+");    dept=in.next();                     }
    void display()                {
        super.display();
        System.out.println("Teacher id:"+tid+"\nSubject:"+sub+"\nDepartment:"+dept);   }}
public class Details{
    public static void main(String args[])                  {
        int i=-1;
        Teacher[] t=new Teacher[10];
        Scanner in=new Scanner(System.in);
        while(true){
        System.out.println("\n1.Insert Teacher\n2.Display Teachers\n3.Exit\n Enter your
        choice");
        int choice = in.nextInt();
            switch(choice)              {
                case 1:t[++i]=new Teacher();     break;
                case 2:for(int j=0;j<=i;j++)
                        t[j].display();     break;
                case 3:System.exit(1);           break;
                default:System.out.println("Invalid Choice");      }  }   }   }
```

OUTPUT



```
vboxuser@Ubuntu:~$ java Details

1.Insert Teacher
2.Display Teachers
3.Exit
Enter your choice!!!
1

Enter name
Sruthy
Enter gender if male M if female F others O
f
Enter address
Dwaraka
Enter age
24

Enter Employee id
111
Enter Company Name
TCS
Enter Education Qualification
Mtech
Enter Salary
60000

Enter Teacher id
444
Enter Subject
Java
Enter Department
Mca

1.Insert Teacher
2.Display Teachers
3.Exit
Enter your choice!!!
2

********Information of all Persons************
Name:Sruthy
Gender:f
Address:Dwaraka
Age:24
Employee id:111
Company name:TCS
Qualification:Mtech
Salary:60000.0
Teacher id:444
Subject:Java
Department:Mca
```



```
1.Insert Teacher
2.Display Teachers
3.Exit
Enter your choice!!!
2

********Information of all Persons************
Name:Sruthy
Gender:f
Address:Dwaraka
Age:24
Employee id:111
Company name:TCS
Qualification:Mtech
Salary:60000.0
Teacher id:444
Subject:Java
Department:Mca

1.Insert Teacher
2.Display Teachers
3.Exit
Enter your choice!!!
3
```

RESULT: The program has been executed successfully and output obtained.

SOURCE CODE

