

**SSN College of Engineering**  
**Department of Computer Science and Engineering**  
**UCS1313 – Object Oriented Programming Using Java Lab**

**II Year CSE - A, B Section ( III Semester)**

**Academic Year 2020-21**

**Batch: 2019- 2023**

**Faculty Incharge : S.Rajalakshmi / S.Lakshmi Priya**

---

**Exercise 1. Java Programs- Classes, Objects and Constructors**

1. Develop a Java application to generate Electricity bill. Create a class with the following members: Consumer no., consumer name, previous month reading, current month reading, type of EB connection (i.e domestic or commercial). Compute the bill amount using the following tariff.

If the type of the EB connection is domestic, calculate the amount to be paid as follows:

- First 100 units - Rs. 1 per unit
- 101-200 units - Rs. 2.50 per unit
- 201 -500 units - Rs. 4 per unit
- 501 units - Rs. 6 per unit

If the type of the EB connection is commercial, calculate the amount to be paid as follows:

- First 100 units - Rs. 2 per unit
- 101-200 units - Rs. 4.50 per unit
- 201 -500 units - Rs. 6 per unit
- 501 units - Rs. 7 per unit

2. Write a java program to create a class named ‘Student’ with name, id, dept, 3 marks as data members. Write function to assign the inputs, calculate grade, display and search. Perform these operations for ‘n’ number of students. [Search using id and dept – use method overloading]
3. Write a java program to create a class named ‘Employee’ with name, id, designation, years-of-experience, basicpay, DA, HRA, LIC, PF and no. of hours worked. Write functions to calculate the gross pay and net pay.

- If the designation is “Intern” then the salary can be calculated based on working hours.

Gross salary = no.of hours worked \* hourly wage + DA as 2000 + HRA as 1000

Deductions=LIC premium amount (if employee opted for that) + PF as 500

Net salary= Gross salary – Deductions

- If the designation is “Manager” then

Gross salary = Basicpay + DA as 40% of basicpay + HRA as 10% of basicpay

Deductions=LIC premium amount (if employee opted for that) + PF as 8% of basicpay

Net salary= Gross salary – Deductions

- If the designation is “others – Trainee, Analyst, Software engineer, TeamLead” then

Gross salary = Basicpay + DA as 30% of basicpay + HRA as 10% of basicpay

Deductions=LIC premium amount (if employee opted for that) + PF as 8% of basicpay

Net salary= Gross salary – Deductions

- Calculate the Payroll for ‘n’ employees and display the salary details for all employees.

- Prepare the payslip for a particular employee.

- Promote a particular employee based on the years of experience.

## Exercise 1

### Java Programs- Classes, Objects and Constructors

#### 1) Electricity bill generation

##### Source Code:

```
import java.util.Scanner;
class ElectBill{

    int cons_no,prev_read,curr_read;
    String cons_name,type;

    public static double computeBill(int prev_read,int curr_read,String ch)
    {
        int units;
        double pay=0.0;
        units=curr_read-prev_read;
        if(ch.equals("domestic"))
        {
            if(units<=100)
            {
                pay=units*1.0;
            }
            else if(units<=200 && units>100)
            {
                pay=100*1.0 + (units-100)*2.50;
            }
            else if(units>200 && units<=500)
            {
                pay=100*1.0 + 100*2.50 + (units-200)*4.0;
            }
            else
            {
                pay= 100*1.0 +100*2.50 + 300*4.0 +(units-500)*6;
            }
        }
        else if(ch.equals("commercial"))
        {
            if(units<=100)
            {
                pay=units*2.0;
            }
            else if(units<=200 && units>100)
            {
                pay=100*2.0 + (units-100)*4.0;
            }
            else if(units>200 && units<=500)
            {
                pay=100*2.0 + 100*4.0 + (units-200)*6.0;
            }
            else
            {
                pay= 100*2.0 +100*4.0 + 300*6.0 +(units-500)*12;
            }
        }
    }
}
```

```

        {
            pay=100*2.0 + (units-100)*4.50;
        }
        else if(units>200 && units<=500)
        {
            pay=100*2.0 + 100*4.50 + (units-200)*6.0;
        }
        else
        {
            pay= 100*2.0 +100*4.50 + 300*6.0 +(units-500)*7;
        }
    }
    return pay;
}

public void display()
{
    double p=computeBill(this.prev_read,this.curr_read,this.type);
    System.out.println("\n\tELECTRICITY BILL\n");
    System.out.println("CONSUMER NUMBER: "+this.cons_no);
    System.out.println("CONSUMER NAME: "+this.cons_name);
    System.out.println("CONNECTION TYPE: "+this.type);
    System.out.println("PREVIOUS MONTH READING: "+this.prev_read);
    System.out.println("CURRENT MONTH READING: "+this.curr_read);
    System.out.println("NUMBER OF UNITS USED: "+(this.curr_read-
this.prev_read));
    System.out.println("PAY AMOUNT: "+p+"\n");
}

public static void main(String args[])
{
    Scanner sc = new Scanner(System.in);
    ElectBill el1= new ElectBill();
    System.out.println("Enter customer no");
    el1.cons_no=sc.nextInt();
    sc.nextLine();
    System.out.println("Enter name");
    el1.cons_name=sc.nextLine();
    System.out.println("Enter previous month reading");
    el1.prev_read=sc.nextInt();
    System.out.println("Enter current month reading");
    el1.curr_read=sc.nextInt();
    sc.nextLine();
    System.out.println("Enter type: domestic or commercial");
    el1.type=sc.nextLine();
    el1.display();
    sc.close();
}
}

```

**Output:**

a) PS D:\01-College\assignments\oops lab\ex1> javac ElectBill.java  
PS D:\01-College\assignments\oops lab\ex1> java ElectBill  
Enter customer no  
122  
Enter name  
Sam  
Enter previous month reading  
4566  
Enter current month reading  
4777  
Enter type: domestic or commercial  
domestic

ELECTRICITY BILL

CONSUMER NUMBER: 122  
CONSUMER NAME: Sam  
CONNECTION TYPE: domestic  
PREVIOUS MONTH READING: 4566  
CURRENT MONTH READING: 4777  
NUMBER OF UNITS USED: 211  
PAY AMOUNT: 394.0

b) PS D:\01-College\assignments\oops lab\ex1> java ElectBill  
Enter customer no  
4566  
Enter name  
Sam  
Enter previous month reading  
677  
Enter current month reading  
88800  
Enter type: domestic or commercial  
commercial

ELECTRICITY BILL

CONSUMER NUMBER: 4566  
CONSUMER NAME: Sam  
CONNECTION TYPE: commercial

PREVIOUS MONTH READING: 677  
CURRENT MONTH READING: 88800  
NUMBER OF UNITS USED: 88123  
PAY AMOUNT: 615811.0

## 2) Student record management

### Source code:

```
import java.util.Scanner;
public class StudRec{

    String name,grade,dept;
    int id;
    int m[]=new int[3];

    private static final Scanner sc = new Scanner(System.in);
    public static void getInput(StudRec s[],int n)
    {
        for(int i=0;i<n;i++)
        {
            System.out.println("\n\tSTUDENT INPUT");
            sc.nextLine();
            System.out.println("Enter student name: ");
            s[i].name = sc.nextLine();
            System.out.println("Enter student ID: ");
            s[i].id = sc.nextInt();
            sc.nextLine();
            System.out.println("Enter student Department: ");
            s[i].dept = sc.nextLine();
            for(int j=0;j<3;j++)
            {
                System.out.println("Enter mark "+(j+1)+": ");
                s[i].m[j]= sc.nextInt();
            }
            //sc.nextLine();
        }
    }

    public String calcGrade()
    {
        int sum=0;
        double avg;
        for(int i=0;i<3;i++)
        {
            sum+=this.m[i];
        }
        avg=sum/3.0;
```

```

        if(avg>=90.0)
    {
        return "A grade";
    }
    else if(avg>=80.0)
    {
        return "B grade";
    }
    else if(avg>=70.0)
    {
        return "C grade";
    }
    else if(avg>=60.0)
    {
        return "D grade";
    }
    else if(avg>=50.0)
    {
        return "E grade";
    }
    else
    {
        return "F grade";
    }
}

void displayRecord()
{
    System.out.println("\n\tSTUDENT DISPLAY\n");
    System.out.println("Student name: "+this.name);
    System.out.println("Student ID: "+this.id);
    System.out.println("Student Department: "+this.dept);
    for(int i=0;i<3;i++)
    {
        System.out.println("Mark "+(i+1)+": "+this.m[i]);
    }
    this.grade=this.calcGrade();
    System.out.println("Student grade: "+this.grade);
}

public static void searchRecord(StudRec s[],int n,int id,String dept)
{
    System.out.println("SEARCHING RECORD...");
    int f=0;
    int i;
    for(i=0;i<n;i++)
    {
        if(s[i].id==id && dept.equals(s[i].dept))
        {
            f=1;
        }
    }
}

```

```

        break;
    }
}
if(f==1)
{
    System.out.println("\nFOUND");
    s[i].displayRecord();
}
else
{
    System.out.println("\nRECORD NOT FOUND!!!");
}
}

public static void searchRecord(StudRec s[],int n,int id)
{
    int f=0;
    int i;
    for(i=0;i<n;i++)
    {
        if(s[i].id==id)
        {
            f=1;
            break;
        }
    }
    if(f==1)
    {
        System.out.println("\nFOUND AT "+(i+1));
        s[i].displayRecord();
    }
    else
    {
        System.out.println("\nRECORD NOT FOUND!!!");
    }
}

public static void main(String args[])
{
    int n,o,id;
    char ch='y';
    String dep;
    System.out.println("Enter the number of student records: ");
    n=sc.nextInt();
    StudRec [] s = new StudRec[n];
    for(int i=0;i<n;i++)
    {
        s[i]=new StudRec();
    }
}

```

```

do
{
    System.out.println("\nMENU\n1.Input students\n2.Display students\
n3.Search using ID\n4.Search using ID and Dept");
    System.out.println("\nEnter option: ");
    o = sc.nextInt();
    switch(o)
    {
        case 1:
            getInput(s,n);
            break;

        case 2:
            for(int j=0;j<n;j++)
            {
                s[j].displayRecord();
            }
            break;

        case 4:
            System.out.println("Enter ID: ");
            id=sc.nextInt();
            sc.nextLine();
            System.out.println("Enter Department: ");
            dep=sc.nextLine();
            searchRecord(s,n,id,dep);
            break;

        case 3:
            System.out.println("Enter ID: ");
            id=sc.nextInt();
            searchRecord(s,n,id);
            break;

        default:
            System.out.println("INVALID!!!\"");
            break;
    }
    System.out.println("DO YOU WANT TO CONTINUE: ENTER Y OR y TO CONT
INUE\n");
    ch=sc.next().charAt(0);
    sc.nextLine();
}while(ch=='Y'||ch=='y');
System.out.println("\nDONE!!!\"");
}
}

```

**Output:**

```
PS D:\01-College\assignments\oops lab\ex1> javac StudRec.java
PS D:\01-College\assignments\oops lab\ex1> java StudRec
Enter the number of student records:
3
```

```
MENU
1.Input students
2.Display students
3.Search using ID
4.Search using ID and Dept
```

```
Enter option:
1
```

```
STUDENT INPUT
Enter student name:
Sam
Enter student ID:
1
Enter student Department:
CSE
Enter mark 1:
90
Enter mark 2:
99
Enter mark 3:
88
```

```
STUDENT INPUT
Enter student name:
Ram
Enter student ID:
2
Enter student Department:
EEE
Enter mark 1:
88
Enter mark 2:
77
Enter mark 3:
86
```

STUDENT INPUT

Enter student name:

James

Enter student ID:

3

Enter student Department:

ECE

Enter mark 1:

77

Enter mark 2:

66

Enter mark 3:

75

DO YOU WANT TO CONTINUE: ENTER Y OR y TO CONTINUE

Y

MENU

- 1.Input students
- 2.Display students
- 3.Search using ID
- 4.Search using ID and Dept

Enter option:

2

STUDENT DISPLAY

Student name: Sam

Student ID: 1

Student Department: CSE

Mark 1: 90

Mark 2: 99

Mark 3: 88

Student grade: A grade

STUDENT DISPLAY

Student name: Ram

Student ID: 2

Student Department: EEE

Mark 1: 88

Mark 2: 77  
Mark 3: 86  
Student grade: B grade

STUDENT DISPLAY

Student name: James  
Student ID: 3  
Student Department: ECE  
Mark 1: 77  
Mark 2: 66  
Mark 3: 75  
Student grade: C grade  
DO YOU WANT TO CONTINUE: ENTER Y OR y TO CONTINUE

Y

MENU  
1.Input students  
2.Display students  
3.Search using ID  
4.Search using ID and Dept

Enter option:  
3  
Enter ID:  
1

FOUND AT 1

STUDENT DISPLAY

Student name: Sam  
Student ID: 1  
Student Department: CSE  
Mark 1: 90  
Mark 2: 99  
Mark 3: 88  
Student grade: A grade  
DO YOU WANT TO CONTINUE: ENTER Y OR y TO CONTINUE

Y

MENU

1. Input students
2. Display students
3. Search using ID
4. Search using ID and Dept

Enter option:

3

Enter ID:

7

RECORD NOT FOUND!!

DO YOU WANT TO CONTINUE: ENTER Y OR y TO CONTINUE

Y

MENU

1. Input students
2. Display students
3. Search using ID
4. Search using ID and Dept

Enter option:

4

Enter ID:

1

Enter Department:

CSE

SEARCHING RECORD...

FOUND

#### STUDENT DISPLAY

Student name: Sam

Student ID: 1

Student Department: CSE

Mark 1: 90

Mark 2: 99

Mark 3: 88

Student grade: A grade

DO YOU WANT TO CONTINUE: ENTER Y OR y TO CONTINUE

Y

MENU  
1.Input students  
2.Display students  
3.Search using ID  
4.Search using ID and Dept

Enter option:

4

Enter ID:

8

Enter Department:

EEE

SEARCHING RECORD...

RECORD NOT FOUND!!!

