

**Object oriented programming lab**

**Experiment No.: 1**

**Aim**

1. Area of different shapes using overloaded functions

**Procedure**

**c**lass 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 obj = new OverloadDemo();

ob.area(8);

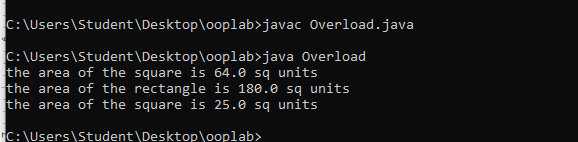
ob.area(12,15);

ob.area(5);

}

}

**Output Screenshot**

****

**Experiment No:2**

**Aim**

2. 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 own data members department, Subjects taught and constructors to initialize these data members and also include display function to display all the data members. Use array of objects to display details of N teachers.

**Procedure**

class EMPS{

public static void main(String[] args) {

Teacher tobj[] = new Teacher[2];

tobj[0] = new Teacher("101","Rekha","Rosevilla",50000,"MCA","DS");

tobj[1] = new Teacher("102","Riya","Deepalayam",110000,"BBA","Commerce");

tobj[0].display();

tobj[1].display();

}

}

class Employees {

String Empid;

String Name;

String Address;

int Salary;

Employees(String id,String name,String addr,int salary){

this.Empid = id;

this.Name = name;

this.Address = addr;

this.Salary = salary;

}

void display(){

System.out.println("EmpID : " + this.Empid);

System.out.println("Name : " + this.Name);

System.out.println("Address : " + this.Address);

System.out.println("Salary : " + this.Salary);

}

}

class Teacher extends Employees{

String Department;

String Subject;

Teacher(String id,String name,String addr,int salary,String dept,String subj){

super(id,name,addr,salary);

this.Department=dept;

this.Subject=subj;

}

void display(){

System.out.println("\*\*\*\*EMPLOYEE DETAILS\*\*\*\*");

super.display();

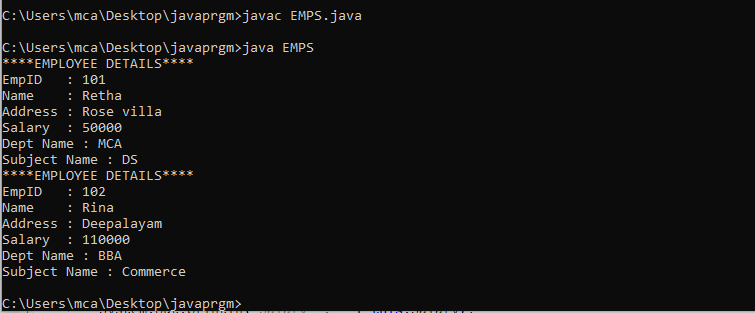
System.out.println("Dept Name : " + this.Department);

System.out.println("Subject Name : " + this.Subject);

}

}

**Output Screenshot**

****

**Experiment No:3**

**Aim**

3. Create a class ‘Person’ with data members Name, Gender, Address, Age and a 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.

**Procedure**

import java.util.Scanner;

class person{

String Name;

String Gender;

String Address;

int Age;

person(String name,String gender,String address,int age){

this.Name = name;

this.Gender = gender;

this.Address = address;

this.Age = age;

}

}

class Employee extends person

{

int Empid;

String Company\_name;

String Qualification;

long Salary;

Employee(String name,String gender,String address,int age,int empid, String company\_name, String qualification, long salary);

{

super(name,gender,address,age);

this.Empid = empid;

this.Company\_name = company\_name;

this.Qualification = qualification;

this.Salary = salary;

}

}

public class Teacher2 extends Employee{

String Subject;

String Department;

String Teacherid;

Teacher2(String name,String gender,String address, int age,int empid, String company\_name, String qualification,long salary, String subject, String department, String teacherid){

super(name,gender,address,age,empid,company\_name,qualification,salary);

this.Subject=subject;

this.Department=department;

this.Teacherid=teacherid;

}

void display(){

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

System.out.println("Gender: "+Gender);