```
import java.util.Scanner;
class Publisher{
   String publisher;
   Publisher(String pub){
      this.publisher=pub;    }  }
class Book extends Publisher{
   String book;
   Book(String pub,String boo){
      super(pub);
      book=boo;    }  }
class Literature extends Book{
   String category;
   Literature(String pub, String boo){
      super(pub, boo);    }
   void display(){
      System.out.println("Publisher :"+publisher);
      System.out.println("Book :"+book);    }  }
class Fiction extends Book{
   Fiction(String pub, String boo){
      super(pub, boo);    }
   void display(){
      System.out.println("Publisher :"+publisher);
      System.out.println("Book :"+book);    }}
public class bookDetails{
   public static void main(String[] args) {
      System.out.println("\nEnter the No. of Literature Books");
      Scanner sc1 = new Scanner(System.in);
      int num = sc1.nextInt();
      Literature arr[]=new Literature[num];
```

Lab cycle:3 Date: 19/06/2023

Experiment no:19

BOOK DETAILS USING HYBRID INHERITANCE

AIM: Write a program class Publisher, Book, Literature and Fiction. Read the information and print the details of books either category using inheritance.

ALGORITHM:

Step 1: Start

Step 2: Ask the user for the number of literature books.

Step 3: Create an array to store the literature book details.

Step 4: Prompt the user to enter the details for each literature book (book name and publisher).

Step 5: Create a new Literature object with the provided details and add it to the array.

Step 6: Repeat steps 3-4 for the specified number of literature books.

Step 7: Ask the user for the number of fiction books.

Step 8: Create an array to store the fiction book details.

Step 9: Prompt the user to enter the details for each fiction book (book name and publisher).

Step 10: Create a new Fiction object with the provided details and add it to the array.

Step 11: Repeat steps 8-9 for the specified number of fiction books.

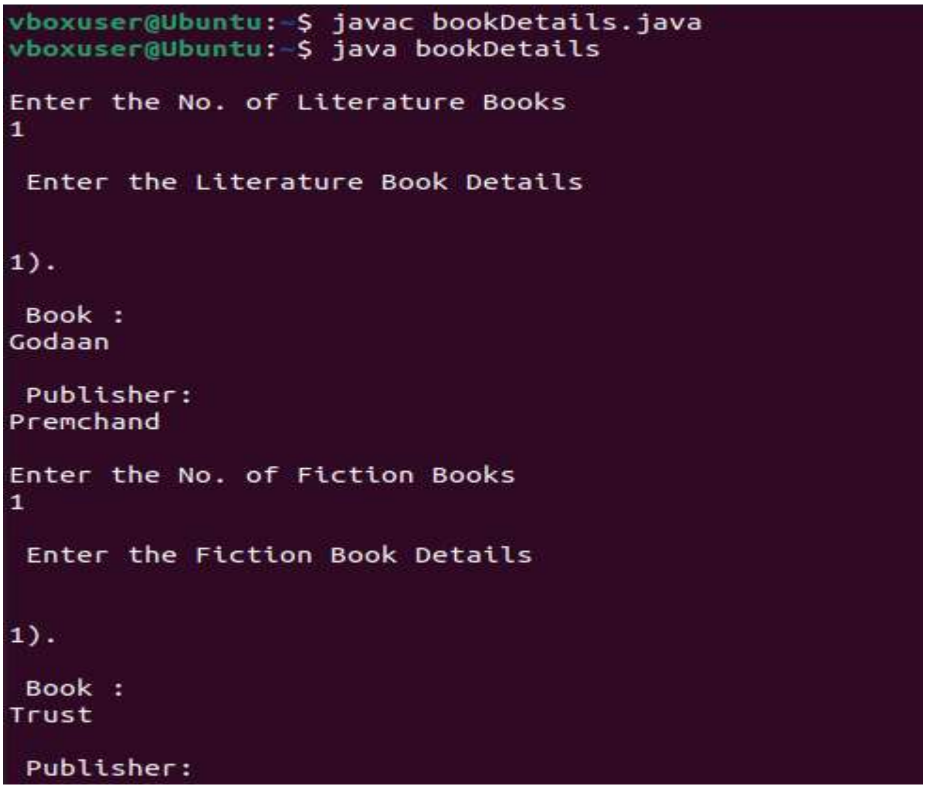
Step 12: Display the information of all the literature books by iterating over the literature book array and calling the display() method for each object.

Step 13: Display the information of all the fiction books by iterating over the fiction book array and calling the display() method for each object.

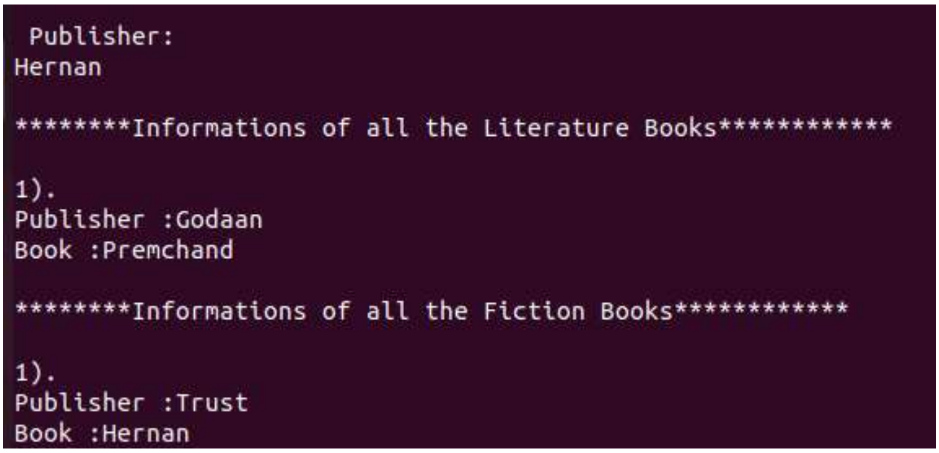
```
System.out.println("\n Enter the Literature Book Details\n");
        int x = 0,j=0;
        Scanner sc =new Scanner(System.in);
        for(int i =0;i<num;i++)        {
            x = i +1;
            System.out.println("\n"+x+").");
            System.out.println("\n Book : ");
            String boo =sc.next();
            System.out.println("\n Publisher: ");
            String pub =sc.next();
            arr[i]=new Literature(boo,pub);    }
        System.out.println("\nEnter the No. of Fiction Books");
        int num1 = sc1.nextInt();
        Fiction arr1[]=new Fiction[num1];
        System.out.println("\n Enter the Fiction Book Details\n");
        int x1 = 0,j1=0;
        for(int i =0;i<num1;i++)    {
            x1 = i +1;
            System.out.println("\n"+x1+").");
            System.out.println("\n Book : ");
            String boo =sc.next();
            System.out.println("\n Publisher: ");
            String pub =sc.next();
            arr1[i]=new Fiction(boo,pub);     }
        sc.close();
        sc1.close();
        System.out.println("\n********Informations of all the Literature
Books************");
        for(int i=0;i<num;i++){
            j=i+1;
            System.out.println("\n"+j+").");
```

```
        arr[i].display();    }
     System.out.println("\n********Informations of all the Fiction Books************");
     for(int i=0;i<num1;i++){
        j1=i+1;
        System.out.println("\n"+j1+").");
        arr1[i].display();       }
   sc1.close();    }    }
```

OUTPUT



```
vboxuser@Ubuntu:~$ javac bookDetails.java
vboxuser@Ubuntu:~$ java bookDetails

Enter the No. of Literature Books
1

 Enter the Literature Book Details


1).