DO YOU WANT TO CONTINUE: ENTER Y OR y TO CONTINUE

Y

MENU  
1.Input students  
2.Display students  
3.Search using ID  
4.Search using ID and Dept

Enter option:

8

INVALID!!!

DO YOU WANT TO CONTINUE: ENTER Y OR y TO CONTINUE

N

DONE!!!

### **3) Employee Payment Record:**

#### **Source Code:**

```
import java.util.Scanner;

public class EmpPay{

    private static final Scanner sc = new Scanner(System.in);
    String name,desig;
    int id,yrs,hrs;
    double bp,da,hra,pf,lic,gp,np,wage;

    public static void getInput(EmpPay emp[],int n)
    {
        for(int i=0;i<n;i++)
        {
            System.out.println("\n\tEMPLOYEE INPUT\n");
            sc.nextLine();
            System.out.println("Enter employee name: ");
            emp[i].name = sc.nextLine();
            System.out.println("Enter employee ID: ");
            emp[i].id = sc.nextInt();
            sc.nextLine();
            System.out.println("Enter employee designation\nManager,Intern,Trainee,Analyst,Software engineer,TeamLead:");
            emp[i].desig = sc.nextLine();
            if(emp[i].desig.equalsIgnoreCase("intern"))
            {
                System.out.println("Enter the hourly wages: ");
                emp[i].wage=sc.nextDouble();
            }
            System.out.println("Enter number of hours worked: ");
            emp[i].hrs = sc.nextInt();
            System.out.println("Enter number of years of experience: ");
            emp[i].yrs = sc.nextInt();
            System.out.println("Enter basic pay: ");
            emp[i].bp = sc.nextDouble();
            System.out.println("Enter LIC amount(0 if not applicable)");
            emp[i].lic = sc.nextDouble();
            System.out.println();
        }
    }

    public void calcPay()
    {
        if(desig.equalsIgnoreCase("intern"))
        {
            this.da=2000;
```

```

        this.hra=1000;
        this(pf=500;
        this.gp=this.hrs*this.wage+da+hra;
        this.np=this.gp-(this.lic+this.pf);
    }

    else if(desig.equalsIgnoreCase("manager"))
    {
        this.da=0.04*this.bp;
        this.hra=0.10*this.bp;
        this.pf=0.08*this.bp;
        this.gp=this.bp+this.da+this.hra;
        this.np=this.gp-(this.lic+this.pf);
    }

    else
    {
        this.da=0.03*this.bp;
        this.hra=0.10*this.bp;
        this.pf=0.08*this.bp;
        this.gp=this.bp+this.da+this.hra;
        this.np=this.gp-this.lic-this.pf;;
    }
}

void dispEmp()
{
    System.out.println("\nPAYOUT SLIP\n");
    System.out.println("Employee name:\t "+this.name);
    System.out.println("Employee designation:\t "+this.desig);
    System.out.println("Employee ID:\t "+this.id);
    System.out.println("Years of experience:\t "+this.yrs);
    System.out.println("Basic pay:\t "+this.bp);
    System.out.println("LIC amount:\t "+this.lic);
    this.calcPay();
    System.out.println("DA:\t "+this.da);
    System.out.println("HRA:\t "+this.hra);
    System.out.println("PF:\t "+this.pf);
    System.out.println("Gross pay:\t "+this.gp);
    System.out.println("Net pay:\t "+this.np);
}

static int searchEmp(EmpPay[ ] emp,int n)
{
    int i,f=0,id;
    System.out.println("Enter employee ID");
    id=sc.nextInt();
    for(i=0;i<n;i++)
    {

```

```

        if(id==emp[i].id)
        {
            f=1;
            break;
        }
    }
    if(f==1)
    {
        return i;
    }
    else
    {
        return -1;
    }
}

static void promoteEmp(EmpPay emp)
{
    System.out.println("\n\tPROMOTING AN EMPLOYEE\n");
    if(emp.yrs>=5)
    {
        System.out.println("\nCONGRATULATIONS\nYou are promoted!!!\n");
    }
    else
    {
        System.out.println("\nSORRY YOU DONT HAVE ENOUGH EXPERIENCE FOR PR
OMOTION!!\n");
    }
}

public static void main(String [] args)
{
    int n,o,i;
    char ch='y';
    System.out.println("\nEnter the number of employees");
    n=sc.nextInt();
    EmpPay emp[] = new EmpPay[n];
    for(i=0;i<n;i++)
    {
        emp[i]=new EmpPay();
    }
    do
    {
        System.out.println("\n\tMENU\n1.Input all records\n2.Display all
records\n3.Display payslip for one employee\n4.Promote an employee");
        o=sc.nextInt();
        switch(o)
        {
            case 1:

```

```

        getInput(emp,n);
        break;

        case 2:
        for(i=0;i<n;i++)
        {
            emp[i].dispEmp();
        }
        break;

        case 3:
        i=searchEmp(emp,n);
        if(i>=0)
        {
            emp[i].dispEmp();
        }
        else
        {
            System.out.println("\nEMPLOYEE NOT FOUND!!!\n");
        }
        break;

        case 4:
        i=searchEmp(emp,n);
        if(i>=0)
        {
            System.out.println("Employee name: "+emp[i].name);
            System.out.println("Number of years of experience: "+emp[i].yrs);
            promoteEmp(emp[i]);
        }
        else
        {
            System.out.println("\nEMPLOYEE NOT FOUND!!!\n");
        }
        break;

        default:
        System.out.println("\nINVALID OPTION!!!\n");
    }
    System.out.println("\nDO YOU WANT TO CONTINUE?\nENTER Y OR y TO C
ONTINUE\n");
    ch=sc.next().charAt(0);
    sc.nextLine();
}while(ch=='y'||ch=='Y');
System.out.println("\nDONE!!!!");
}
}

```

**Output:**

PS D:\01-College\assignments\oops\_lab\ex1> javac EmpPay.java

PS D:\01-College\assignments\oops\_lab\ex1> java EmpPay

Enter the number of employees

3

MENU

1. Input all records
2. Display all records
3. Display payslip for one employee
4. Promote an employee

1

EMPLOYEE INPUT

Enter employee name:

Sam

Enter employee ID:

1

Enter employee designation

Manager, Intern, Trainee, Analyst, Software engineer, TeamLead:

manager

Enter number of hours worked:

40

Enter number of years of experience:

6

Enter basic pay:

120000

Enter LIC amount(0 if not applicable)

3500

#### EMPLOYEE INPUT

Enter employee name:

Ram

Enter employee ID:

2

Enter employee designation

Manager,Intern,Trainee,Analyst,Software engineer,TeamLead:

INTERN

Enter the hourly wages:

60

Enter number of hours worked:

50

Enter number of years of experience:

2

Enter basic pay:

4000

Enter LIC amount(0 if not applicable)

0

## EMPLOYEE INPUT

Enter employee name:

James

Enter employee ID:

3

Enter employee designation

Manager,Intern,Trainee,Analyst,Software engineer,TeamLead:

Trainee

Enter number of hours worked:

38

Enter number of years of experience:

1

Enter basic pay:

3500

Enter LIC amount(0 if not applicable)

80

DO YOU WANT TO CONTINUE?

ENTER Y OR y TO CONTINUE

y

## MENU

1.Input all records

- 2.Display all records
- 3.Display payslip for one employee
- 4.Promote an employee

2

#### PAYMENT SLIP

Employee name: Sam  
Employee designation: manager  
Employee ID: 1  
Years of experience: 6  
Basic pay: 120000.0  
LIC amount: 3500.0  
DA: 4800.0  
HRA: 12000.0  
PF: 9600.0  
Gross pay: 136800.0  
Net pay: 123700.0

#### PAYMENT SLIP

Employee name: Ram  
Employee designation: INTERN  
Employee ID: 2  
Years of experience: 2  
Basic pay: 4000.0

LIC amount: 0.0  
DA: 2000.0  
HRA: 1000.0  
PF: 500.0  
Gross pay: 6000.0  
Net pay: 5500.0

#### PAYMENT SLIP

Employee name: James  
Employee designation: Trainee  
Employee ID: 3  
Years of experience: 1  
Basic pay: 3500.0  
LIC amount: 80.0  
DA: 105.0  
HRA: 350.0  
PF: 280.0  
Gross pay: 3955.0  
Net pay: 3595.0

DO YOU WANT TO CONTINUE?

ENTER Y OR y TO CONTINUE

y

MENU

1. Input all records
2. Display all records
3. Display payslip for one employee
4. Promote an employee

3

Enter employee ID

2

PAYMENT SLIP

Employee name: Ram

Employee designation: INTERN

Employee ID: 2

Years of experience: 2

Basic pay: 4000.0

LIC amount: 0.0

DA: 2000.0

HRA: 1000.0

PF: 500.0

Gross pay: 6000.0

Net pay: 5500.0

DO YOU WANT TO CONTINUE?

ENTER Y OR y TO CONTINUE

y

MENU

1. Input all records
2. Display all records
3. Display payslip for one employee
4. Promote an employee

3

Enter employee ID

8

EMPLOYEE NOT FOUND!!!

DO YOU WANT TO CONTINUE?

ENTER Y OR y TO CONTINUE

y

MENU

1. Input all records
2. Display all records
3. Display payslip for one employee
4. Promote an employee

4

Enter employee ID

1

Employee name: Sam

Number of years of experience: 6

#### PROMOTING AN EMPLOYEE

CONGRATULATIONS

You are promoted!!!

DO YOU WANT TO CONTINUE?

ENTER Y OR y TO CONTINUE

y

#### MENU

1. Input all records
2. Display all records
3. Display payslip for one employee
4. Promote an employee

4

Enter employee ID

2

Employee name: Ram

Number of years of experience: 2

PROMOTING AN EMPLOYEE

SORRY YOU DONT HAVE ENOUGH EXPERIENCE FOR PROMOTION!!

DO YOU WANT TO CONTINUE?

ENTER Y OR y TO CONTINUE

y

MENU

1. Input all records
2. Display all records
3. Display payslip for one employee
4. Promote an employee

4

Enter employee ID

8

EMPLOYEE NOT FOUND!!!

DO YOU WANT TO CONTINUE?

ENTER Y OR y TO CONTINUE

y

MENU

1. Input all records
2. Display all records
3. Display payslip for one employee
4. Promote an employee

9

INVALID OPTION!!!

DO YOU WANT TO CONTINUE?

ENTER Y OR y TO CONTINUE

n

DONE!!!

**SSN College of Engineering**  
**Department of Computer Science and Engineering**  
**UCS1313 – Object Oriented Programming Using Java Lab**  
**II Year CSE - A, B Section ( III Semester)**  
**Academic Year 2020-21**  
**Batch: 2019- 2023**  
**Faculty Incharge : S.Rajalakshmi / S.Lakshmi Priya**

---

**Exercise 2. Java Programs- Arrays and Strings**

1. Write a java program to get ‘n’ elements in an array. Perform the linear and binary search.
2. Write a java program to find matrix addition, subtraction and multiplication.
3. Write a Java program to get a sentence and find the longest word in it. Also find its index position.
4. Write a Java program to get a string and verify whether it's a Palindrome or not.
5. Write a Java program to check if the given sentence is a Pangram. Pangram string is the string in which all the alphabets should occur atleast once. Also count the number of occurrences of each letter.

Example:

The quick brown fox jumps over the lazy dog

## Exercise 2. Java Programs- Arrays and Strings

### 1. Linear and binary search

Code:

```
import java.util.Scanner;

public class ArraySearch {
    private static Scanner sc = new Scanner(System.in);
    int [] arr;

    void dispArray(int[] arr)
    {
        for(int i=0;i<arr.length;i++)
        {
            System.out.print(arr[i]+\t");
        }
    }

    void sortArray(int arr[])
    {
        for(int i=0;i<arr.length;i++)
        {
            for(int j=i;j<arr.length;j++)
            {
                if(arr[i]>arr[j])
                {
                    int temp=arr[i];
                    arr[i]=arr[j];
                    arr[j]=temp;
                }
            }
        }
        dispArray(arr);
    }

    int searchLinear(int [] arr,int x)
    {
        dispArray(arr);
        for(int i=0;i<arr.length;i++)
        {
            if(arr[i]==x)
            {
                return i;
            }
        }
        return -1;
    }
}
```

```

int searchBinary(int[] arr,int x)
{
    sortArray(arr);
    int high,low,mid;
    high=arr.length-1;
    low=0;
    while(low<high)
    {
        mid=(high+low)/2; //finding middle element
        if(arr[low]==x)
        {
            return low;
        }
        else if(arr[high]==x)
        {
            return high;
        }
        else if(arr[mid]==x)
        {
            return mid;
        }
        else if(arr[mid]<x) //searching in second half of the array
        {
            low=mid;
        }
        else if(arr[mid]>x) //searching in first half of array
        {
            high=mid;
        }
    }
    return -1; //if element isnt found
}

public static void main(String [] args)
{
    int ch=2,x,ls,bs;
    ArraySearch as = new ArraySearch();
    System.out.println("\nEnter size of the array: ");
    int n= sc.nextInt();
    as.arr = new int[n];
    for(int i=0;i<n;i++)
    {
        System.out.print("Enter the element "+(i+1)+": ");
        as.arr[i]=sc.nextInt();
    }
    System.out.print("\nMENU\n1.Linear search\n2.Binary search\nEnter cho
ice: ");
    ch=sc.nextInt();
    if(ch==1)
    {

```

```

        System.out.print("Enter element (linear search): ");
        x=sc.nextInt();
        ls=as.searchLinear(as.arr,x);
        if(ls!=-1)
        {
            System.out.print("\nFound at: "+(ls+1));    //printing the position of the element
        }
        else
        {
            System.out.print("\nNot found\n");
        }
    }
    else if(ch==2)
    {
        System.out.print("\nEnter element (binary search): ");
        x=sc.nextInt();
        bs=as.searchBinary(as.arr,x);
        if(bs!=-1)
        {
            System.out.print("\nFound at: "+(bs+1));
        }
        else
        {
            System.out.print("\nNot found\n");
        }
    }
}
}

```

### Output

a)Enter size of the array:

7

Enter the element 1: 2

Enter the element 2: 3

Enter the element 3: 1

Enter the element 4: 5

Enter the element 5: 6

Enter the element 6: 9

Enter the element 7: 4

MENU

1.Linear search

2.Binary search

Enter choice: 1

Enter element (linear search): 1

2        3        1        5        6        9        4

Found at: 3

b)Enter size of the array:

8

Enter the element 1: 4

Enter the element 2: 5

Enter the element 3: 1

Enter the element 4: 2

Enter the element 5: 9

Enter the element 6: 7

Enter the element 7: 0

Enter the element 8: 3

MENU

1.Linear search

2.Binary search

Enter choice: 2

Enter element (binary search): 0

0        1        2        3        4        5        7        9

Found at: 1

c)Enter size of the array:

6

Enter the element 1: 2

Enter the element 2: 1

Enter the element 3: 5

Enter the element 4: 4

Enter the element 5: 6

Enter the element 6: 3

MENU

1.Linear search

2.Binary search

Enter choice: 1

Enter element (linear search): 9

2        1        5        4        6        3

Not found

PS D:\01-College\assignments\oops\_lab\ex2> java ArraySearch

d)Enter size of the array:

5

Enter the element 1: 7

Enter the element 2: 8

Enter the element 3: 6

Enter the element 4: 4

Enter the element 5: 3

MENU

1.Linear search

2.Binary search

Enter choice: 2

Enter element (binary search): 1

3        4        6        7        8

Not found

ArraySearch
~arr[]:int
~dispArray(arr):void
~sortArray(arr):void
~searchLinear(arr,x):int
~searchBinary(arr,x):int

## 2) Matrix addition, subtraction and multiplication

### Code:

```
import java.util.*;  
  
public class Matrix {  
  
    int arr[][];  
    int r,c;  
  
    private static Scanner sc = new Scanner(System.in);  
    Matrix(int row,int col)  
    {  
        r=row;  
        c=col;  
        arr=new int[r][c];  
    }  
    static void getInput(Matrix a)  
    {  
        System.out.println("INPUT\n");  
        for(int i=0;i<a.r;i++)  
        {  
            for(int j=0;j<a.c;j++)  
            {  
  
                System.out.print("Enter element ["+(i+1)+"]["+ (j+1)+ "]: ");  
                sc.nextLine();  
                a.arr[i][j]=sc.nextInt();  
            }  
        }  
    }  
  
    void initZero()  
    {  
        for(int i=0;i<this.r;i++)  
        {  
            for(int j=0;j<this.c;j++)  
            {  
                this.arr[i][j]=0;  
            }  
        }  
    }  
  
    static void dispMatrix(Matrix a)  
    {  
        for(int i=0;i<a.r;i++)  
        {  
            for(int j=0;j<a.c;j++)  
            {  
                System.out.print(a.arr[i][j] + " ");  
            }  
            System.out.println();  
        }  
    }  
}
```

```

        {
            System.out.print(a.arr[i][j]+\t");
        }
        System.out.println();
    }
}

static void addMatrix(Matrix m,Matrix n,Matrix res)
{
    for(int i=0;i<m.r;i++)
    {
        for(int j=0;j<m.c;j++)
        {
            res.arr[i][j]=m.arr[i][j]+n.arr[i][j];
        }
    }
    dispMatrix(res);
}

static void subMatrix(Matrix m,Matrix n,Matrix res)
{
    for(int i=0;i<m.r;i++)
    {
        for(int j=0;j<m.c;j++)
        {
            res.arr[i][j]=m.arr[i][j]-n.arr[i][j];
        }
    }
    dispMatrix(res);
}

static void mulMatrix(Matrix m,Matrix n,Matrix res)
{
    res.initZero();
    for(int i=0;i<m.r;i++)
    {
        for(int j=0;j<n.c;j++)
        {
            for(int k=0;k<m.c;k++)
            {
                res.arr[i][j]+=m.arr[i][k]*n.arr[k][j];
            }
        }
    }
    dispMatrix(res);
}

public static void main(String []args)
{
    int ch;
}

```

```

char c;
do
{
    System.out.print("\nMENU\n1.Add\n2.Subtract\n3.Multiply\nEnter ch
oice: ");
    ch=sc.nextInt();
    int r1,r2,c1,c2;
    System.out.print("Enter rows for A: ");
    r1=sc.nextInt();
    System.out.print("Enter columns for A: ");
    c1=sc.nextInt();
    System.out.print("Enter rows for B: ");
    r2=sc.nextInt();
    System.out.print("Enter columns for B: ");
    c2=sc.nextInt();
    if(ch==1||ch==2)
    {
        if(r1==r2&&c1==c2)
        {
            Matrix a = new Matrix(r1,c1);
            Matrix b = new Matrix(r2,c2);
            Matrix res = new Matrix(r1,c1);

            getInput(a);
            getInput(b);
            if(ch==1)
            {
                addMatrix(a,b,res);
            }
            else if(ch==2)
            {
                subMatrix(a,b,res);
            }
        }
        else
        {
            System.out.println("invalid input");
        }
    }
    else if(ch==3)
    {
        if(c1==r2)
        {

            Matrix a = new Matrix(r1,c1);
            Matrix b = new Matrix(r2,c2);
            Matrix res = new Matrix(r1,c2);
            getInput(a);
            getInput(b);
        }
    }
}

```

```

        mulMatrix(a,b,res);
    }
    else
    {
        System.out.println("invalid input");
    }

}
else if(ch>3)
{
    break;
}
System.out.print("Do you want to continue?\ny or Y to continue: "
);
c=sc.next().charAt(0);

}while(c=='y' || c=='Y');
}
}

```

**Output:**

MENU

1.Add

2.Subtract

3.Multiply

Enter choice: 1

Enter rows for A: 1

Enter columns for A: 2

Enter rows for B: 2

Enter columns for B: 1

invalid input

Do you want to continue?

y or Y to continue: y

MENU

- 1.Add
- 2.Subtract
- 3.Multiply

Enter choice: 1

Enter rows for A: 2

Enter columns for A: 2

Enter rows for B: 2

Enter columns for B: 2

INPUT

Enter element [1][1]: 1

Enter element [1][2]: 2

Enter element [2][1]: 3

Enter element [2][2]: 4

INPUT

Enter element [1][1]: 1

Enter element [1][2]: 2

Enter element [2][1]: 3

Enter element [2][2]: 4

2        4

6        8

Do you want to continue?

y or Y to continue: y

MENU

1.Add  
2.Subtract  
3.Multiply

Enter choice: 2

Enter rows for A: 1

Enter columns for A: 3

Enter rows for B: 2

Enter columns for B: 4

invalid input

Do you want to continue?

y or Y to continue: y

MENU

1.Add  
2.Subtract  
3.Multiply

Enter choice: 2

Enter rows for A: 3

Enter columns for A: 3

Enter rows for B: 3

Enter columns for B: 3

INPUT

Enter element [1][1]: 1  
Enter element [1][2]: 6  
Enter element [1][3]: 5

Enter element [2][1]: 4

Enter element [2][2]: 8

Enter element [2][3]: 7

Enter element [3][1]: 9

Enter element [3][2]: 6

Enter element [3][3]: 5

INPUT

Enter element [1][1]: 2

Enter element [1][2]: 7

Enter element [1][3]: 5

Enter element [2][1]: 3

Enter element [2][2]: 0

Enter element [2][3]: 9

Enter element [3][1]: 8

Enter element [3][2]: 7

Enter element [3][3]: 6

-1        -1        0

1        8        -2

1        -1        -1

Do you want to continue?

y or Y to continue: y

MENU

1.Add

2.Subtract

3.Multiply

Enter choice: 3

Enter rows for A: 1

Enter columns for A: 2

Enter rows for B: 1

Enter columns for B: 2

invalid input

Do you want to continue?

y or Y to continue: y

MENU

1.Add

2.Subtract

3.Multiply

Enter choice: 1

Enter rows for A: 2

Enter columns for A: 2

Enter rows for B: 1

Enter columns for B: 6

invalid input

Do you want to continue?

y or Y to continue: y

MENU

1.Add

2.Subtract

3.Multiply

Enter choice: 3

Enter rows for A: 2

Enter columns for A: 3

Enter rows for B: 3

Enter columns for B: 4

INPUT

Enter element [1][1]: 1

Enter element [1][2]: 3

Enter element [1][3]: 2

Enter element [2][1]: 5

Enter element [2][2]: 6

Enter element [2][3]: 4

INPUT

Enter element [1][1]: 6

Enter element [1][2]: 5

Enter element [1][3]: 4

Enter element [1][4]: 3

Enter element [2][1]: 2

Enter element [2][2]: 7

Enter element [2][3]: 3

Enter element [2][4]: 5

Enter element [3][1]: 3

Enter element [3][2]: 2

Enter element [3][3]: 6

Enter element [3][4]: 5

18        30        25        28

54        75        62        65

Do you want to continue?

y or Y to continue: n

Matrix
~arr[][]:int
~r:int
~c:int
+Matrix(int row,int col)
~getInput(Matrix a):void
~initZero():void
~dispMatrix(Matrix a):void
~addMatrix(Matrix m,Matrix n,Matrix res):void
~subMatrix(Matrix m,Matrix n,Matrix res):void
~mulMatrix(Matrix m,Matrix n,Matrix res):void

### 3) Longest word in a sentence and its position

Code:

```
import java.util.*;  
  
class Longest {  
    private static Scanner sc = new Scanner(System.in);  
    String word[],longword;  
    int count[],wc,pos;  
  
    Longest(String s)  
    {  
        wc=1;  
        for(int i=0;i<s.length();i++)  
        {  
            if(s.charAt(i)==' ')  
            {  
                wc++;  
            }  
        }  
        word=new String[wc+1];  
        count=new int[wc+1];  
        for(int i=0;i<wc;i++)  
        {  
            count[i]=0;  
            word[i]=" ";  
        }  
    }  
  
    void wordCount(String s)  
    {  
        int j=0;  
        for(int i=0;i<s.length();i++)  
        {  
            if(s.charAt(i)!=' ')  
            {  
                word[j]+=s.charAt(i);  
                count[j]++;  
            }  
            else  
            {  
                word[j]+='\0';  
                j++;  
            }  
        }  
    }  
}
```

```

int longWord(String s)
{
    wordCount(s);
    int i=0,max=0,pos=0;
    for(i=0;i<wc;i++)
    {
        if(count[i]>max)
        {
            max=count[i];
            pos=i;
        }
    }
    return pos;
}

public static void main(String args[])
{
    int pos;
    String sent;
    System.out.print("\nEnter sentence: ");
    sent=sc.nextLine();
    Longest l = new Longest(sent);
    pos=l.longWord(sent);
    System.out.print("\nLongest word is"+l.word[pos]+" and position is: "
+pos);
}
}

```

**Output:**

a) Enter sentence: I am a Computer Science student

Longest word is Computer and position is: 3

b) Enter sentence: I am good

Longest word is good and position is: 2

c) Enter sentence: World is big

Longest word is World and position is: 0

<b>Longest</b>
~count[]:int ~longword:String ~word[]:String ~wc:int ~pos:int
+Longest(String s) ~wordCount(String s):void ~longWord(String s):int

#### 4) Palindrome

##### Code:

```

import java.util.*;

public class Palindrome {
    private static Scanner sc = new Scanner(System.in);

    String s;
    int checkPalindrome()
    {
        for(int i=0;i<(s.length()/2);i++)
        {
            int j= s.length()-1-i;
            if(s.charAt(i)!=s.charAt(j))
            {
                return 0;
            }
        }
        return 1;
    }

    public static void main(String []args)
    {
        Palindrome p = new Palindrome();
        System.out.println("Enter a string: ");
        p.s=sc.nextLine();
        int f=p.checkPalindrome();
        if(f==1)
        {
            System.out.println("PALINDROME");
        }
        else
        {
            System.out.println("NOT PALINDROME");
        }
    }
}

```

```
    }  
}  
}
```

**Output:**

a) Enter a string:

shreya

NOT PALINDROME

b) Enter a string:

malayalam

PALINDROME

c) Enter a string:

2002

PALINDROME

Palindrome
~s:String
~checkPalindrome():int

## 5) Pangram

### Code:

```
import java.util.*;
public class Pangram {

    private static Scanner sc = new Scanner(System.in);
    String s;
    int count[];
    Pangram()
    {
        this.count=new int[26];
        for(int i=0;i<26;i++)
        {
            count[i]=0;
        }
    }

    void freqCount()
    {
        for(int i=0;i<this.s.length();i++)
        {
            if(s.charAt(i)!=' ')
            {
                char c=s.charAt(i);
                count[c-65]+=1;
            }
        }
    }

    void displayFreq()
    {
        for(int i=0;i<26;i++)
        {
            char c=(char)('a'+i);
            System.out.println("Letter: "+c+"\tCount: "+count[i]);
        }
    }

    int checkPangram()
    {
        freqCount();
        for(int i=0;i<26;i++)
        {
            if(count[i]==0)
            {
                return 0;
            }
        }
    }
}
```

```

        return 1;
    }

    public static void main(String[] args)
    {
        Pangram p = new Pangram();
        System.out.println("Enter sentence: ");
        p.s=sc.nextLine().toUpperCase();
        int f=p.checkPangram();
        p.displayFreq();
        if(f==1)
        {
            System.out.println("PANGRAM");
        }
        else
        {
            System.out.println("NOT PANGRAM");
        }
    }
}

```

**Output:**

a) Enter sentence:

the quick brown fox jumps over the lazy dog

Letter: a      Count: 1

Letter: b      Count: 1

Letter: c      Count: 1

Letter: d      Count: 1

Letter: e      Count: 3

Letter: f      Count: 1

Letter: g      Count: 1

Letter: h      Count: 2

Letter: i      Count: 1

Letter: j      Count: 1

Letter: k      Count: 1

Letter: l      Count: 1

Letter: m	Count: 1
Letter: n	Count: 1
Letter: o	Count: 4
Letter: p	Count: 1
Letter: q	Count: 1
Letter: r	Count: 2
Letter: s	Count: 1
Letter: t	Count: 2
Letter: u	Count: 2
Letter: v	Count: 1
Letter: w	Count: 1
Letter: x	Count: 1
Letter: y	Count: 1
Letter: z	Count: 1

PANGRAM

b) Enter sentence:

hello world we are learning computer science

Letter: a	Count: 2
Letter: b	Count: 0
Letter: c	Count: 3
Letter: d	Count: 1
Letter: e	Count: 7
Letter: f	Count: 0
Letter: g	Count: 1
Letter: h	Count: 1
Letter: i	Count: 2

Letter: j        Count: 0  
Letter: k        Count: 0  
Letter: l        Count: 4  
Letter: m        Count: 1  
Letter: n        Count: 3  
Letter: o        Count: 3  
Letter: p        Count: 1  
Letter: q        Count: 0  
Letter: r        Count: 4  
Letter: s        Count: 1  
Letter: t        Count: 1  
Letter: u        Count: 1  
Letter: v        Count: 0  
Letter: w        Count: 2  
Letter: x        Count: 0  
Letter: y        Count: 0  
Letter: z        Count: 0

NOT PANGRAM

Pangram
~s:String
~count[][]:int
+Pangram()
~freqCount():void
~displayFreq():void
~checkPangram():int

**SSN College of Engineering**  
**Department of Computer Science and Engineering**  
**UCS1313 – Object Oriented Programming Using Java Lab**  
**II Year CSE - A, B Section ( III Semester)**  
**Academic Year 2020-21**  
**Batch: 2019- 2023**  
**Faculty Incharge : S.Rajalakshmi / S.Lakshmi Priya**

---

**Exercise 3. Inheritance**

**Objective:**

1. To test the following Inheritance types: single-level, multi-level and hierarchical inheritance.
2. To test the scope of private and protected variables, constructors in inherited class hierarchy.

