

UNIVERSITY OF CALCUTTA

B.Sc Semester 5 Honours Examination-2024

Under CBCS System

University Registration Number: 012-1114-1749-21

University Roll Number: 213012-21-0396

Subject: Computer Science

Subject Code: CMSA

Paper: Object Oriented Programming Lab

Paper Code:CC-12

Semester: 5

Year: 2024

Index

Sl No.	Date	Assignment	Page No.	Teacher's Signature
01		Assignment 1	01	
02		Assignment 2	03	
03		Assignment 3	06	
04		Assignment 4	08	
05		Assignment 5	10	
06		Assignment 6	12	
07		Assignment 7	13	
08		Assignment 8	17	
09		Assignment 9	19	
10		Assignment 10	21	

Assignment 1

A class called MyPoint, which models a 2D point with x and y coordinates. It contains:

- Two instance variables x (int) and y (int).
- A default (or "no-argument" or "no-arg") constructor that construct a point at the default location of (0, 0).
- A overloaded constructor that constructs a point with the given x and y coordinates.
- A method setXY() to set both x and y.
- A method getXY() which returns the x and y in a 2-element int array.
- A toString() method that returns a string description of the instance in the format "(x, y)".
- A method called distance(int x, int y) that returns the distance from this point to other point at the given (x, y) coordinates, Write the MyPoint class. Also write a test driver (called TestMyPoint) to test all the public methods defined in the class.

Code:

[MyPoint.java]

```
package assignment01;

public class MyPoint {
    private int x;
    private int y;

    public MyPoint() {
        this.x = 0;
        this.y = 0;
    }

    public MyPoint(int x, int y) {
        this.setXY(x, y);
    }

    public void setXY(int x, int y) {
        this.x = x;
        this.y = y;
    }

    public int[] getXY() {
        int[] arr = { this.x, this.y };
        return arr;
    }
}
```

```

    }

    public double distance(int x, int y) {
        return Math.sqrt(Math.pow((this.x - x), 2) + Math.pow((this.y - y), 2));
    }

    public String toString() {
        return "(" + this.x + ", " + this.y + ")";
    }
}
[TestMyPoint.java]
package assignment01;

public class TestMyPoint {
    public static void main(String[] args) {
        MyPoint p1 = new MyPoint();
        System.out.println("p1: " + p1);
        p1.setXY(3, 4);
        System.out.println("p1: " + p1);
        System.out.println("Distance of p1 from (5,6) : " + p1.distance(5, 6));
        MyPoint p2 = new MyPoint(8, 6);
        System.out.println("p2: " + p2);
        System.out.println("Distance of p2 from point (2,3) : " + p2.distance(2, 3));
        System.out.println("Distance of P2 from origin (0,0) : " + p2.distance(0, 0));
    }
}

```

Output:

```

p1: (0, 0)
p1: (3, 4)
Distance of p1 from (5,6) : 2.8284271247461903
p2: (8, 6)
Distance of p2 from point (2,3) : 6.708203932499369
Distance of P2 from origin (0,0) : 10.0

```

Assignment 2

Create a superclass 'Person' and two subclasses 'Student' and 'Staff'. The following are the instance variables and methods:

a. For 'Person' instance variables: name:String, address:String. Initiate variable through constructor, incorporate one method setPerson() that updates Person variables , another method toString() that shows Person details as "Person[name=?,address=?"].

b. For 'Student' sub class instance variables: program:String, year:String, fees:double. Initiate both 'Student' and 'Person' variables through constructor, incorporate one method setStudent() that updates both student and 'Person' data, another method toString() that shows 'Person-Student' details as "Person[name=?,address=?,Program=?,Year=?,Fees=?"].

c. For 'Staff' subclass instance variables: school:String, pay:double. Initiate both 'Staff' and 'Person' variables through constructor, incorporate one method setStaff() that updates both 'staff' and 'Person' data, another method toString() that shows 'Person-Staff' details as "Person[name=?,address=?,School=?,Pays=?"].