 Book :
Godaan

 Publisher:
Premchand

Enter the No. of Fiction Books
1

 Enter the Fiction Book Details


1).

 Book :
Trust

 Publisher:
```



```
 Publisher:
Hernan

********Informations of all the Literature Books************

1).
Publisher :Godaan
Book :Premchand

********Informations of all the Fiction Books************

1).
Publisher :Trust
Book :Hernan
```

RESULT: The program has been executed successfully and output obtained.

SOURCE CODE

```
import java.util.Scanner;
class sports{
    String sport;
    int score;
    sports(String spo, int sc){
        sport = spo;
        score = sc;    }   }
class student extends sports{
    String Grade;
    double Overall_per;
    student(String spo, int sc,String gd, double per ){
        super(spo, sc);
        Grade = gd;
        Overall_per = per;    }  }
public class result extends student {
    result(String spo, int sc,String gd, double per ){
        super(spo, sc, gd, per);    }
    void display(){
        System.out.println("\nSports Details of Student");
        System.out.println("Name of Sports :"+sport);
        System.out.println("Sports score :"+score);
        System.out.println("\nAcademic Details of Student");
        System.out.println("Academic Grade :"+Grade);
        System.out.println("Overall percentage :"+Overall_per);      }
    public static void main(String[] args) {
        Scanner sc =new Scanner(System.in);
        System.out.println("\nEnter the Sports Details of Student");
        System.out.println("\n Name of Sports: ");
        String a =sc.next();
```

Lab cycle:3 Date: 19/06/2023

Experiment no:20

SPORTS DETAILS USING MULTI LEVEL INHERITANCE

AIM: Create class Student and sports. Create another class Result inherited from Student and sports. Display the academic and sports score of a student.

ALGORITHM:

Step 1: Import the Scanner class for user input.

Step 2: Define a class "sports" with sport name and score variables, and a constructor to initialize them.

Step 3: Define a class "student" extending "sports" with additional grade and overall percentage variables, and a constructor to initialize them.

Step 4: Define a class "result" extending "student" with a constructor similar to "student".

Step 5: Add a "display" method in "result" to print sports and academic details.

Step 6: In the "main" method, create a Scanner object for user input.

Step 7: Prompt and read the sport name and score.

Step 8: Prompt and read the grade and overall percentage.

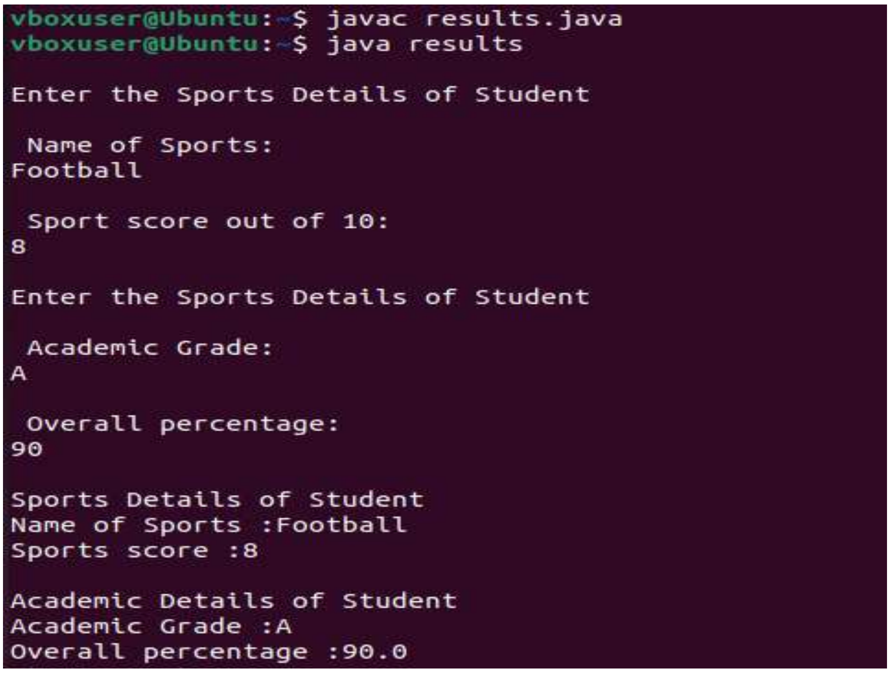
Step 9: Close the Scanner object.

Step 10: Create an instance of "result" using the input values.

Step 11: Call the display method to print the details.