**Sample Learning Outcome:**

1. Need of inheritance and its implementation in Java
2. Type of inheritance
3. Working of constructors in inherited class
4. Accessing inherited class through base class reference
5. Method overloading and overriding in inheritance

**Best Practices:**

1. Class Diagram usage
2. Naming convention – for file names, variables
3. Comment usage at proper places
4. Prompt messages during reading input and displaying output
5. Incremental program development
6. Modularity
7. All possible test cases in output

I) Create a class hierarchy for the classes defined below: Design a class called Person as described below:

- Private members
- + Public members
- # Protected members
- ~ Default (Package private)

Person
-aadhaar:int
-name:String
-address:String
-gender:char
+Person(aadhaar,name,address,gender)
+getName():String
+getAddress():String
+setAddress(address):void
+getGender():char

A sub-class Student of class Person is designed as shown below:

Student
-program:String
-year:int
-totalmark:float
+Student(aadhaar,name,address,gender,program,year,total)
+getProgram():String
+getYear():int
+setYear(year):void
+getTotal():float
+setTotal(tot):void
+calGPA():float

A sub-class Faculty of class Person is designed as shown below:

	Faculty
-designation:String	
-department:String	
-basicpay:float	
+Faculty(aadhaar,name,address,gender,designation,dept,pay)	
+getDesig():String	
+setDesig(desig):void	
+setBasic(basic):void	
+getBasic():float	
+calSalary():float	

Note the following:

1. The hierarchy Person -> Student (or) Person -> Faculty is a ***Single-level inheritance*** type.
  2. The type of above entire class hierarchy (Person -> Student , Person -> Faculty) is the ***Hierarchical Inheritance***.
  3. Note the use of constructors at all levels of class hierarchy.

## EXERCISE : I)

1. Draw the class diagram of the above class hierarchy.
  2. Write a *test driver* called `TestInheritance` to test all the `public` methods that display the student and faculty details.

Use the following to calculate Net Salary:

Gross salary = Basicpay + DA as 60% of basic + HRA as 10% of basic

Deductions = Medical Insurance as 8.5% of basic + PF as 8% of basic

**Net salary = Gross salary – Deductions**

**II) Create a class hierarchy for the classes/interface as defined below: Design a class Shape as described below: # - *protected***

Shape
#color:String="red"
+Shape()
+Shape(color)
+getColor():String
+setColor(color):void

A sub-class **Circle** of class *Shape* is designed as shown below:

Circle
#radius:float=1.0
+Circle()
+Circle(radius)
+Circle(radius,color)
+getRadius():float
+setRadius(radius):void
+getArea():float
+getPerimeter():float

A sub-class **Rectangle** of class *Shape* is designed as shown below:

Rectangle
#width:float=1.0
#length:float=1.0
+Rectangle()
+Rectangle(width,length)
+Rectangle(width,length,color)
+getWidth():float
+setWidth(width):void
+getLength():float
+setLength(length):void
+getArea():float
+getPerimeter():float

A sub-class **Square** of class *Rectangle* is designed as shown below:

Square
+Square()
+Square(side)
+Square(side,color)
+getSide():float
+setSide(side):void

Note the following:

1. The hierarchy Shape --> Rectangle --> Square is a ***Multi-level inheritance*** type.
2. The type of above entire class hierarchy is the ***Hierarchical Inheritance***.
3. Note the constructor overloading at all the levels.
4. # denotes **protected** variable. The **protected** variables can be accessed by its subclasses and classes in the same package.

EXERCISE : II)

1. Draw the class diagram of the above class hierarchy.
2. Write a *test driver* called `TestShape` to test all the `public` methods. Use an array of objects of type *Shape* and display the area and perimeter of all the shapes (Circle, Rectangle and Square).
3. Note down the scope of the variable declared as **protected**.

#####\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$#\$\$\$\$\$\$\$\$#\$\$\$\$#\$\$\$\$#\$\$\$\$#\$\$\$\$#\$\$\$\$#\$\$\$\$#\$\$\$\$#

## EXERCISE 3- INHERITANCE

### Code:

```
import java.util.*;
class Person
{
    private int aadhaar;
    private String name;
    private String address;
    private char gender;

    Person(){}
    Person(int aadhaar, String name, String address, char gender)
    {
        this.aadhaar=aadhaar;
        this.name=name;
        this.address=address;
        this.gender=gender;
    }

    public String getName()
    {
        return this.name;
    }

    public String getAddress()
    {
        return this.address;
    }

    public char getGender()
    {
        return this.gender;
    }

    public void setAddress(String address)
    {
        this.address=address;
    }

    public void display()
    {
        System.out.println("Aadhaar: "+this.aadhaar);
        System.out.println("Name: "+this.name);
        System.out.println("Address: "+this.address);
        System.out.println("Gender: "+this.gender);
    }
}
```

```
        }
    }

class Student extends Person
{
    private
        String program;
        int year;
        float totalmark;

    public
        Student(int aadhaar, String name, String address, char gender, String program
        ,int year, float total)
    {
        super(aadhaar, name, address, gender);
        this.program=program;
        this.year=year;
        this.totalmark=total;
    }

    public String getProgram()
    {
        return this.program;
    }

    public int getYear()
    {
        return this.year;
    }

    public void setYear(int year)
    {
        this.year=year;
    }

    public float getTotal()
    {
        return this.totalmark;
    }

    public void setTotal(float tot)
    {
        totalmark=tot;
    }

    public float calGPA()
    {
        return totalmark/10;
    }
}
```

```
public void display()
{
    super.display();
    System.out.println("Program: "+this.program);
    System.out.println("Year of joining: "+this.year);
    System.out.println("Total marks: "+this.totalmark);
}
}

class Faculty extends Person
{
    private
    String designation;
    String department;
    float basicpay;

    public
    Faculty(int aadhaar,String name,String address,char gender,String designation,String dept,float pay)
    {
        super(aadhaar,name,address,gender);
        this.designation=designation;
        this.department=dept;
        this.basicpay=pay;
    }

    public String getDesig()
    {
        return designation;
    }

    public void setDesig(String desig)
    {
        designation=desig;
    }

    public void setBasic(float basic)
    {
        basicpay=basic;
    }

    public float getBasic()
    {
        return basicpay;
    }

    public float calSalary()
    {
        float gp,np;
        gp=(float)(basicpay+0.06*basicpay+0.10*basicpay);
```

```

        np=(float)(gp-0.085*basicpay-0.08*basicpay);
        return np;
    }

    public void display()
    {
        super.display();
        System.out.println("Designation: "+this.designation);
        System.out.println("Department: "+this.department);
        System.out.println("Basic pay: "+this.basicpay);
    }
}

class TestInheritance
{
    public static void main(String []args)
    {
        Scanner sc = new Scanner(System.in);
        int ch,aad;
        String name,add;
        char g;
        System.out.println("1.Student\n2.Faculty");
        ch=sc.nextInt();
        System.out.print("Enter aadhaar number: ");
        aad=sc.nextInt();
        sc.nextLine();
        System.out.print("Enter name: ");
        name=sc.nextLine();
        System.out.print("Enter address: ");
        add=sc.nextLine();
        System.out.print("Enter gender: ");
        g=sc.next().charAt(0);
        sc.nextLine();
        if(ch==1)
        {
            String prog;
            float tot;
            int yr;
            System.out.print("Enter program: ");
            prog=sc.nextLine();
            System.out.print("Enter year: ");
            yr=sc.nextInt();
            System.out.print("Enter total marks: ");
            tot=sc.nextFloat();
            Student s = new Student(aad,name,add,g,prog,yr,tot);
            System.out.println("\tTesting the get functions in Student class"
);
            System.out.println("Name: "+s.getName());
            System.out.println("Address: "+s.getAddress());
            System.out.println("Gender: "+ s.getGender());
        }
    }
}
```

```

        System.out.println("Program: "+s.getProgram());
        System.out.println("Year of joining: "+s.getYear());
        System.out.println("Calculating the CGPA: CGPA: "+s.calGPA());
        System.out.println("\tTesting set methods in Student class");
        sc.nextLine();
        System.out.print("Enter new address: ");
        s.setAddress(sc.nextLine());
        System.out.print("Enter new total: ");
        s.setTotal(sc.nextFloat());
        System.out.print("Enter new year: ");
        s.setYear(sc.nextInt());
        s.display();
        System.out.println("Calculating the new CGPA: CGPA: "+s.calGPA())
    ;
}
else if(ch==2)
{
    String dep,desig;
    float bp;
    System.out.print("Enter department: ");
    dep=sc.nextLine();
    System.out.print("Enter designation: ");
    desig=sc.nextLine();
    System.out.print("Enter basic pay: ");
    bp=sc.nextFloat();
    Faculty f = new Faculty(aad,name,add,g,desig,dep,bp);
    System.out.println("\tTesting the get functions in Student class"
);
    System.out.println("Name: "+f.getName());
    System.out.println("Address: "+f.getAddress());
    System.out.println("Gender: "+ f.getGender());
    System.out.println("Designation: "+f.getDesig());
    System.out.println("Calculating the Net Pay: Net pay: "+f.calSalary());
    System.out.println("\tTesting set methods in Student class");
    System.out.print("Enter new address: ");
    f.setAddress(sc.nextLine());
    sc.nextLine();
    System.out.print("Enter new basic pay: ");
    f.setBasic(sc.nextFloat());
    System.out.print("Enter new designation: ");
    f.setDesig(sc.nextLine());
    f.display();
    System.out.println("Calculating the new net pay: Net pay: "+f.calSalary());
}
sc.close();
}
}

```

**Output:**

1.Student

2.Faculty

1

Enter aadhaar number: 134222

Enter name: Shreya

Enter address: Ashok Pillar

Enter gender: F

Enter program: CSE

Enter year: 2019

Enter total marks: 98

Testing the get functions in Student class

Name: Shreya

Address: Ashok Pillar

Gender: F

Program: CSE

Year of joining: 2019

Calculating the CGPA: CGPA: 9.8

Testing set methods in Student class

Enter new address: Triplicane

Enter new total: 94

Enter new year: 2020

Aadhaar: 134222

Name: Shreya

Address: Triplicane

Gender: F

Program: CSE

Year of joining: 2020

Total marks: 94.0

Calculating the new CGPA: CGPA: 9.4

b)

1.Student

2.Faculty

2

Enter aadhaar number: 1455526

Enter name: Sriram

Enter address: TNagar

Enter gender: M

Enter department: ECE

Enter designation: Professor

Enter basic pay: 90000

Testing the get functions in Student class

Name: Sriram

Address: TNagar

Gender: M

Designation: Professor

Calculating the Net Pay: Net pay: 89550.0

Testing set methods in Student class

Enter new address: Kalavakkam

Enter new basic pay: 80000

Enter new designation: Aadhaar: 1455526

Name: Sriram

Address:

Gender: M

Designation:

Department: ECE

Basic pay: 80000.0

Calculating the new net pay: Net pay: 79600.0

**Class Diagram:**

Person
<pre>-aadhaar:int -name:String -address:String -gender:char</pre>
<pre>+Person(aadhaar,name,address,gender) +getName():String +getAddress():String +setAddress(address):void +getGender():char +display():void</pre>

### Student

```
-program:String  
-year:int  
-totalmark:float  
  
+Student(aadhaar,name,address,gender,program,year,total)  
+getProgram():String  
+getYear():int  
+setYear(year):void  
+getTotal():float  
+setTotal(tot):void  
+calGPA():float  
+display():void
```

### Faculty

```
-designation:String  
-department:String  
-basicpay:float  
  
+Faculty(aadhaar,name,address,gender,designation,dept,pay)  
+getDesig():String  
+setDesig(desig):void  
+setBasic(basic):void  
+getBasic():float  
+calSalary():float  
+display():void
```