Write the classes and a test driver main class to test all functions mentioned above.

Code:

[Person.java]

```
package assignment02;

public class Person {
    protected String name;
    protected String address;
    public Person() {
        this.name = "";
        this.address = "";
    }
    public Person(String name, String address) {
        this.setPerson(name, address);
    }
    public void setPerson(String name, String address) {
        this.name = name;
        this.address = address;
    }
    public String toString() {
        return "Person[Name=" + this.name + ", Address=" + this.address + "]";
    }
}
```

[Staff.java]

```
package assignment02;

public class Staff extends Person{

    private String school;

    private double pay;

    public Staff() {

        this.school = "";

        this.pay = 0.0;

    }

    public Staff(String name, String address, String school, double pay) {

        super(name, address);

        this.setStaff(school, pay);

    }

    public void setStaff(String school, double pay) {

        this.school = school;

        this.pay = pay;

    }

    public String toString() {

        return "Person[Name=" + super.name + ", Adress=" + super.address + ", School=" + this.school

            + ", Pay=" + this.pay + "];"

    }

}
```

[Student.java]

```
package assignment02;

public class Student extends Person {

    private String program;

    private String year;

    private double fees;

    public Student() {

        super();

        this.program = "";

        this.year = "";

        this.fees = 0.0;

    }

}
```

```

    }

    public Student(String name, String address, String program, String year, double fees) {
        super(name, address);
        this.setStudent(program, year, fees);
    }

    public void setStudent(String program, String year, double fees) {
        this.program = program;
        this.year = year;
        this.fees = fees;
    }

    public String toString() {
        return "Person[Name=" + super.name + ", Address="+super.address + ", Program=" + this.program
            + ", Year=" + this.year + ", Fees=" + this.fees + "]\n";
    }
}

```

[Main.java]

```

package assignment02;

public class Main {

    public static void main(String[] args) {
        Person p = new Person("Sayantan", "Chandannagar");
        System.out.println(p);

        Person p1 = new Person("Samaita", "Chandannagar");
        System.out.println(p1);

        Student s = new Student("Shreya", "Kolkata", "B.Tech", "2024", 500000);
        System.out.println(s);

        Staff st = new Staff("Mehuli", "Kolkata", "IIT Kharagpur", 100000);
        System.out.println(st);
    }
}

```

Output:

Person[Name=Sayantan, Address=Chandannagar]

Person[Name=Samaita, Address=Chandannagar]

Person[Name=Shreya, Address=Kolkata, Program=B.Tech, Year=2024, Fees=500000.0]

Person[Name=Mehuli, Address=Kolkata, School=IIT Kharagpur, Pay=100000.0]

Assignment 3

Create a base class 'Square' having instance variable side:double. Initiate variable using constructor, a method 'getVolume() : double' that calculates volume and print it. Create a derived class 'Cylinder' having instance variable height:double. Initiate variables of both classes through constructor, override method 'getVolume() : double' to calculate volume of cylinder taking 'side' variable of base class as 'radius' and print it.

Code:

[Square.java]

```
package assignment03;

public class Square {

    protected double side;

    public Square() {
        this.side = 0.0;
    }

    public Square(double side) {
        this.side = side;
    }

    public double getVolume() {
        return this.side * this.side * this.side;
    }
}
```

[Cylinder.java]

```
package assignment03;

public class Cylinder extends Square {

    private double height;

    public Cylinder() {
        super();
        this.height = 0.0;
    }

    public Cylinder(double side, double height) {
        super(side);
        this.height = height;
    }
}
```



```

    public double getVolume() {
        return super.side*super.side * this.height * Math.PI;
    }
}

[Main.java]
package assignment06;
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the Radius of the Cylinder: ");
        double radius = sc.nextDouble();
        System.out.print("Enter the Height of the Cylinder: ");
        double height = sc.nextDouble();
        Cylinder c = new Cylinder(radius, height);
        System.out.println("Volume of Cylinder is :"+ c.getVolume());
        System.out.print("Enter the Side of the Cube: ");
        double side = sc.nextDouble();
        Square s = new Square(side);
        System.out.println("Volume of Cube is :"+ s.getVolume());
        sc.close();
    }
}