```
System.out.println("\n Sport score out of 10: ");
int b =sc.nextInt();
System.out.println("\nEnter the Sports Details of Student");
System.out.println("\n Academic Grade: ");
String c =sc.next();
System.out.println("\n Overall percentage: ");
double d =sc.nextDouble();
sc.close();
result obj= new result(a,b,c,d);
obj.display();    }  }
```

OUTPUT



```
vboxuser@Ubuntu:~$ javac results.java
vboxuser@Ubuntu:~$ java results

Enter the Sports Details of Student

 Name of Sports:
Football

 Sport score out of 10:
8

Enter the Sports Details of Student

 Academic Grade:
A

 Overall percentage:
90

Sports Details of Student
Name of Sports :Football
Sports score :8

Academic Details of Student
Academic Grade :A
Overall percentage :90.0
```

RESULT: The program has been executed successfully and output obtained.

SOURCE CODE

```
import java.util.Scanner;
interface prop{
   void getdata();
   void area();
   void perimeter();     }
class Circle implements prop    {
   double pi = 3.14;
   double r;
   Scanner sc = new Scanner(System.in);
   public void getdata()    {
      System.out.println("Enter the radius of the circle:");
      r = sc.nextDouble();    }
   public void perimeter()    {
      System.out.println("Area of the circle: "+(2*pi*r));    }
   public void area()    {
      System.out.println("Perimeter of the circle: "+(pi*r*r));    }  }
class Rectangle implements prop   {
   double l,b;
   Scanner sc = new Scanner(System.in);
   public void getdata()    {
      System.out.println("Enter the length of the rectangle:");
      l = sc.nextDouble();
      System.out.println("Enter the breadth of the rectangle:");
      b = sc.nextDouble();    }
   public void area()      {
      System.out.println("Area of a rectangle: "+(l*b));    }
   public void perimeter()   {
      System.out.println("Perimeter of a rectangle: "+(2*(l+b)));   }  }
public class object    {
```

Lab cycle:3 Date: 19/06/2023
Experiment no:21

AREA AND PERIMETER OF SHAPES

AIM: Create an inheritance having prototypes of functions Area() and Perimeter(). Create two classes Circle and Rectangle which implements the above interface. Create a menu driven program to find Area and Perimeter of objects.

ALGORITHM:

Step 1: Define an interface "prop" with "getdata," "area," and "perimeter" methods.

Step 2: Create a "Circle" class implementing the "prop" interface.

- 2.1. Declare variables for pi and radius.
- 2.2.Implement "getdata" to read the radius.
- 2.3.Implement "perimeter" to calculate and print the circle's perimeter.
- 2.4.Implement "area" to calculate and print the circle's area.

Step 3: Create a "Rectangle" class implementing the "prop" interface.

- 3.1.Declare variables for length and breadth.
- 3.2.Implement "getdata" to read the length and breadth.
- 3.3.Implement "area" to calculate and print the rectangle's area.
- 3.4.Implement "perimeter" to calculate and print the rectangle's perimeter.

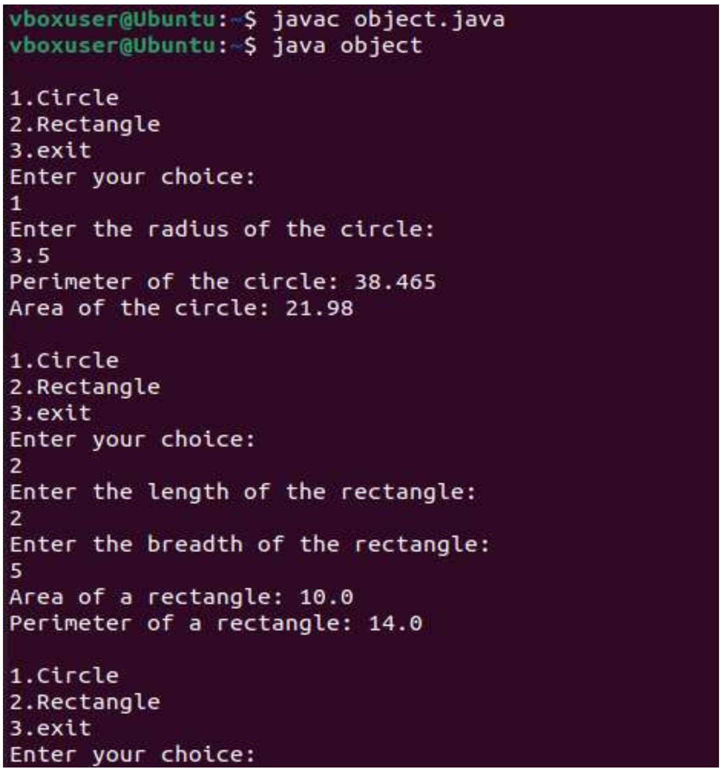
Step 4: In the "object" class:

- 4.1.Declare variables for choice and create objects of "Circle" and "Rectangle."
- 4.2.Enter a loop to repeatedly prompt the user for shape choice.
- 4.3.Inside the loop, display shape options and read the user's choice.
- 4.4.Use a switch statement to perform actions based on the choice:
    - 4.1.1. Case 1: Call methods of the "Circle" object to get data, calculate, and display area and perimeter.
    - 4.1.2. Case 2: Call methods of the "Rectangle" object to get data, calculate, and display area and perimeter.
    - 4.1.3. Case 3: Exit the program.

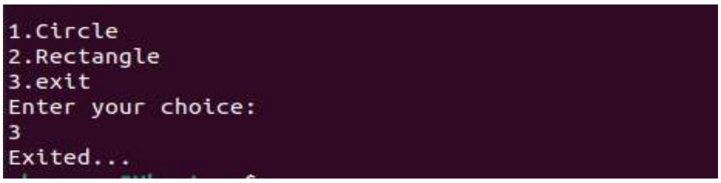
Step 5: Repeat the loop until the program is manually exited.