**Code:**

```
import java.util.*;
class Shape {
    protected String color;

    Shape()
    {
        this.color="red";
    }

    Shape(String color)
    {
        this.color=color;
    }

    public String getColor()
    {
        return color;
    }

    public void setColor(String color)
    {
        this.color=color;
    }
}

class Circle extends Shape
{
    protected float radius;

    Circle()
    {
        this.radius=(float)1.0;
    }

    Circle(float radius)
    {
        this.radius=radius;
    }

    Circle(float radius, String color)
    {
        super(color);
        this.radius=radius;
    }

    public float getRadius()
    {
        return this.radius;
```

```
}

public void setRadius(float radius)
{
    this.radius=radius;
}

public float getArea()
{
    return (float)Math.PI*this.radius*this.radius;
}

public float getPerimeter()
{
    return (float)Math.PI*2*this.radius;
}

public void display()
{
    System.out.println("Radius: "+radius);
    System.out.println("Colour: "+color);
}

}

class Rectangle extends Shape
{

protected float width,length;//=1.0

public
Rectangle()
{
    this.width=(float)1.0;
    this.length=(float)1.0;
}

Rectangle(float width,float Length)
{
    this.width=width;
    this.length=length;
}

Rectangle(String color)
{
    super(color);
}

Rectangle(float width,float Length,String color)
```

```
{  
    this(color);  
    this.width=width;  
    this.length=length;  
}  
  
public float getWidth()  
{  
    return width;  
}  
  
public void setWidth(float width)  
{  
    this.width=width;  
}  
  
public float getLength()  
{  
    return this.length;  
}  
  
public void setLength(float length)  
{  
    this.length=length;  
}  
  
public float getArea()  
{  
    return (float)length*width;  
}  
  
public float getPerimeter()  
{  
    return (float)2*(length+width);  
}  
  
public void display()  
{  
    System.out.println("Length: "+length);  
    System.out.println("Width: "+width);  
    System.out.println("Colour: "+color);  
}  
}  
  
class Square extends Rectangle  
{  
    public float side;  
  
    Square(){  
    Square(float side)
```

```

{
    this.side=side;
}

Square(float side,String color)
{
    super(color);
    this.side=side;
}

public float getSide()
{
    return this.side;
}

public void setSide(float side)
{
    this.side=side;
}

public void display()
{
    System.out.println("Side: "+side);
    System.out.println("Colour: "+color);
}

}

class TestShape
{
    public static void main(String []args)
    {
        Scanner sc= new Scanner(System.in);
        System.out.print("Enter number of shapes: ");
        int n,ch;
        String colour;
        n=sc.nextInt();
        for(int i=0;i<n;i++)
        {
            System.out.print("MENU\n1.Circle\n2.Rectangle\n3.Square\nEnter option");
            ch=sc.nextInt();
            switch(ch)
            {
                case 1:
                    Circle c1 = new Circle();
                    System.out.println("\n\tTesting get functions");
                    System.out.println("Radius: "+c1.getRadius());
                    System.out.println("Colour: "+c1.getColor());
                    System.out.print("Enter radius: ");
            }
        }
    }
}

```

```

        float radius=sc.nextFloat();
        sc.nextLine();
        System.out.print("Enter colour: ");
        colour=sc.nextLine();
        Circle c2 = new Circle(radius);
        System.out.println("\n\tTesting set functions");
        System.out.print("Enter new colour: ");
        c2.setColor(sc.nextLine());
        System.out.print("Enter new radius: ");
        c2.setRadius(sc.nextFloat());
        c2.display();
        Circle c3 = new Circle(radius,colour);
        System.out.println("\n\tCalculating area and perimeter");
        System.out.println("Area: "+c3.getArea()+"\nPerimeter: "+c3.getPerimeter());
    break;

    case 2:
        Rectangle r1 = new Rectangle();
        System.out.println("\n\tTesting get functions");
        System.out.println("Length: "+r1.getLength());
        System.out.println("Width: "+r1.getWidth());
        System.out.println("Colour: "+r1.getColor());
        System.out.print("Enter length: ");
        float length=sc.nextFloat();
        System.out.print("Enter width: ");
        float width=sc.nextFloat();
        sc.nextLine();
        System.out.print("Enter colour: ");
        colour=sc.nextLine();
        Rectangle r2 = new Rectangle(length,width);
        System.out.println("\n\tTesting set functions");
        System.out.print("Enter new colour: ");
        r2.setColor(sc.nextLine());
        System.out.print("Enter new length: ");
        r2.setLength(sc.nextFloat());
        System.out.print("Enter new width: ");
        r2.setWidth(sc.nextFloat());
        r2.display();
        Rectangle r3 = new Rectangle(length,width,colour);
        System.out.println("\n\tCalculating area and perimeter");
        System.out.println("Area: "+r3.getArea()+"\nPerimeter: "+r3.getPerimeter());
    break;

    case 3:
        Square s1 = new Square();
        System.out.println("\n\tTesting get functions");
        System.out.println("Side: "+s1.getSide());
        System.out.println("Colour: "+s1.getColor());

```

```

        System.out.print("Enter side: ");
        float side=sc.nextFloat();
        sc.nextLine();
        System.out.print("Enter colour: ");
        colour=sc.nextLine();
        Square s2 = new Square(side);
        System.out.println("\n\tTesting set functions");
        System.out.print("Enter new colour: ");
        s2.setColor(sc.nextLine());
        System.out.print("Enter new side: ");
        s2.setSide(sc.nextFloat());
        s2.display();
        Circle s3 = new Circle(side,colour);
        System.out.println("\n\tCalculating area and perimeter");
        System.out.println("Area: "+s3.getArea()+"\nPerimeter: "+s3.getPerimeter());
    break;

    default:
        System.out.println("Invalid!!!");
        i--;
        break;
    }
}
sc.close();
}
}

```

**Output:**

Enter number of shapes: 3

MENU

1.Circle

2.Rectangle

3.Square

Enter option1

Testing get functions

Radius: 1.0

Colour: red

Enter radius: 3

Enter colour: yellow

#### Testing set functions

Enter new colour: pink

Enter new radius: 10

Radius: 10.0

Colour: pink

#### Calculating area and perimeter

Area: 28.274334

Perimeter: 18.849556

MENU

1.Circle

2.Rectangle

3.Square

Enter option4

Invalid!!

MENU

1.Circle

2.Rectangle

3.Square

Enter option2

#### Testing get functions

Length: 1.0

Width: 1.0

Colour: red

Enter length: 4

Enter width: 5

Enter colour: blue

#### Testing set functions

Enter new colour: grey

Enter new length: 6

Enter new width: 4

Length: 6.0

Width: 4.0

Colour: grey

#### Calculating area and perimeter

Area: 20.0

Perimeter: 18.0

MENU

1.Circle

2.Rectangle

3.Square

Enter option3

#### Testing get functions

Side: 0.0

Colour: red

Enter side: 5

Enter colour: blue

#### Testing set functions

Enter new colour: green

Enter new side: 4

Side: 4.0

Colour: green

#### Calculating area and perimeter

Area: 78.53982

Perimeter: 31.415928

### Class Diagrams

Shape
#color:String="red"
+Shape() +Shape(color) +getColor():String +setColor(color):void

Circle
#radius:float=1.0
+Circle() +Circle(radius) +Circle(radius,color) +getRadius():float +setRadius(radius):void +getArea():float +getPerimeter():float +display():void

Rectangle
#width:float=1.0 #length:float=1.0
+Rectangle() +Rectangle(width,length) +Rectangle(width,length,color) +getWidth():float +setWidth(width):void +getLength():float +setLength(length):void +getArea():float +getPerimeter():float +display():void

Square
+side:float
+Square() +Square(side) +Square(side,color) +getSide():float +setSide(side):void +display():void

**SSN College of Engineering**  
**Department of Computer Science and Engineering**  
**UCS1313 – Object Oriented Programming Using Java Lab**  
**II Year CSE - A, B Section ( III Semester)**  
**Academic Year 2020-21**  
**Batch: 2019- 2023**  
**Faculty Incharge : S.Rajalakshmi / S.Lakshmi Priya**

---

### **Exercise – 4 – Abstract class and Interfaces**

#### **Objective:**

1. To test the following Inheritance type: multiple inheritance.
  2. To test the Polymorphism through Interface / abstract classes by method overriding.
- 

#### **Sample Learning Outcome:**

1. Need of interface and it's implementation in Java
2. Need of abstract class and it's implementation in Java
3. Multiple inheritance
4. Accessing the derived class objects through base class/interface reference – Dynamic method dispatch/Dynamic binding

#### **Best Practices:**

1. Class Diagram usage
2. Naming convention – for file names, variables
3. Comment usage at proper places
4. Prompt messages during reading input and displaying output
5. Incremental program development
6. Modularity
7. All possible test cases in output

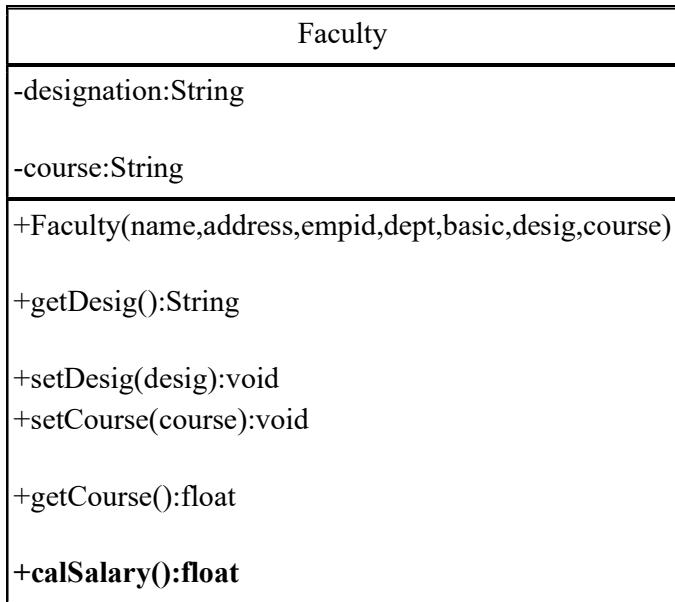
1. Design a class called **Person** as described below:

Person
-name:String
-address:String
+Person(name,address)
+getName():String
+getAddress():String
+setAddress(address):void

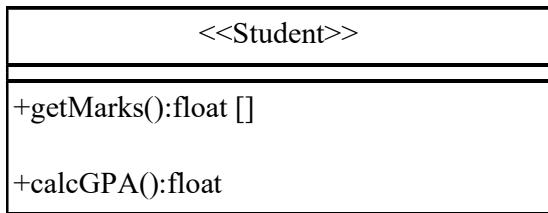
A sub-class Employee of class Person is designed as shown below:

Employee
-empid:String
-dept:String
-basic:int
+Employee(name,address,empid,dept,basic)
+getEmpid():int
+getDept():String
+setDept(dept):void
+setBasic(basic):void
+getBasic():int
+calSalary():float

A sub-class Faculty of class Employee is designed as shown below:



Design an Interface Student:



Design a sub-class TeachingAssitant of class Employee, implements <<Student>>

TeachingAssitant
-project:String
-course:String
-marks:float []
+TeachingAssitant(name,address,empid,dept,basic,project,course,marks)
+getProject():String
+getCourse():String
+setCourse(course):void
+getMarks():float []
+calcGPA():float
+calSalary():float

*Write a TestDriver function to get input for Faculty and TeachingAssistant and display their details. Find the class that can be kept as abstract.*

---

II) Create a class hierarchy for the following using Interface / Abstract class:

Design **Shape** as described below:

Shape
#color:String="red"
+Shape()
+Shape(color)
+getColor():String
+setColor(color):void
<i>abs getArea():float</i>
<i>abs getPerimeter():float</i>

Where *abs* – abstract method

A sub-class **Circle** of class *Shape* is designed as shown below:

Circle
#radius:float=1.0
+Circle()
+Circle(radius)
+Circle(radius,color)
+getRadius():float
+setRadius(radius):void
<b>+getArea():float</b>
<b>+getPerimeter():float</b>

A sub-class **Rectangle** of class *Shape* is designed as shown below:

Rectangle
#width:float=1.0
#length:float=1.0
+Rectangle()
+Rectangle(width,length)
+Rectangle(width,length,color)
+getWidth():float
+setWidth(width):void
+getLength():float
+setLength(length):void
<b>+getArea():float</b>
<b>+getPerimeter():float</b>

A sub-class **Square** of class *rectangle* designed as shown below (Square is one where the length and width of rectangle are same):

Square
+Square()
+Square(side)
+Square(side,color)
+getSide():float
+setSide(side):void
<b>+getArea():float</b>
<b>+getPerimeter():float</b>

Note the following:

1. Shape contains the abstract methods.
2. Those abstract methods are to be implemented by the defining classes.

EXERCISE :

1. Draw the class diagram of the above class hierarchy.
2. Implement the above class hierarchy by using Interface and Abstract class.

**Hint:**

To write an Interface:

- a. Only abstract methods can be declared inside the Interface.
- b. Identify the common behavior of the set of objects and declare that as abstract methods inside the Interface.
- c. The classes that implements the Interface will provide the actual implementation of those abstract methods.

To write an Abstract class:

- a. An abstract class can have constructor(s), abstract or non-abstract method(s).
- b. Define the constructors and non-abstract method in the Abstract class Shape.  
Declare the common behavior as the abstract method.
- c. Let the classes Rectangle, Circle, Square define its own constructors, member variable and methods.

3. Write a *test driver* called `TestInterface` | `TestAbstract`. Use an array of objects of type Shape to display the area, perimeter of all the shapes (Circle, Rectangle, Square).