```

Output:

```

Enter the Radius of the Cylinder: 4
Enter the Height of the Cylinder: 8
Volume of Cylinder is :402.1238596594935
Enter the Side of the Cube: 9
Volume of Cube is :729.0

```

Assignment 4

Consider you are designing vehicles engine with 'speed:int, gear:int'. you can define your engine functionalities 'speedUp(value)' and 'changeGear(value)' in an interface. The class which is implementing the interface should implement all the methods in the interface.

Code:

[Engine.java]

```
package assignment04;

public interface Engine {

    void speedUp(int speed);
    void changeGear(int gear);
}
```

[Vehicle.java]

```
package assignment04;

public class Vehicle implements Engine {

    private int speed;
    private int gear;

    public Vehicle() {
        this.speed = 0;
        this.gear = 0;
    }

    public Vehicle(int speed, int gear) {
        this.speed = speed;
        this.gear = gear;
    }

    public void speedUp(int speed) {
        this.speed += speed;
    }

    public void changeGear(int gear) {
        this.gear = gear;
    }

    public String toString() {
        return ("Vehicle: Speed: " + this.speed + " Gear: " + this.gear);
    }
}
```

[Main.java]

```
package assignment04;
```

```
public class Main {
```

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

```
        Vehicle v = new Vehicle(10, 2);
```

```
        System.out.println(v);
```

```
        v.speedUp(24);
```

```
        v.changeGear(3);
```

```
        System.out.println("After speedUp(24) and changeGear(3)");
```

```
        System.out.println(v);
```

```
    }
```

```
}
```

Output:

Vehicle: Speed: 10 Gear: 2

After speedUp(24) and changeGear(3)

Vehicle: Speed: 34 Gear: 3

Assignment 5

Write a Java program to insert an element (specific position) into an array

Code:.

[Main.java]

```
package assignment5;

import java.util.Scanner;

public class Main {

    public static void main(String[] args) {

        System.out.print("Enter the size of the array: ");

        Scanner sc = new Scanner(System.in);

        int size = sc.nextInt();

        int[] array = new int[size];

        System.out.println("Enter the elements of the array: ");

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

            System.out.print("Enter the element at index "+i+": ");

            array[i] = sc.nextInt();

        }

        System.out.println("The array is: ");

        printArray(array);

        System.out.print("Enter the element to be inserted: ");

        int element = sc.nextInt();

        System.out.print("Enter the position at which the element is to be inserted: ");

        int pos = sc.nextInt();

        sc.close();

        array = insertElement(array, element, pos);

        System.out.println("The new array is: ");

        printArray(array);

    }

    static int[] insertElement(int[] array, int element, int pos){

        int[] newArray = new int[array.length+1];

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

            newArray[i] = array[i];

        }

        newArray[pos] = element;
```

```

        for(int i=pos+1; i<newArray.length; i++){
            newArray[i] = array[i-1];
        }
        return newArray;
    }

    static void printArray(int[] array){
        for(int i=0; i<array.length; i++){
            System.out.print(array[i]+" ");
        }
        System.out.println();
    }
}

```

Output:

Enter the size of the array: 5

Enter the elements of the array:

Enter the element at index 0: 1

Enter the element at index 1: 2

Enter the element at index 2: 4

Enter the element at index 3: 5

Enter the element at index 4: 6

The array is:

1 2 4 5 6

Enter the element to be inserted: 3

Enter the position at which the element is to be inserted: 2

The new array is:

1 2 3 4 5 6

Assignment 6

Write a Java method to count all words in a string.

Test Data:

Input the string: The quick brown fox jumps over the lazy dog.