```
    public static void main(String[] args)    {
        int ch;
        Scanner sc = new Scanner(System.in);
        Circle ob = new Circle();
        Rectangle obj = new Rectangle();
        do       {
            System.out.println("\n1.Circle\n2.Rectangle\n3.exit");
            System.out.println("Enter your choice:");
            ch = sc.nextInt();
            switch(ch)      {
                case 1 :ob.getdata();
                        ob.area();
                        ob.perimeter();
                        break;
                case 2 :obj.getdata();
                        obj.area();
                        obj.perimeter();
                        break;
                case 3 :System.out.println("Exited...");
                        System.exit(0);           }       }
while(true);    }   }
```

OUTPUT



```
vboxuser@Ubuntu:~$ javac object.java
vboxuser@Ubuntu:~$ java object

1.Circle
2.Rectangle
3.exit
Enter your choice:
1
Enter the radius of the circle:
3.5
Perimeter of the circle: 38.465
Area of the circle: 21.98

1.Circle
2.Rectangle
3.exit
Enter your choice:
2
Enter the length of the rectangle:
2
Enter the breadth of the rectangle:
5
Area of a rectangle: 10.0
Perimeter of a rectangle: 14.0

1.Circle
2.Rectangle
3.exit
Enter your choice:
```



```
1.Circle
2.Rectangle
3.exit
Enter your choice:
3
Exited...
```

RESULT: The program has been executed successfully and output obtained

SOURCE CODE