4. Note down the differences while implementing the Inheritance through Interface and Abstract class.

5. Note the run-time polymorphism in resolving the method call exhibited by Java through method overriding.

```
#####$$$$$$$$$$$$$$$$$$$$$$$$$$$$#$$$$$$$$$$$$$$$$#  
#
```

## Ex-4 abstract classes and interfaces

1)

Code:

```
import java.util.*;
class Person
{
    private String name,address;
    public Person(String name,String address)
    {
        this.name=name;
        this.address=address;
    }

    public String getName()
    {
        return this.name;
    }

    public String getAddress()
    {
        return this.address;
    }

    public void setAddress(String address)
    {
        this.address=address;
    }
}

abstract class Employee extends Person
{
    Scanner sc = new Scanner(System.in);
    private String empid,dept;
    private int basic;
    public Employee(String name,String address,String empid,String dept,int basic)
    {
        super(name,address);
        this.empid=empid;
        this.dept=dept;
        this.basic=basic;
    }
    public String getEmpid()
    {
        return empid;
    }
    public String getDept()
```

```

{
    return dept;
}
public void setDept(String dept)
{
    this.dept=dept;
}
public void setBasic(int basic)
{
    this.basic=basic;
}
public int getBasic()
{
    return basic;
}
abstract public float calSalary();
}

class Faculty extends Employee
{

private String designation,course;
public Faculty(String name,String address,String empid,String dept,int basic,String desig,String course)
{
    super(name,address,empid,dept,basic);
    designation=desig;
    this.course=course;
}
public String getDesig()
{
    return designation;
}
public void setDesig(String desig)
{
    designation=desig;
}
public void setCourse(String course)
{
    this.course=course;
}
public String getCourse()
{
    return course;
}
public float calSalary()
{
    int basic = getBasic();
    return (float)(getBasic()+0.10*basic-2000);
}

```

```

}

interface Student
{
    public float[] getMarks();
    public float calcGPA();
}

class TeachingAssistant extends Employee implements Student
{
    private String project, course;
    private float[] marks;
    public TeachingAssistant(String name, String add, String eid, String dept, int basic, String proj, String course, float[] marks)
    {
        super(name, add, eid, dept, basic);
        project=proj;
        this.course=course;
        this.marks=marks;
    }
    public String getProject()
    {
        return project;
    }

    public String getCourse()
    {
        return course;
    }

    public void setCourse(String course)
    {
        this.course=course;
    }

    public float [] getMarks()
    {
        return marks;
    }

    public float calcGPA()
    {
        int tot=0;
        float []marks = this.getMarks();
        for(int i=0;i<3;i++)
        {
            tot+=marks[i];
        }
        return (float)(tot/30.0);
    }
}

```

```

public float calSalary()
{
    return (float)(getBasic()*0.05+getBasic()-1000);
}
}

class TestPerson{
    private static Scanner sc = new Scanner(System.in);
    public static void main(String args[])
    {
        System.out.println("MENU\n1.Faculty\n2.Teaching assistant\nEnter option");
        int n= sc.nextInt();
        if(n==1)
        {
            System.out.println("Enter details:");
            sc.nextLine();
            System.out.print("Enter name:");
            String name=sc.nextLine();
            System.out.print("Enter address:");
            String add=sc.nextLine();
            System.out.print("Enter empid:");
            String empid=sc.nextLine();
            System.out.print("Enter department:");
            String dept=sc.nextLine();
            System.out.print("Enter basic pay:");
            int bp=sc.nextInt();
            sc.nextLine();
            System.out.print("Enter designation:");
            String des=sc.nextLine();
            System.out.print("Enter course:");
            String cou=sc.nextLine();
            Faculty f=new Faculty(name,add,empid,dept,bp,des,cou);
            System.out.println("NAME: "+f.getName());
            System.out.println("ADDRESS: "+f.getAddress());
            System.out.println("EMPID: "+f.getEmpid());
            System.out.println("DEPARTMENT: "+f.getDept());
            System.out.println("BASIC PAY: "+f.getBasic());
            System.out.println("DESIGNATION: "+f.getDesig());
            System.out.println("COURSE: "+f.getCourse());
            System.out.println("SALARY: "+f.calSalary());
            System.out.println("Do you want to change values?\n1.No\n2.Address\n3.Dept\n4.Designation\n5.Course");
            int ch =sc.nextInt();
            if(ch==1)
            {
                return;
            }
        }
    }
}

```

```

        else if(ch==2)
    {
        System.out.println("Changed value: ");
        sc.nextLine();
        add= sc.nextLine();
        f.setAddress(add);
        System.out.println("New value: "+f.getAddress());
    }
    else if(ch==3)
    {
        System.out.println("Changed value: ");
        sc.nextLine();
        dept= sc.nextLine();
        f.setDept(dept);
        System.out.println("New value: "+f.getDept());
    }
    else if(ch==4)
    {
        System.out.println("Changed value: ");
        sc.nextLine();
        des= sc.nextLine();
        f.setDesig(des);
        System.out.println("New value: "+f.getDesig());
    }
    else if(ch==5)
    {
        System.out.println("Changed value: ");
        sc.nextLine();
        cou= sc.nextLine();
        f.setCourse(cou);
        System.out.println("New value: "+f.getCourse());
    }

}
else if(n==2)
{
    float[] marks=new float [3];
    System.out.println("Enter details:");
    sc.nextLine();
    System.out.print("Enter name:");
    String name=sc.nextLine();
    System.out.print("Enter address:");
    String add=sc.nextLine();
    System.out.print("Enter empid:");
    String empid=sc.nextLine();
    System.out.print("Enter department:");
    String dept=sc.nextLine();
    sc.nextLine();
    System.out.print("Enter basic pay:");
    int bp=sc.nextInt();
}

```

```

sc.nextLine();
System.out.print("Enter project details:");
String proj=sc.nextLine();
System.out.print("Enter course:");
String cou=sc.nextLine();
for(int i=0;i<3;i++)
{
System.out.print("Enter mark "+(i+1)+" :");
marks[i]=sc.nextFloat();
}
TeachingAssistant f=new TeachingAssistant(name,add,empid,dept,bp,
proj,cou,marks);
System.out.println("NAME: "+f.getName());
System.out.println("ADDRESS: "+f.getAddress());
System.out.println("EMPID: "+f.getEmpid());
System.out.println("DEPARTMENT: "+f.getDept());
System.out.println("BASIC PAY: "+f.getBasic());
sc.nextLine();
System.out.println("DESIGNATION: "+f.getProject());
System.out.println("COURSE: "+f.getCourse());
System.out.println("CGPA: "+f.calcGPA());
System.out.println("SALARY: "+f.calSalary());
System.out.println("Do you want to change values?\n1.No\n2.Address\n3.Dept\n4.Course");
int ch =sc.nextInt();
if(ch==1)
{
    return;
}
else if(ch==2)
{
    System.out.println("Changed value: ");
    sc.nextLine();
    add= sc.nextLine();
    f.setAddress(add);
    System.out.println("New value: "+f.getAddress());
}
else if(ch==3)
{
    System.out.println("Changed value: ");
    sc.nextLine();
    dept= sc.nextLine();
    f.setDept(dept);
    System.out.println("New value: "+f.getDept());
}
else if(ch==4)
{
    System.out.println("Changed value: ");
    sc.nextLine();
    cou= sc.nextLine();
}

```

```
        f.setCourse(cou);
        System.out.println("New value: "+f.getCourse());
    }
}
}
```

MENU

1.Faculty

2.Teaching assistant

Enter option

1

Enter details:

Enter name:shreya

Enter address:chennai

Enter empid:s1

Enter department:cse

Enter basic pay:34000

Enter designation:prof

Enter course:cse

NAME: shreya

ADDRESS: chennai

EMPID: s1

DEPARTMENT: cse

BASIC PAY: 34000

DESIGNATION: prof

COURSE: cse

SALARY: 35400.0

Do you want to change values?

1.No

2.Address

3.Dept

4.Designation

5.Course

4

Changed value:

assistant prof

New value: assistant prof

MENU

1.Faculty

2.Teaching assistant

Enter option

2

Enter details:

Enter name:Sriram

Enter address:Madurai

Enter empid:s2

Enter department:CSE

Enter basic pay:25000

Enter project details:Artificial intelligence

Enter course:CSE

Enter mark 1 :99

Enter mark 2 :90

Enter mark 3 :90

NAME: Sriram

ADDRESS: Madurai

EMPID: s2

DEPARTMENT: CSE

BASIC PAY: 25000

DESIGNATION: Artificial intelligence

COURSE: CSE

CGPA: 9.3

SALARY: 25250.0

Do you want to change values?

1.No

2.Address

3.Dept

4.Course

1

## 2. SOURCE CODE :

```
import java.util.*;  
  
/*interface S{  
    abstract float getArea();  
    abstract float getPerimeter();  
}  
  
abstract class Shape{      //abstract class since it contains both abstract  
and non-abstract methods  
  
    String color = "red";  
  
    Shape(){}
    Shape(String c){  
        color = c;
    }
    public String getColor(){
        return color;
    }
    public void setColor(String c){
        color = c;
    }
    public float getRadius(){return 0.0f;}
    public float getWidth(){return 0.0f;}
    public float getLength(){return 0.0f;}
    public float getSide(){return 0.0f;}
    abstract float getArea();
    abstract float getPerimeter();
}  
  
class Circle extends Shape{
```

```
float radius = 1.0f;

Circle(){}
Circle(float r){
    setRadius(r);
}

Circle(float r, String c){
    setRadius(r);
    setColor(c);
}

public float getRadius(){
    return radius;
}

public void setRadius(float r){
    radius = r;
}

public float getArea(){      //has to be defined since Circle inherits
from Shape having abstract method

    return 3.14f*radius*radius;
}

public float getPerimeter(){      //has to be defined since Circle
inherits from Shape having abstract method

    return 2.0f*3.14f*radius;
}

}

class Rectangle extends Shape{

float width = 1.0f;

float length = 1.0f;
```

```
Rectangle(){}
Rectangle(float w, float l){
    setWidth(w);
    setLength(l);
}
Rectangle(float w, float l, String c){
    setWidth(w);
    setLength(l);
    setColor(c);
}
public float getWidth(){
    return width;
}
public void setWidth(float w){
    width = w;
}
public float getLength(){
    return length;
}
public void setLength(float l){
    length = l;
}
public float getArea(){      //has to be defined since Rectangle inherits
from Shape having abstract method
    return width*length;
}
```

```
    public float getPerimeter(){      //has to be defined since Rectangle
inherits from Shape having abstract method

        return 2.0f*(width+length);

    }

}

class Square extends Rectangle{

    Square(){}
    Square(float s){
        setSide(s);
    }
    Square(float s, String c){
        setSide(s);
        setColor(c);
    }
    public float getSide(){
        return width;
    }
    public void setSide(float s){
        width = s;
        length = s;
    }
    public float getArea(){      //has to be defined since Square extends
Rectangle which inherits from Shape having abstract method

        return width*length;
    }

    public float getPerimeter(){      //has to be defined since Square
extends Rectangle which inherits from Shape having abstract method
```

```

        return 2.0f*(width+length);

    }

}

public class Test{

    public static void main(String args[]){

        Scanner sc = new Scanner(System.in);

        Shape[] s = new Shape[3];

        s[0] = new Circle();

        s[1] = new Rectangle(2.0f,3.0f);

        s[2] = new Square(2.0f,"blue");

        for(int i=0;i<3;i++){

            System.out.println("Shape : " + s[i].getClass().getName());

            System.out.println("Color : " + s[i].getColor());

            System.out.println("Area : " + s[i].getArea());

            System.out.println("Perimeter : " + s[i].getPerimeter());

            System.out.println();

        }

    }

}

```

**OUTPUT :**

```

Shape : Circle

Color : red

Area : 3.14

Perimeter : 6.28

```

Shape : Rectangle

Color : red

Area : 6.0

Perimeter : 10.0

Shape : Square

Color : blue

Area : 4.0

Perimeter : 8.0

CLASS DIAGRAM :

Shape

```
# color : String = "red"  
  
+ Shape()  
+ Shape(color)  
+ getColor() : String  
+ setColor(color) : void  
+ getRadius() : float  
+ getWidth() : float  
+ getLength() : float  
+ getSide() : float  
abs getArea() : float  
abs getPerimeter() : float
```

### Circle

```
# radius : float = 1.0

+ Circle()
+ Circle(radius)
+ Circle(radius, color)
+ getRadius() : float
+ setRadius(radius) : void
+ getArea() : float
+ getPerimeter() : float
```

### Rectangle

```
# width : float = 1.0
# length : float = 1.0

+ Rectangle()
+ Rectangle(length, width)
+ Rectangle(length, width, color)
+ getWidth() : float
+ setWidth(width) : void
+ getLength() : float
+ setLength(length) : void
+ getArea() : float
+ getPerimeter() : float
```

### Square

No data members

```
+ Square()
+ Square(side)
+ Square(side, color)
+ getSize() : float
+ setSize(side) : void
```



**SSN College of Engineering**  
**Department of Computer Science and Engineering**  
**UCS1313 – Object Oriented Programming Using Java Lab**  
**II Year CSE - A, B Section ( III Semester)**  
**Academic Year 2020-21**  
**Batch: 2019- 2023**  
**Faculty Incharge : S.Rajalakshmi / S.Lakshmi Priya**

---

### **Exercise – 5 – Exception handling**

**Objective:**

1. To test the working of exception handling mechanism in Java

**Sample Learning Outcome:**

1. Need of exception handling and it's implementation in Java
2. User defined exception creation and usage
3. Usage of try, catch, finally, throw, throws

**Best Practices:**

1. Class Diagram usage
2. Naming convention – for file names, variables
3. Comment usage at proper places
4. Prompt messages during reading input and displaying output
5. Incremental program development
6. Modularity
7. All possible test cases in output

1. Create a class named “Person” which consists of name, age, aadharnumber. Create methods getInput(), display(), canVote(), hasAadhar(). Create and handle the following Exceptions.
  - a. For age -> if you give alphabets then throw NumberFormatException  
(Check for the condition explicitly and throw builtin exception)
  - b. For voting -> if age is less than 18 then throw MinorCitizenException  
(Check for the condition explicitly and throw user-defined exception)
  - c. For aadhar -> if no valid aadhar then throw NullPointerException  
(Check for the condition explicitly and throw builtin exception)
2. Create a class named “Account” which contains name, acct\_num, branch, balance, PAN\_num. Create functions for deposit and withdrawal. Write user-defined exceptions for the following conditions:
  - a. In deposit function, if the customer deposits money more than 25000, then throw the user defined exception “PANRequiredException” and get the PAN number and proceed the deposit.
  - b. In withdrawal function, if the customer requesting some money, check on withdrawal will it satisfy the minimum\_bal amount and throw the “MinBalRequiredException” exception. If the withdrawal amount is more than the balance amount then throw “NotEnoughMoneyInAccountException”.
  - c. Search for a particular acct\_num. If not present then throw “AccountNotFoundException”.
  - d. On PAN number entry check the format of 10 characters. First 5 characters then 4 numbers and then 1 character. If the format not matched then throw “PANFormatMismatchException”.