Expected Output:

Number of words in the string: 9

Code:

[Main.java]

```
package assignment18;

import java.util.Scanner;

public class Main {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.print("Enter the string: ");

        String str = sc.nextLine();

        str = str.trim();

        str = str.replaceAll("[^a-zA-Z0-9]", " "); // replace all non-alphanumeric characters with space not
        required in this case but still doing it

        String[] words = str.split("\\s+");

        System.out.print("Input the string after removing white spaces: ");

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

            System.out.print(words[i] + " ");

        }

        System.out.println();

        System.out.println("Number of words in the string: " + words.length);

        sc.close();

    }

}
```

Output:

Enter the string: The quick brown fox jumps over the lazy dog.

Input the string after removing white spaces: The quick brown fox jumps over the lazy dog

Number of words in the string: 9

Assignment 7

Write a program in java with class Employee and do the following operations on it

- a) Create two constructor default and with Object as parameter to initialize class variables.
- b) Create a function Calculate which calculates the pf and allowances on the salary of employee and return the all values as an object.

Code:

[Employee.java]

```
package assignment07;

public class Employee {

    private String ename;

    private int eid;

    private double salary;

    private double pf=0.0;

    private double allowance=0.0;

    private double totalSalary=0.0;

    Employee(){

        this.ename = "";

        this.eid = 0;

        this.salary = 0.0;

    }

    Employee(Employee emp){

        this.ename = emp.ename;

        this.eid = emp.eid;

        this.salary = emp.salary;

        this.pf = emp.pf;

        this.allowance = emp.allowance;

        this.totalSalary = emp.totalSalary;

    }

    Employee(String ename, int eid, double salary){

        this.ename = ename;

        this.eid = eid;

        this.salary = salary;

    }

}
```

```

Employee calculate(){
    this.pf = this.salary * 0.12;
    this.allowance = this.salary * 0.4;
    this.totalSalary = this.salary + this.allowance - this.pf;
    return this;
}

@Override
public String toString(){
    return
this.eid+"\t"+this.ename+"\t\t"+this.salary+"\t\t"+this.pf+"\t\t"+this.allowance+"\t\t"+this.totalSalary;
}
}

```

[Main.java]

```

package assignment07;
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter number of employees:");
        int n = sc.nextInt();
        Employee[] employees = new Employee[n];
        for (int i = 0; i < n; i++) {
            System.out.println("Enter details of employee " + (i + 1));
            System.out.print("Enter name: ");
            String name = sc.next();
            System.out.print("Enter ID: ");
            int id = sc.nextInt();
            System.out.print("Enter salary: ");
            double salary = sc.nextDouble();
            employees[i] = new Employee(name, id, salary);
            employees[i] = employees[i].calculate();
        }
    }
}

```



```

        System.out.println("Employee Details");
        printEmpTable(employees);
        sc.close();
    }
    static void printEmpTable(Employee[] employees) {
        System.out.println("ID\tName\tSalary\tPF\tAllowance\tTotal Salary");
        for (int i = 0; i < employees.length; i++) {
            System.out.println(employees[i]);
        }
    }
}

```

OUTPUT

Enter number of employees:5

Enter details of employee 1

Enter name: Ram

Enter ID: 01

Enter salary: 12000

Enter details of employee 2

Enter name: Shyam

Enter ID: 02

Enter salary: 14000

Enter details of employee 3

Enter name: Ajay

Enter ID: 03

Enter salary: 17000

Enter details of employee 4

Enter name: Bijoy

Enter ID: 4

Enter salary: 13000

Enter details of employee 5

Enter name: Sujai

Enter ID: 05

Enter salary: 25000

Employee Details

ID	Name	Salary	PF	Allowance	Total Salary
1	Ram	12000.0	1440.0	4800.0	15360.0
2	Shyam	14000.0	1680.0	5600.0	17920.0
3	Ajay	17000.0	2040.0	6800.0	21760.0
4	Bijoy	13000.0	1560.0	5200.0	16640.0
5	Sujai	25000.0	3000.0	10000.0	32000.0

Assignment 8

Write a program to create your own exception as NegativeSizeException whenever negative values are put in an array.