```
import java.util.Scanner;
interface calc  {
   void calculate();    }
class bill implements calc   {
   String date,name,p_id,order_id;
   int quantity;
   double unit_price,total,namount=0;
   Scanner sc = new Scanner(System.in);
   public void getdata()  {
      System.out.println("\nEnter order ID:");
      order_id = sc.nextLine();
      System.out.println("\nEnter product id:");
      p_id = sc.nextLine();
      System.out.println("Enter product name:");
      name = sc.nextLine();
      System.out.println("Enter the Quantity:");
      quantity = sc.nextInt();
      System.out.println("Enter the unit price:");
      unit_price = sc.nextDouble();    }
   public void calculate()   {
      total = quantity * unit_price;   }
   public void display()    {
      System.out.println(p_id+"\t\t"+name+"\t\t"+quantity+"\t\t"+unit_price+"\t"+total);    }  }
public class calculator{
   public static void main(String[] args)    {
      int n,i;
      double namount=0;
      String date;
      Scanner sc = new Scanner(System.in);
```

Lab cycle:3 Date: 26/06/2023

Experiment no:22

PRODUCT BILL USING INTERFACE

AIM: Prepare Bill with the given format using and calculate method from interface.

Product Id	Name	Quantity	Unit price	Total
			Net amount:	

<u>ALGORITHM</u>:

Step 1: Import the necessary classes, including Scanner.

Step 2:Define an interface calc with a method calculate().

Step 3: Create a class bill that implements the calc interface.

Step 4: Inside the bill class, declare variables for order details and create methods to get input, calculate the total, and display the details.

Step 5: Create a class calculator with the main method.

Step 6: Prompt the user for order ID, date, and the number of products.

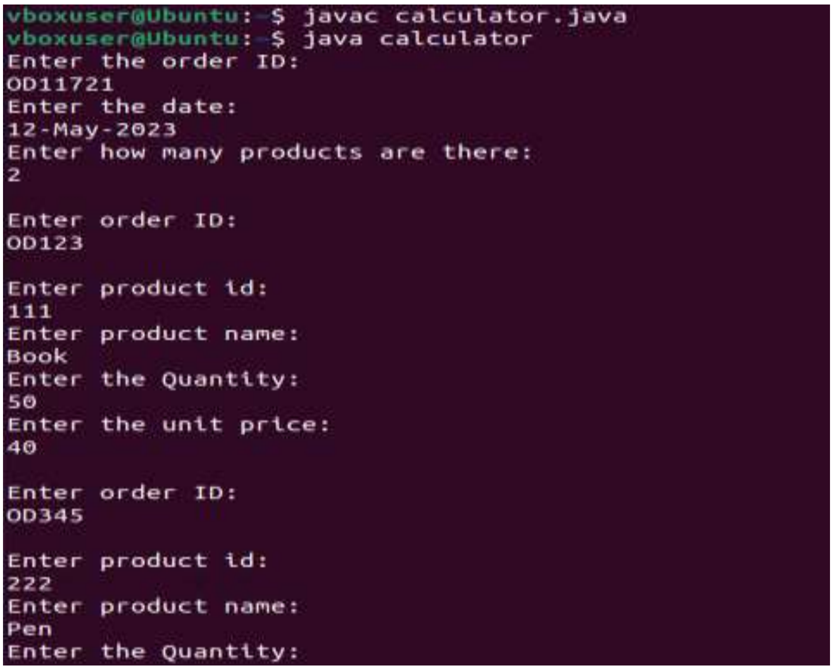
Step 7: Create an array of bill objects to store the product information.

Step 8: Iterate over the array to get input and calculate the total for each product.

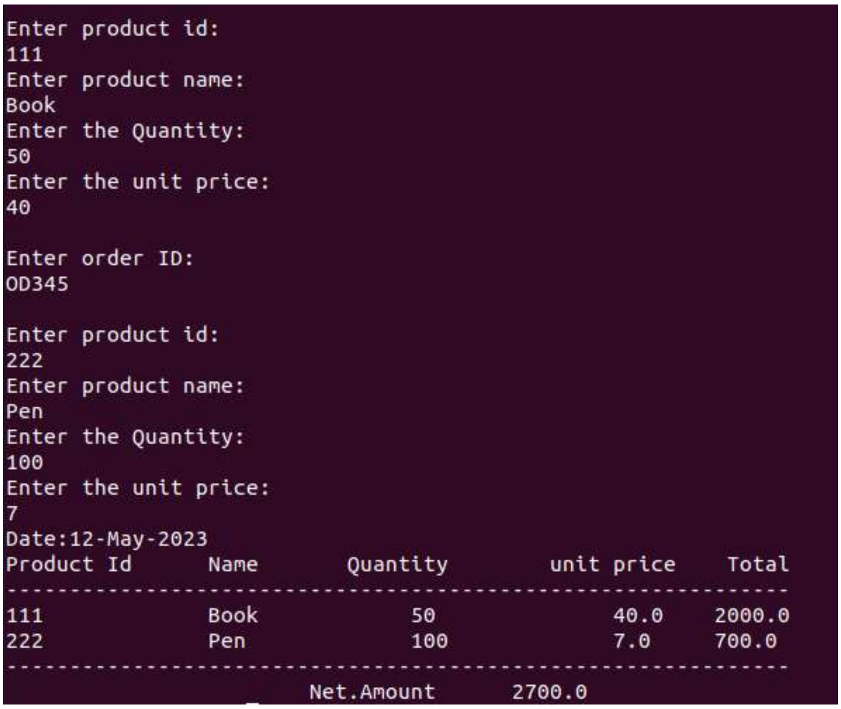
Step 9: Display the bill details, including the net amount.

```
System.out.println("Enter the order ID:");
String order_id = sc.nextLine();
System.out.println("Enter the date:");
date = sc.nextLine();
System.out.println("Enter how many products are there:");
n = sc.nextInt();
bill ob[] = new bill[n];
for(i=0;i<n;i++)
   ob[i] = new bill();
for(i=0;i<n;i++){
   ob[i].getdata();
   ob[i].calculate();      }
System.out.println("Date:"+date);
System.out.println("Product Id \tName\t   Quantity\t   unit price\t Total ");
System.out.println("------------------------------------------------------------");
for(i=0;i<n;i++){
   ob[i].display();
   namount += ob[i].total;        }
System.out.println("------------------------------------------------------------");
System.out.println("\t\t\tNet.Amount\t"+ namount);    }   }
```

OUTPUT