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

System.out.println("Age: "+Age);

System.out.println("Employee id: "+Empid);

System.out.println("Company Name: "+Company\_name);

System.out.println("Qualification: "+Qualification);

System.out.println("Salary: "+Salary);

System.out.println("Subject: "+Subject);

System.out.println("Department: "+Department);

System.out.println("Teacher id: "+Teacherid);

}

public static void main(String[] args) {

System.out.println("\nEnter the No. of Teacher's");

Scanner sc1 = new Scanner(System.in);

int num = sc1.nextInt();

Teacher2 arr[]=new Teacher2[num];

System.out.println("\n Enter the Teacher 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 Name: ");

String a =sc.next();

System.out.println("\n Gender: ");

String b =sc.next();

System.out.println("\n Address: ");

String c =sc.next();

System.out.println("\n Age: ");

int d =sc.nextInt();

System.out.println("\n Employee id: ");

int e =sc.nextInt();

System.out.println("\n Company name: ");

String f =sc.next();

System.out.println("\n Qualification: ");

String g =sc.next();

System.out.println("\n Salary: ");

long h =sc.nextLong();

System.out.println("\n Subject: ");

String k =sc.next();

System.out.println("\n Department: ");

String l =sc.next();

System.out.println("\n Teacher Id: ");

String n =sc.next();

arr[i]=new Teacher2(a,b,c,d,e,f,g,h,k,l,n);

}

sc.close();

System.out.println("\n\*\*\*\*\*\*\*\*Informations of all the Teacher's\*\*\*\*\*\*\*\*\*\*\*\*");

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

j=i+1;

System.out.println("\n"+j+").");

arr[i].display();

}

sc1.close();

}

}

**Output Screenshot**

**Experiment No:4**

**Aim**

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

**Procedure**import java.util.Scanner;

class Publisher

{ int p\_id;

String p\_name;

Publisher(int p\_id,String p\_name)

{ this.p\_id=p\_id;

this.p\_name=p\_name;

}

}

class Book extends Publisher

{ int b\_id;

String b\_name;

Book(int p\_id, String p\_name, int b\_id, String b\_name)

{ super(p\_id, p\_name);

this.b\_id=b\_id;

this.b\_name=b\_name;

}

}

class Literature extends Book

{ String cat;

Literature(int p\_id, String p\_name, int b\_id, String b\_name)

{ super(p\_id, p\_name, b\_id, b\_name);

this.cat="Literature";

}

void Display4()

{ System.out.println("\n\n.............Literature book details...........");

System.out.println("Category name : "+this.cat);

System.out.println("Publisher id : "+this.p\_id);

System.out.println("Publisher name : "+ this.p\_name);

System.out.println("Book id : " +this.b\_id);

System.out.println("Book name : "+ this.b\_name);

}

}

class Fiction extends Book

{ String cat;

Fiction(int p\_id, String p\_name, int b\_id, String b\_name)

{ super(p\_id, p\_name, b\_id, b\_name);

this.cat="Fiction";

}

void Display4()

{ System.out.println("\n\n.............Fiction book details...........");

System.out.println("Category name : "+this.cat);

System.out.println("Publisher id : "+this.p\_id);

System.out.println("Publisher name : "+ this.p\_name);

System.out.println("Book id : " +this.b\_id);

System.out.println("Book name : "+ this.b\_name);

}

}

public class PublisherBooks

{ public static void main(String[] args)

{ int p\_id, b\_id;

String p\_name, b\_name,t;

Scanner s=new Scanner(System.in);

System.out.print("\nEnter the type of book(Type 'l/L' for Literature/'f/F' for Fiction)? ");

t=s.next();

if(t.equals("l")||t.equals("L"))

{ System.out.print("\nEnter the Publisher ID :");

p\_id=s.nextInt();

System.out.print("Enter the Publisher Name :");

p\_name=s.next();

System.out.print("Enter the Book ID :");

b\_id=s.nextInt();

System.out.print("Enter the Book Name :");

b\_name=s.next();