Code:

[NegativeSizeException.java]

```
package assignment08;

public class NegativeSizeException extends Exception{

    NegativeSizeException(String s){

        super(s);

    }

}
```

[Main.java]

```
package assignment08;

import java.util.Scanner;

public class Main {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        try {

            System.out.print("Enter size of array: ");

            int n = sc.nextInt();

            int[] arr = new int[n];

            System.out.println("Enter elements of array:");

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

                System.out.print("Enter element " + (i + 1) + ": ");

                arr[i] = sc.nextInt();

                if (arr[i] < 0) {

                    throw new NegativeSizeException("Array elements cannot be negative");

                }

            }

            System.out.println("Array elements are:");

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

                System.out.print(arr[i]+" ");

            }

        }

    }

}
```

```
    } catch (Exception e) {  
        System.out.println(e);  
    } finally {  
        sc.close();  
    }  
}  
}
```

Output (Without Invoking Exception):

Enter size of array: 5

Enter elements of array:

Enter element 1: 1

Enter element 2: 2

Enter element 3: 5

Enter element 4: 4

Enter element 5: 3

Array elements are:

1 2 5 4 3

Output (Invoking Exception):

Enter size of array: 6

Enter elements of array:

Enter element 1: 1

Enter element 2: 2

Enter element 3: 3

Enter element 4: -4

Assignment08.NegativeSizeException: Array elements cannot be negative

Assignment 9

Create a class Student with following operations

- a) create parameterized constructor to initialize the objects.
- b) create a function isEqual() to check whether the two objects are equal or not which returns the Boolean value and gets two objects.
- c) print the result in main method if objects are equals or not (take variables as your assumption)

Code:

[Student.java]

```
package assignment09;

public class Student {

    private String name;
    private int roll;

    Student(String name, int roll) {
        this.name = name;
        this.roll = roll;
    }

    String getName() {
        return this.name;
    }

    int getRoll() {
        return this.roll;
    }

    public String toString() {
        return "Name: " + this.name + ", Roll: " + this.roll;
    }

    public boolean isEqual(Student s) {
        this.name = this.name.toLowerCase();
        s.name = s.name.toLowerCase();
        this.name = this.name.trim();
        s.name = s.name.trim();
        this.name = this.name.replaceAll("\\s+", " ");
        s.name = s.name.replaceAll("\\s+", " ");
```

```
        return (this.name.equals(s.name) && this.roll == s.roll) ? true : false; //use of ternary operator to return boolean value directly without using if-else block
```

```
    }
```

```
}
```

[Main.java]

```
package assignment09;
```

```
public class Main {
```

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

```
        Student student1 = new Student("John", 101);
```

```
        Student student2 = new Student("Jane", 102);
```

```
        Student student3 = new Student("John", 101);
```

```
        System.out.println("Student 1: " + student1);
```

```
        System.out.println("Student 2: " + student2);
```

```
        System.out.println("Student 3: " + student3);
```

```
        System.out.println("Is Student 1 equal to Student 2: " + student1.isEquals(student2));
```

```
        System.out.println("Is Student 1 equal to Student 3: " + student1.isEquals(student3));
```

```
    }
```

```
}
```

Output:

Student 1: Name: John, Roll: 101

Student 2: Name: Jane, Roll: 102

Student 3: Name: John, Roll: 101

Is Student 1 equal to Student 2: false

Is Student 1 equal to Student 3: true

Assignment 10

Create an abstract class employee, having its properties and abstract function for calculating net salary and displaying the information. Derive manager and clerk class from this abstract class and implement the abstract method net salary and override the display method.