```
vboxuser@Ubuntu:~$ javac calculator.java
vboxuser@Ubuntu:~$ java calculator
Enter the order ID:
OD11721
Enter the date:
12-May-2023
Enter how many products are there:
2

Enter order ID:
OD123

Enter product id:
111
Enter product name:
Book
Enter the Quantity:
50
Enter the unit price:
40

Enter order ID:
OD345

Enter product id:
222
Enter product name:
Pen
Enter the Quantity:
```



```
Enter product id:
111
Enter product name:
Book
Enter the Quantity:
50
Enter the unit price:
40

Enter order ID:
OD345

Enter product id:
222
Enter product name:
Pen
Enter the Quantity:
100
Enter the unit price:
7
Date:12-May-2023
Product Id       Name        Quantity         unit price    Total
-------------------------------------------------------------
111              Book             50               40.0    2000.0
222              Pen             100                7.0     700.0
-------------------------------------------------------------
                        Net.Amount      2700.0
```

RESULT: The program has been executed successfully and output obtained.

SOURCE CODE

```
import java.util.Scanner;
public class methodOverriding {
    static double l, b, r;
    Scanner scanner = new Scanner(System.in);
    public static void main(String[] args) {
        System.out.println("Enter the length and breadth of the Rectangle: ");
        Rectangle rectangle = new Rectangle(scanner.nextDouble(), scanner.nextDouble());
        System.out.println(String.format("Rectangle area: %,.2f", rectangle.calculateArea()));
        System.out.println("Enter the radius of the circle");
        Circle circle = new Circle(scanner.nextDouble());
        System.out.println(String.format("Circle area: %,.2f", circle.calculateArea()));
        System.out.println("Enter the length of one side of the sqaure");
        Square square = new Square(scanner.nextDouble());
        System.out.println(String.format("Square area: %,.2f", square.calculateArea()));     }
    private static void out(String msg) {
        System.out.println(msg);    }  }
class Shape {
    public double calculateArea() {
        return 0;   }  }
class Rectangle extends Shape {
    private double length;
    private double width;
    public Rectangle(double length, double width) {
        this.length = length;
        this.width = width;    }
    public double calculateArea() {
        return length * width;    }}
class Circle extends Shape {
    private double radius;
```

Lab cycle:3 Date: 26/06/2023

Experiment no:23

AREA OF SHAPES USING METHOD OVERRIDING

AIM: Using the concept of method overriding find the area of shapes- Rectangle, Circle and Square.

ALGORITHM:

Step 1: Define the Shape class as the base class for all shapes. It has a method calculateArea() that returns 0 (since the base class doesn't have a specific shape).

Step 2: Create the Rectangle class, which extends the Shape class. It has length and width attributes and overrides the calculateArea() method to calculate and return the area of a rectangle using the formula length * width.

Step 3: Create the Circle class, which also extends the Shape class. It has a radius attribute and overrides the calculateArea() method to calculate and return the area of a circle using the formula π * radius^2.

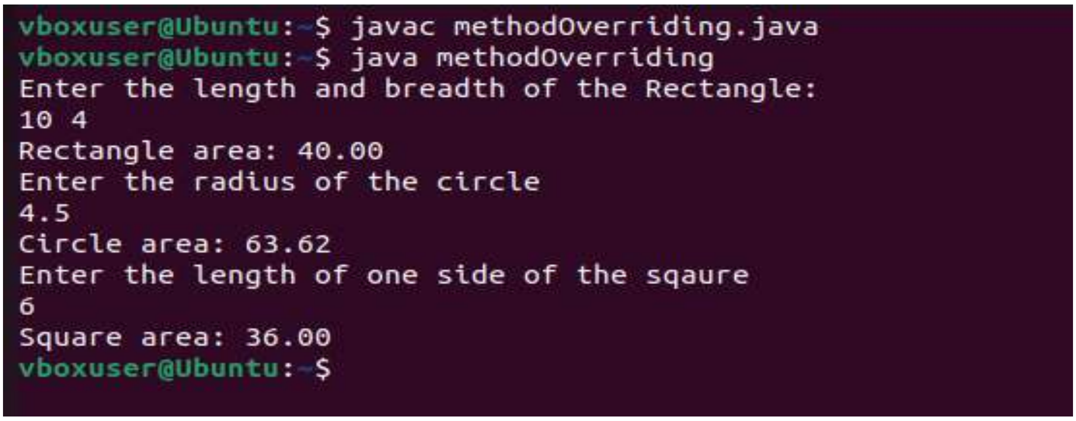
Step 4: Create the Square class, which extends the Rectangle class. It takes a single side length parameter and passes it to the Rectangle constructor by calling super(side, side).

Step 5: In the main method, create instances of Rectangle, Circle, and Square classes by taking user input for the required dimensions.

Step 6: Use the created objects to calculate and display the areas of each shape.