On account creation if user gives a non branch, then throw “BranchNotFoundException”

## Exercise-5 Exception handling

### Source code:

```
import java.util.*;  
  
class MinorCitizenException extends Exception  
{  
    private int age;  
    MinorCitizenException(int a)  
    {  
        age=a;  
    }  
    public String toString()  
    {  
        return ("Person of age "+age+" is a minor not eligible to vote");  
    }  
}  
  
class Person  
{  
    Scanner sc = new Scanner(System.in);  
    String name,aadharno,a;  
    int age;  
    void getInput() throws NumberFormatException  
    {  
        System.out.println("Enter details");  
        System.out.print("Enter name: ");  
        name=sc.nextLine();  
        System.out.print("Enter aadhar number: ");  
        aadharno=sc.nextLine();  
        System.out.print("Enter age: ");  
        a=sc.nextLine();  
        for(int i=0;i<a.length();i++)  
        {  
            if(Character.isLetter(a.charAt(i)))  
            {  
                throw new NumberFormatException();  
            }  
        }  
        age=Integer.parseInt(a);  
    }  
    void display()  
{  
        System.out.println("Name: "+name);  
        System.out.println("Aadhar no: "+aadharno);  
        System.out.println("Age: "+age);  
    }  
    void canVote() throws MinorCitizenException
```

```

{
    if(age<18)
    {
        throw new MinorCitizenException(age);
    }
    else
    {
        System.out.println("Eligible to vote");
    }
}

void hasAadhar() throws NullPointerException
{
    if(aadharno.length()!=10)
    {
        throw new NullPointerException();
    }
    else
    {
        System.out.println("Candidate has aadhar");
    }
}

class TestException
{
    public static void main(String args[])
    {
        Person p = new Person();
        try
        {
            p.getInput();
            p.canVote();
            p.hasAadhar();
            p.display();
        }catch(MinorCitizenException e)
        {
            System.out.println(e);
        }
        catch(NullPointerException e)
        {
            System.out.println("Aadhar number not valid!!!!");
        }
        catch(NumberFormatException e)
        {
            System.out.println("Enter valid age");
        }
    }
}

```

Output:

Enter details

Enter name: Shreya

Enter aadhar number: 1231231231

Enter age: e

Enter valid age

Enter details

Enter name: Shraddha

Enter aadhar number: 1233

Enter age: 20

Eligible to vote

Aadhar number not valid!!!

Enter details

Enter name: Shreya

Enter aadhar number: 1234516621

Enter age: 12

Person of age 12 is a minor not eligible to vote

Enter details

Enter name: Sriya

Enter aadhar number: 12333222111

Enter age: 2

Person of age 2 is a minor not eligible to vote

Source code:

```
import java.util.*;
import java.lang.*;
//user defined exceptions
class PANRequiredException extends Exception{}
class MinBalRequiredException extends Exception{}
class NotEnoughMoneyInAccountException extends Exception{}
class AccountNotFoundException extends Exception{}
class PANFormatMismatchException extends Exception{}
class BranchNotFoundException extends Exception{}

public class Account{
    String name, branch, PAN_num;
    int acct_num;
    float balance;

    void getInput(String n, String b, String p, int a) throws
BranchNotFoundException, PANFormatMismatchException{
        name = n;
        if(b.equals("Chennai")||b.equals("Trichy")||b.equals("Madurai")){
            branch = b;
        }
        else{
            throw new BranchNotFoundException();
        }
        if(p.equals("N/A")){
            PAN_num = p;
        }
        else{
```

```
if(p.length()!=10){

    throw new PANFormatMismatchException();

}

for(int i=0;i<4;i++){

    if(Character.isLetter(p.charAt(i))){


        continue;

    }

    else{

        throw new PANFormatMismatchException();

    }

}

for(int i=5;i<9;i++){

    if(Character.isDigit(p.charAt(i))){


        continue;

    }

    else{


        throw new PANFormatMismatchException();

    }

}

if(!Character.isLetter(p.charAt(9))){


    throw new PANFormatMismatchException();

}

PAN_num = p;

acct_num = a;

balance = 1000f; //opening account balance
```

```
}

void depositAmt(float amt) throws PANRequiredException{
    if(amt>=25000&&PAN_num.equals("N/A")){
        throw new PANRequiredException();
    }
    else{
        balance+=amt;
        System.out.println("After deposit, balance : " + balance);
    }
}

void withdrawAmt(float amt) throws MinBalRequiredException,
NotEnoughMoneyInAccountException{
    if(amt>balance){
        throw new NotEnoughMoneyInAccountException();
    }
    else if((balance-amt)<1000){
        throw new MinBalRequiredException();
    }
    else{
        balance-=amt;
        System.out.println("After withdrawal, balance : " + balance);
    }
}

public static void main(String args[]){
    Scanner sc = new Scanner(System.in);
    int n=0,no,f=0,anum,ch=0,x=0;
    String name,br,pan;
```

```

float damt=0.0f,wamt=0.0f;

char ans='y';

System.out.println("Welcome to SBI Online Portal !!! ");

System.out.println("Enter no. of customers : ");

no = sc.nextInt();

Account[] a = new Account[no]; //array of type Account

for(int i=0;i<no;i++){

    a[i] = new Account(); //creating memory for each array obj.

}

try{

    for(int i=0;i<no;i++){

        System.out.println("\nCustomer " + (i+1));

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

        name = sc.next();

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

        br = sc.next();

        System.out.println("Enter PAN number : ");

        pan = sc.next();

        System.out.println("Enter account number : ");

        anum = sc.nextInt();

        a[i].getInput(name,br,pan,anum); //input from user passed as

arguments

    }

    do{

        System.out.println("\n1.Deposit\n2.Withdraw\n3.Exit");

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

        ch = sc.nextInt();

```

```
if(ch==1){

System.out.println("Enter account number : ");

n = sc.nextInt();

for(int i=0;i<no;i++){

x=i;

if(a[i].acct_num==n){

f=1;

System.out.println("Account Number : " + a[i].acct_num);

System.out.println("Name : " + a[i].name);

System.out.println("Branch : " + a[i].branch);

System.out.println("PAN Number : " + a[i].PAN_num);

System.out.println("Balance : " + a[i].balance);

System.out.println("Enter amount to deposit : ");

damt = sc.nextFloat();

a[i].depositAmt(damt);

}

}

if(f==0){

throw new AccountNotFoundException();

}

}

else if(ch==2){

System.out.println("Enter account number : ");

n = sc.nextInt();

for(int i=0;i<no;i++){

if(a[i].acct_num==n){
```

```
f=1;

System.out.println("Account Number : " + a[i].acct_num);

System.out.println("Name : " + a[i].name);

System.out.println("Branch : " + a[i].branch);

System.out.println("PAN Number : " + a[i].PAN_num);

System.out.println("Balance : " + a[i].balance);

System.out.println("Enter amount to withdraw : ");

wamt = sc.nextFloat();

a[i].withdrawAmt(wamt);

}

}

if(f==0){

    throw new AccountNotFoundException();

}

}

else if(ch==3){

System.exit(0);

}

else{

System.out.println("Invalid option !!");

}

System.out.println("Do you want to continue ?");

ans = sc.next().charAt(0);

}while(ans=='y' | ans=='Y');

}

catch(BranchNotFoundException b){
```

```
System.out.println(b.getClass().getName());  
}  
  
catch(PANRequiredException p){  
    System.out.println(p.getClass().getName());  
  
    try{  
        System.out.println("Enter PAN Number : ");  
        pan = sc.next();  
        if(pan.length()!=10){  
            throw new PANFormatMismatchException();  
        }  
        for(int i=0;i<4;i++){  
            if(Character.isLetter(pan.charAt(i))){  
                continue;  
            }  
            else{  
                throw new PANFormatMismatchException();  
            }  
        }  
        for(int i=5;i<9;i++){  
            if(Character.isDigit(pan.charAt(i))){  
                continue;  
            }  
            else{  
                throw new PANFormatMismatchException();  
            }  
        }  
    }
```

```

if(!Character.isLetter(pan.charAt(9))){
    throw new PANFormatMismatchException();
}

a[x].PAN_num = pan;

System.out.println("After modifying, PAN Number : " +
a[x].PAN_num);

a[x].depositAmt(damt);

System.out.println("Do you want to continue ?");

ans = sc.next().charAt(0);

if(ans=='n'){System.exit(0);}

do{

System.out.println("\n1.Deposit\n2.Withdraw\n3.Exit");

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

ch = sc.nextInt();

if(ch==1){

System.out.println("Enter account number : ");

n = sc.nextInt();

for(int i=0;i<no;i++){

if(a[i].acct_num==n){

f=1;

System.out.println("Account Number : " + a[i].acct_num);

System.out.println("Name : " + a[i].name);

System.out.println("Branch : " + a[i].branch);

System.out.println("PAN Number : " + a[i].PAN_num);

System.out.println("Balance : " + a[i].balance);

System.out.println("Enter amount to deposit : ");

damt = sc.nextFloat();
```

```
a[i].depositAmt(damt);

}

}

if(f==0){

    throw new AccountNotFoundException();

}

}

else if(ch==2){

    System.out.println("Enter account number : ");

    n = sc.nextInt();

    for(int i=0;i<no;i++){

        if(a[i].acct_num==n){

            f=1;

            System.out.println("Account Number : " + a[i].acct_num);

            System.out.println("Name : " + a[i].name);

            System.out.println("Branch : " + a[i].branch);

            System.out.println("PAN Number : " + a[i].PAN_num);

            System.out.println("Balance : " + a[i].balance);

            System.out.println("Enter amount to withdraw : ");

            wamt = sc.nextFloat();

            a[i].withdrawAmt(wamt);

        }

    }

    if(f==0){

        throw new AccountNotFoundException();

    }

}
```

```
        }

        else if(ch==3){

            System.exit(0);

        }

        else{

            System.out.println("Invalid option !!");

        }

        System.out.println("Do you want to continue ?");

        ans = sc.next().charAt(0);

        }while(ans=='y' || ans=='Y');

    }

    catch(PANFormatMismatchException e){

        System.out.println(e.getClass().getName());

    }

    catch(PANRequiredException q){

        System.out.println(q.getClass().getName());

    }

    catch(MinBalRequiredException m){

        System.out.println("Cannot withdraw : " +
m.getClass().getName());

    }

    catch(NotEnoughMoneyInAccountException i){

        System.out.println("Cannot withdraw : " +
i.getClass().getName());

    }

    catch(AccountNotFoundException r){

        System.out.println(r.getClass().getName());

    }

}
```

```

        }
    }

    catch(PANFormatMismatchException e){
        System.out.println(e.getClass().getName());
    }

    catch(MinBalRequiredException m){
        System.out.println("Cannot withdraw : " +
m.getClass().getName());
    }

    catch(NotEnoughMoneyInAccountException i){
        System.out.println("Cannot withdraw : " +
i.getClass().getName());
    }

    catch(AccountNotFoundException r){
        System.out.println(r.getClass().getName());
    }

}
}

```

### Test Case 1

Enter no. of customers :

1

Customer 1

Enter name :

Shreya

Enter branch :

Coimbatore

Enter PAN number :

N/A

Enter account number :

12451

**BranchNotFoundException**

**Test case 2**

Enter no. of customers :

1

Customer 1

Enter name :

Shreya

Enter branch :

Chennai

Enter PAN number :

12345abcd6

Enter account number :

12345

**PANFormatMismatchException**

**Test case 3**

Enter no. of customers :

1

Customer 1

Enter name :

Shreya

Enter branch :

Chennai

Enter PAN number :

N/A

Enter account number :

12345

1.Deposit

2.Withdraw

3.Exit

Enter choice :

3

#### **Test case 4**

Enter no. of customers :

2

Customer 1

Enter name :

Shreya

Enter branch :

Chennai

Enter PAN number :

N/A

Enter account number :

12345

Customer 2

Enter name :

Sneha

Enter branch :

Trichy

Enter PAN number :

N/A

Enter account number :

23456

1.Deposit

2.Withdraw

3.Exit

Enter choice :

1

Enter account number :

34567

**AccountNotFoundException**

#### Test case 5

Enter no. of customers :

2

Customer 1

Enter name :

Aravind

Enter branch :

Chennai

Enter PAN number :

N/A

Enter account number :

12345

Customer 2

Enter name :

Sneha

Enter branch :

Trichy

Enter PAN number :

abcde1234f

Enter account number :

23456

1.Deposit

2.Withdraw

3.Exit

Enter choice :

1

Enter account number :

12345

Account Number : 12345

Name : Arjun

Branch : Chennai

PAN Number : N/A

Balance : 1000.0

Enter amount to deposit :

26000

**PANRequiredException**

Enter PAN Number :

bcdef2345g

After modifying, PAN Number : bcdef2345g

After deposit, balance : 27000.0

Do you want to continue ?

y

1.Deposit

2.Withdraw

3.Exit

Enter choice :

2

Enter account number :

12345

Account Number : 12345

Name : Arjun

Branch : Chennai

PAN Number : bcdef2345g

Balance : 27000.0

Enter amount to withdraw :

26500

**Cannot withdraw : MinBalRequiredException**

**Test case 6**

Enter no. of customers :

2

Customer 1

Enter name :

Aravind

Enter branch :

Chennai

Enter PAN number :

N/A

Enter account number :

12345

Customer 2

Enter name :

Sneha

Enter branch :

Trichy

Enter PAN number :

N/A

Enter account number :

23456

1.Deposit

2.Withdraw

3.Exit

Enter choice :

2

Enter account number :

23456

Account Number : 23456

Name : Sneha

Branch : Trichy

PAN Number : N/A

Balance : 1000.0

Enter amount to withdraw :

5000

**Cannot withdraw : NotEnoughMoneyInAccountException**

**SSN College of Engineering**  
**Department of Computer Science and Engineering**  
**UCS1313 – Object Oriented Programming Using Java Lab**

**II Year CSE - A, B Section ( III Semester)**

**Academic Year 2020-21**

**Batch: 2019- 2023**

**Faculty Incharge : S.Rajalakshmi / S.Lakshmi Priya**

---

**Exercise - 6 – Packages**

**Objective:**

1. To test the creation and usage of packages in Java

**Sample Learning Outcome:**

1. Need of packages and steps to create it in Java
2. Create a user defined package and access it outside the package
3. Create subpackages and their usage

**Best Practices:**

1. Class Diagram usage
2. Naming convention – for file names, variables
3. Comment usage at proper places
4. Prompt messages during reading input and displaying output
5. Incremental program development
6. Modularity
7. All possible test cases in output

1. Create a package named ‘shapes’ with the following three java classes. Create class for geometric shapes like Square, Triangle and Circle. The classes should contain the methods to calculate area and perimeter. Use this package to find area and perimeter of different shapes as chosen by the user.
2. Create a package named ‘mypack.converter’ with the following java classes. Implement distance converter (meter to KM, miles to KM and vice versa) in DistanceConverter class, time converter (hours to minutes, minute to seconds and vice versa) in TimeConverter class. Write the main program to access these classes outside the package.