Code:

[Employee.java]

```
package assignment23;

public abstract class Employee {

    String name;

    int id;

    double basicSalary;

    double netSalary;

    Employee(String name, int id, double basicSalary) {

        this.name = name;

        this.id = id;

        this.basicSalary = basicSalary;

    }

    abstract void calculateNetSalary();

    void display() {

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

        System.out.println("ID: " + this.id);

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

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

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

    }

}
```

[Manager.java]

```
package assignment23;

public class Manager extends Employee {

    String department;

    Manager(String name, int id, double basicSalary, String department) {

        super(name, id, basicSalary);

        this.department = department;

    }

}
```

```

    }

    void calculateNetSalary() {
        this.netSalary = this.basicSalary + (this.basicSalary * 0.1) + (this.basicSalary * 0.05); // 10% HRA and 5% DA
    }

    @Override
    void display() {
        System.out.println("Manager of Department: " + this.department);
        super.display();
    }
}

```

[Clerk.java]

```

package assignment23;

public class Clerk extends Employee {
    String department;

    Clerk(String name, int id, double basicSalary, String department) {
        super(name, id, basicSalary);
        this.department = department;
    }

    void calculateNetSalary() {
        this.netSalary = this.basicSalary + (this.basicSalary * 0.08) + (this.basicSalary * 0.03); // 8% HRA and 3% DA
    }

    @Override
    void display() {
        System.out.println("Clerk of Department: " + this.department);
        super.display();
    }
}

```


[Main.java]

```
package assignment23;

public class Main {

    public static void main(String[] args) {

        //sales dept

        Manager manager1 = new Manager("John", 101, 10000, "Sales");
        Clerk clerk1 = new Clerk("Jane", 102, 5500, "Sales");
        Clerk clerk2 = new Clerk("Jack", 103, 7000, "Sales");
        Clerk clerk3 = new Clerk("Jill", 104, 7500, "Sales");

        //marketing dept

        Manager manager2 = new Manager("James", 105, 10000, "Marketing");
        Clerk clerk4 = new Clerk("Jenny", 106, 6000, "Marketing");
        Clerk clerk5 = new Clerk("Jared", 107, 65000, "Marketing");
        Clerk clerk6 = new Clerk("Jasmine", 108, 8000, "Marketing");

        //finance dept

        Manager manager3 = new Manager("Jasper", 109, 10000, "Finance");
        Clerk clerk7 = new Clerk("Jade", 110, 9000, "Finance");
        Clerk clerk8 = new Clerk("Jasper", 111, 5500, "Finance");

        Employee[] employees = {manager1, clerk1, clerk2, clerk3, manager2, clerk4, clerk5, clerk6,
manager3, clerk7, clerk8};

        for (Employee employee : employees) {
            employee.calculateNetSalary();
            employee.display();
        }
    }
}
```

Output:

Manager of Department: Sales

Name: John

ID: 101

Basic Salary: 10000.0

Net Salary: 11500.0

Clerk of Department: Sales

Name: Jane

ID: 102

Basic Salary: 5500.0

Net Salary: 6105.0

Clerk of Department: Sales

Name: Jack

ID: 103

Basic Salary: 7000.0

Net Salary: 7770.0

Clerk of Department: Sales

Name: Jill

ID: 104

Basic Salary: 7500.0

Net Salary: 8325.0

Manager of Department: Marketing

Name: James

ID: 105

Basic Salary: 10000.0

Net Salary: 11500.0

Clerk of Department: Marketing

Name: Jenny

ID: 106

Basic Salary: 6000.0

Net Salary: 6660.0

Clerk of Department: Marketing

Name: Jared

ID: 107

Basic Salary: 65000.0

Net Salary: 72150.0

Clerk of Department: Marketing

Name: Jasmine

ID: 108

Basic Salary: 8000.0

Net Salary: 8880.0

Manager of Department: Finance

Name: Jasper

ID: 109

Basic Salary: 10000.0

Net Salary: 11500.0

Clerk of Department: Finance

Name: Jade

ID: 110

Basic Salary: 9000.0

Net Salary: 9990.0

Clerk of Department: Finance

Name: Jasper

ID: 111

Basic Salary: 5500.0

Net Salary: 6105.0