```
    public Circle(double radius) {
        this.radius = radius;    }
    public double calculateArea() {
        return Math.PI * Math.pow(radius, 2);    }  }
class Square extends Rectangle {
    public Square(double side) {
        super(side, side);    }  }
```

OUTPUT



```
vboxuser@Ubuntu:~$ javac methodOverriding.java
vboxuser@Ubuntu:~$ java methodOverriding
Enter the length and breadth of the Rectangle:
10 4
Rectangle area: 40.00
Enter the radius of the circle
4.5
Circle area: 63.62
Enter the length of one side of the sqaure
6
Square area: 36.00
vboxuser@Ubuntu:~$
```

RESULT: The program has been executed successfully and output obtained.

SOURCE CODE

```
import java.util.Scanner;
public class abstractShapes {
    static double l, b, r;
    Scanner scanner = new Scanner(System.in);
    public static void main(String[] args) {
        System.out.println("Enter the length and breadth of the Rectangle: ");
        Rectangle rectangle = new Rectangle(scanner.nextDouble(), scanner.nextDouble());
        System.out.println(String.format("Rectangle area: %,.2f", rectangle.findArea()));
       System.out.println("Enter the radius of the circle");
        Circle circle = new Circle(scanner.nextDouble());
        System.out.println(String.format("circle area: %,.2f", circle.findArea()));
        System.out.println("Enter the length of one side of the sqaure");
        Square square = new Square(scanner.nextDouble());
        System.out.println(String.format("Square area: %,.2f", square.findArea()));     }
    private static void out(String msg) {
        System.out.println(msg);    }   }
abstract class Shape {
    abstract double findArea();       }
class Rectangle extends Shape {
    private double length;
    private double width;
    public Rectangle(double length, double width) {
        this.length = length;
        this.width = width;     }
    double findArea() {
        return length * width;    }   }
class Circle extends Shape {
    private double radius;
    public Circle(double radius) {
```

Lab cycle:3 Date: 26/06/2023

Experiment no:24

AREA OF SHAPES USING ABSTRACT CLASS

AIM: Create an abstract class shape with an abstract method to find area of different shapes. Create subclasses Rectangle, Circle, and Square from shape. Calculate and display area of Rectangle, Circle, and Square

ALGORITHM:

Step 1: Declare the necessary variables for length, breadth, and radius.

Step 2: Create a scanner object to read user input.

Step 3: Prompt the user to enter the length and breadth of a rectangle.

Step 4: Read and store the user input values for length and breadth.

Step 5: Calculate the area of the rectangle using the formula: area = length * breadth.

Step 6: Display the calculated area of the rectangle.

Step 7: Prompt the user to enter the radius of a circle.

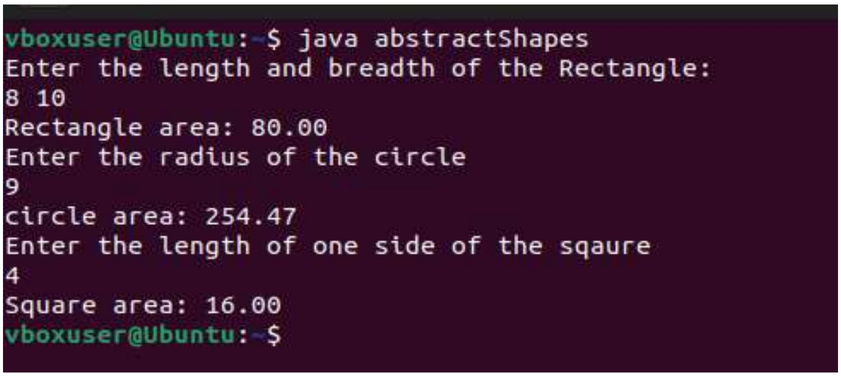
Step 8: Read and store the user input value for the radius.

Step 9: Calculate the area of the circle using the formula: area = π * radius^2.

Step 10: Display the calculated area of the circle.

```
      public Circle(double radius) {
        this.radius = radius;    }
    double findArea() {
        return Math.PI * Math.pow(radius, 2);     }    }
class Square extends Shape {
    private double side;
    public Square(double side) {
        this.side = side;        }
    double findArea() {
        return Math.pow(side, 2);    }     }
```

OUTPUT



```
vboxuser@Ubuntu:~$ java abstractShapes
Enter the length and breadth of the Rectangle:
8 10
Rectangle area: 80.00
Enter the radius of the circle
9
circle area: 254.47
Enter the length of one side of the sqaure
4
Square area: 16.00
vboxuser@Ubuntu:~$
```

RESULT: The program has been executed successfully and output obtained.