Literature lit=new Literature(p\_id, p\_name, b\_id, b\_name);

lit.Display4();

}

else if(t.equals("f")||t.equals("F"))

{ System.out.print("\nEnter the Publisher ID :");

p\_id=s.nextInt();

System.out.print("Enter the Publisher Name :");

p\_name=s.next();

System.out.print("Enter the Book ID :");

b\_id=s.nextInt();

System.out.print("Enter the Book Name :");

b\_name=s.next();

Fiction fic=new Fiction(p\_id, p\_name, b\_id, b\_name);

fic.Display4();

}

else

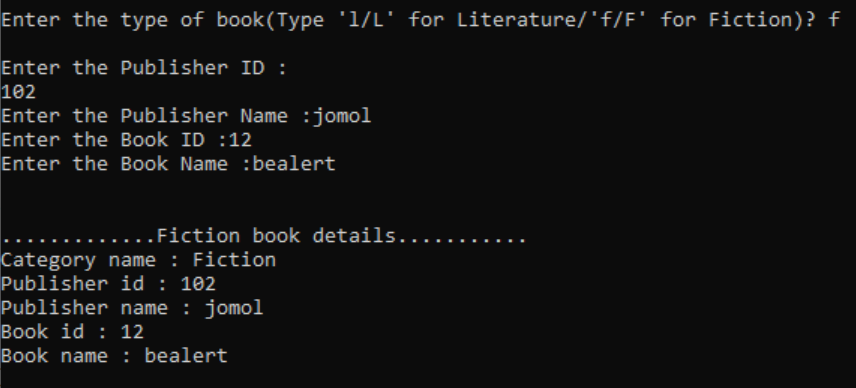
{ System.out.println("\n\n!!!!!!!!!!!!!!!Entry for type of book is not valid!!!!!!!!!!!!!! ");

}

}

}

**Output Screenshot**

****

**Experiment No:5**

**Aim:**

5. Create classes Student and Sports. Create another class Result inherited from Student and Sports. Display the academic and sports score of a student.

**Procedure**

import java.util.Scanner;

class sports{

String sport;

int Rating;

sports(String spo, int ra){

sport = spo;

Rating = ra;

}

}

class student extends sports{

String Grade;

double Overall\_per;

student(String spo, int ra,String gd, double per ){

super(spo, ra);

Grade = gd;

Overall\_per = per;

}

}

public class result extends student {

result(String spo, int ra,String gd, double per ){

super(spo, ra, gd, per);

}

void display(){

System.out.println("\nSports Details of Student");

System.out.println("Sport :"+sport);

System.out.println("Rating :"+Rating);

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 Sport: ");

String a =sc.next();

System.out.println("\n Sport Rating 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();

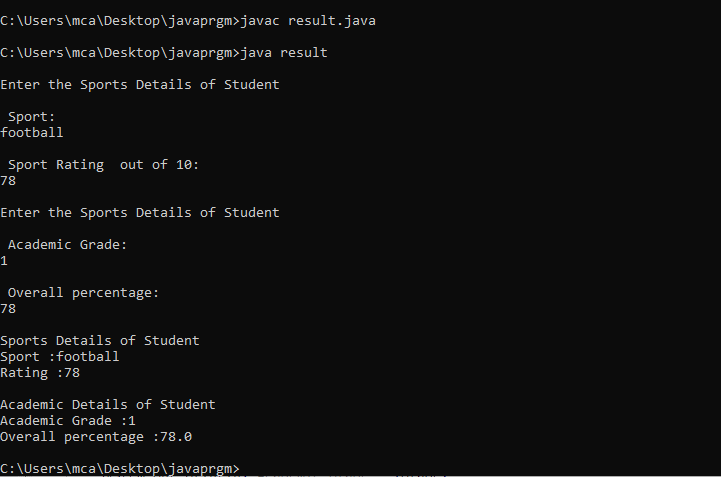
result obj= new result(a,b,c,d);

obj.display();

}

}

**Output Screenshot**

****