## Exercise-6 Packages

Code:

### Circle.java

```
package shapes;          //to include the class in the package shapes
public class Circle {   //separate files for each class since they are public

    double radius;
    public Circle(double r)
    {
        radius = r;
    }

    public double calcPerimeter()
    {
        return 2*Math.PI*radius;
    }

    public double calcArea()
    {
        return Math.PI*radius*radius;
    }

    public void display()      //declared public to be used outside package
    {
        System.out.println("Radius: "+radius);
        System.out.println("Area: "+this.calcArea());
        System.out.println("Perimeter: "+this.calcPerimeter());
    }
}
```

### Square.java

```
package shapes;
public class Square {

    double side;
    public Square(double s)           //initialize using constructor
    {
        side = s;
    }
```

```

public double calcPerimeter()          //calculating perimeter
{
    return 4*side;
}

public double calcArea()
{
    return side*side;
}

public void display()
{
    System.out.println("Side: "+side);
    System.out.println("Area: "+this.calcArea());
    System.out.println("Perimeter: "+this.calcPerimeter());
}

}

```

### Triangle.java

```

package shapes;
public class Triangle {

    double base,height;
    public Triangle(double b,double h)
    {
        base = b;
        height = h;
    }

    public double calcPerimeter()
    {
        return Math.sqrt(base*base+height*height)+height+base;
    }

    public double calcArea()          //calculating area
    {
        return 0.5*base*height;
    }

    public void display()           //displaying details
    {
        System.out.println("Base: "+base);
        System.out.println("Height: "+height);
        System.out.println("Area: "+this.calcArea());
        System.out.println("Perimeter: "+this.calcPerimeter());
    }
}

```

```
}
```

### Main.java

```
import shapes.*;
import java.util.*;

public class MyShape {
    private static Scanner sc = new Scanner(System.in);
    public static void main(String args[])
    {
        int n;
        do
        {
            System.out.print("MENU\n1.Triangle\n2.Circle\n3.Square\n4.Exit\nEnter choice: ");
            n = sc.nextInt();
            switch(n)
            {
                case 1:
                {
                    System.out.print("Enter base and height: ");
                    Triangle t = new Triangle(sc.nextDouble(),sc.nextDouble());
                    t.display();
                    break;
                }

                case 2:
                {
                    System.out.print("Enter radius: ");
                    Circle c = new Circle(sc.nextDouble());
                    c.display();
                    break;
                }

                case 3:
                {
                    System.out.print("Enter side: ");
                    Square s = new Square(sc.nextDouble());
                    s.display();
                    break;
                }
            }
        }while(n!=4);
    }
}
```

**Output:**

MENU

1.Triangle

2.Circle

3.Square

4.Exit

Enter choice: 1

Enter base and height: 5 12

Base: 5.0

Height: 12.0

Area: 30.0

Perimeter: 30.0

MENU

1.Triangle

2.Circle

3.Square

4.Exit

Enter choice: 2

Enter radius: 10.2

Radius: 10.2

Area: 326.851299679482

Perimeter: 64.08849013323177

MENU

1.Triangle

2.Circle

3.Square

4.Exit

Enter choice: 3

Enter side: 12.1

Side: 12.1

Area: 146.41

Perimeter: 48.4

MENU

1.Triangle

2.Circle

3.Square

4.Exit

Enter choice: 4

PACKAGE : shapes

Circle	Triangle	Square
<code>~radius:double</code>	<code>~base:double</code> <code>~height:double</code>	<code>~side:double</code>
<code>+Circle(radius)</code> <code>+calcPerimeter():double</code> <code>+calcArea():double</code>	<code>+Triangle(base,height)</code> <code>+calcPerimeter():double</code> <code>+calcArea():double</code>	<code>+Square(side)</code> <code>+calcPerimeter():double</code> <code>+calcArea():double</code>

**Code:**

**DistanceConvertor.java**

```
package mypack.convertor;           //including classes to the package

public class DistanceConvertor {

    double inp;
    public DistanceConvertor(double x)      //constructor
    {
        inp = x;
    }

    public double metTokm()
    {
        return inp/1000;
    }

    public double kmTomet()
    {
        return inp*1000;
    }

    public double kmTomile()           //calculating conversions
    {
        return 0.621371*inp;
    }

    public double mileTokm()
    {
        return 1.60934*inp;
    }

}
```

**TimeConvertor.java**

```
package mypack.convertor;

public class TimeConvertor {

    public double inp;
    public TimeConvertor(double x)
    {
        inp = x;
    }
```

```

public double hrsTomin() //calculating time conversions
{
    return 60*inp;
}

public double minTosec()
{
    return 60*inp;
}

public double sectomin()
{
    return 0.0166667*inp;
}

public double minTohrs()
{
    return 0.0166667*inp;
}
}

```

### Main.java

```

import java.util.*;
import mypack.convertor.*;

public class MyConvertor {

    private static Scanner sc = new Scanner(System.in);
    public static void main(String [] args)
    {
        int n,c;
        do
        {

            System.out.print("MENU\n1.LENGTH\n2.TIME\n3.EXIT\nEnter choice: ");
        };
        n = sc.nextInt();
        switch(n)
        {
            case 1:
                DistanceConvertor d;
                System.out.print("MENU\n1.metre to km\n2.km to metre\n3.mile
to km\n4.km to mile\nEnter choice: ");
                c = sc.nextInt();
                switch(c)
                {
                    case 1:
                        System.out.print("Enter distance in metres: ");
                        d = new DistanceConvertor(sc.nextDouble());

```

```

        System.out.println("Distance in kms is: "+d.metTokm());
        break;

    case 2:
        System.out.print("Enter distance in kilometres: ");
        d = new DistanceConvertor(sc.nextDouble());
        System.out.println("Distance in metres is: "+d.kmToMet())
;
        break;

    case 3:
        System.out.print("Enter distance in miles: ");
        d = new DistanceConvertor(sc.nextDouble());
        System.out.println("Distance in kms is: "+d.mileTokm());
        break;

    case 4:
        System.out.print("Enter distance in kms: ");
        d = new DistanceConvertor(sc.nextDouble());
        System.out.println("Distance in mile is: "+d.kmTomile());
        break;
    }
    break;

case 2:
{
    TimeConvertor t;
    System.out.print("MENU\n1.min to hr\n2.hr to min\n3.second to min\n4.min to second\nEnter choice: ");
    c = sc.nextInt();
    switch(c)
    {
        case 1:
            System.out.print("Enter time in minutes: ");
            t = new TimeConvertor(sc.nextDouble());
            System.out.println("Time in hours is: "+t.minTohrs())
;
            break;

        case 2:
            System.out.print("Enter time in hours: ");
            t = new TimeConvertor(sc.nextDouble());
            System.out.println("Time in minutes is: "+t.hrsTomin(
));
            break;

        case 3:
            System.out.print("Enter time in seconds: ");
            t = new TimeConvertor(sc.nextDouble());

```

```

        System.out.println("Time in minutes is: "+t.secToMin(
));
        break;

    case 4:
        System.out.print("Enter time in minutes: ");
        t = new TimeConvertor(sc.nextDouble());
        System.out.println("Time in seconds is: "+t.minToSec(
));
        break;
    }
}
break;
}
}while(n!=3);

}
}
}

```

**Output:**

MENU

1.LENGTH

2.TIME

3.EXIT

Enter choice: 1

MENU

1.metre to km

2.km to metre

3.mile to km

4.km to mile

Enter choice: 1

Enter distance in metres: 123112

Distance in kms is: 123.112

MENU

1.LENGTH

2.TIME

3.EXIT

Enter choice: 1

MENU

1.metre to km

2.km to metre

3.mile to km

4.km to mile

Enter choice: 2

Enter distance in kilometres: 1231

Distance in metres is: 1231000.0

MENU

1.LENGTH

2.TIME

3.EXIT

Enter choice: 1

MENU

1.metre to km

2.km to metre

3.mile to km

4.km to mile

Enter choice: 3

Enter distance in miles: 1922

Distance in kms is: 3093.15148

MENU

1.LENGTH

2.TIME

3.EXIT

Enter choice: 1

MENU

1.metre to km

2.km to metre

3.mile to km

4.km to mile

Enter choice: 4

Enter distance in kms: 211

Distance in mile is: 131.109281

MENU

1.LENGTH

2.TIME

3.EXIT

Enter choice: 2

MENU

1.min to hr

2.hr to min

3.second to min

4.min to second

Enter choice: 1

Enter time in minutes: 20

Time in hours is: 0.333334

MENU

1.LENGTH

2.TIME

3.EXIT

Enter choice: 2

MENU

1.min to hr

2.hr to min

3.second to min

4.min to second

Enter choice: 2

Enter time in hours: 45

Time in minutes is: 2700.0

MENU

1.LENGTH

2.TIME

3.EXIT

Enter choice: 2

MENU

1.min to hr

2.hr to min

3.second to min

4.min to second

Enter choice: 3

Enter time in seconds: 20

Time in minutes is: 0.333334

MENU

1.LENGTH

2.TIME

3.EXIT

Enter choice: 2

MENU

1.min to hr

2.hr to min

3.second to min

4.min to second

Enter choice: 4

Enter time in minutes: 30

Time in seconds is: 1800.0

MENU

1.LENGTH

2.TIME

3.EXIT

Enter choice: 3

PACKAGE : mypack.convertor

DistanceConvertor	TimeConvertor
~inp:double	~inp:double
+DistanceConvertor(double x) +metTokm():double +kmTomet():double +mileTokm():double +kmTomile():double	+TimeConvertor(double x) +hrsTomin():double + minTohrs():double + secTomin():double + minTosec():double

**SSN College of Engineering**  
**Department of Computer Science and Engineering**  
**UCS1313 – Object Oriented Programming Using Java Lab**  
**II Year CSE - A, B Section ( III Semester)**  
**Academic Year 2020-21**  
**Batch: 2019- 2023**  
**Faculty Incharge : S.Rajalakshmi / S.Lakshmi Priya**

---

**Exercise - 7 – Generics**

**Objective:**

1. To implement generic types – generic classes and methods

**Sample Learning Outcome:**

1. Learning to create a generic class and method
2. Use the generic class to store/manipulate elements of different datatypes

**Best Practices:**

1. Class Diagram usage
2. Naming convention – for file names, variables
3. Comment usage at proper places
4. Prompt messages during reading input and displaying output
5. Incremental program development
6. Modularity
7. All possible test cases in output

**Exercises:**

1. Write a Java program to find the maximum value from the given type of elements using a generic function.
2. Write a Java program to create a generic stack and perform the operations.
3. Write a Java program to perform a sorting operation on various types of elements using a generic method.

## Exercise - 7 - Generics

### Code:

```
import java.util.*;  
  
class MaxGen <T extends Comparable> //Objects of type T can be compared  
{  
    T[] e;  
    MaxGen(T[] ob)  
    {  
        e = ob;  
    }  
    T findMax()  
    {  
        T max = e[0];           //initializing max to first element  
        for(int i=0;i<e.length;i++)  
        {  
            if(e[i].compareTo(max)>0)  
            {  
                max = e[i];  
            }  
        }  
        return max;  
    }  
}  
  
class MaxMain  
{  
    private static Scanner sc = new Scanner(System.in);  
    public static void main(String[] args)  
    {  
        int c,n;  
        System.out.print("MENU\n1.Integer\n2.Double\n3.String\nEnter choice: ");  
        c = sc.nextInt();  
        System.out.print("Enter size: ");  
        n = sc.nextInt();  
        sc.nextLine();  
        if(c == 1)  
        {  
            Integer[] iob = new Integer[n];  
            for(int i=0;i<n;i++)  
            {  
                System.out.print("Enter the array element "+(i+1)+": ");  
                iob[i] = (Integer)sc.nextInt();  
            }  
            MaxGen<Integer> l1 = new MaxGen<Integer>(iob);  
            System.out.println("MAXIMUM ELEMENT: "+l1.findMax());  
        }  
    }  
}
```

```

        else if(c == 2)
    {
        Double[] dob = new Double[n];
        for(int i=0;i<n;i++)
        {
            System.out.print("Enter the array element "+(i+1)+": ");
            dob[i] = (Double)sc.nextDouble();
        }
        MaxGen<Double> l2 = new MaxGen<Double>(dob);
        System.out.println("MAXIMUM ELEMENT: "+l2.findMax());
    }
    else if(c == 3)
    {
        String[] ob = new String[n];
        for(int i=0;i<n;i++)
        {
            System.out.print("Enter the array element "+(i+1)+": ");
            ob[i] = (String)sc.nextLine();
        }
        MaxGen<String> l3 = new MaxGen<String>(ob);
        System.out.println("MAXIMUM ELEMENT: "+l3.findMax());
    }
}
}

```

**Output:**

MENU

1.Integer

2.Double

3.String

Enter choice: 1

Enter size: 5

Enter the array element 1: 2

Enter the array element 2: 8

Enter the array element 3: 7

Enter the array element 4: 9

Enter the array element 5: 1

MAXIMUM ELEMENT: 9

MENU

1.Integer

2.Double

3.String

Enter choice: 2

Enter size: 5

Enter the array element 1: 2

Enter the array element 2: 8.7

Enter the array element 3: 6.4

Enter the array element 4: 9.6

Enter the array element 5: 9.9

MAXIMUM ELEMENT: 9.9

MENU

1.Integer

2.Double

3.String

Enter choice: 3

Enter size: 5

Enter the array element 1: hello

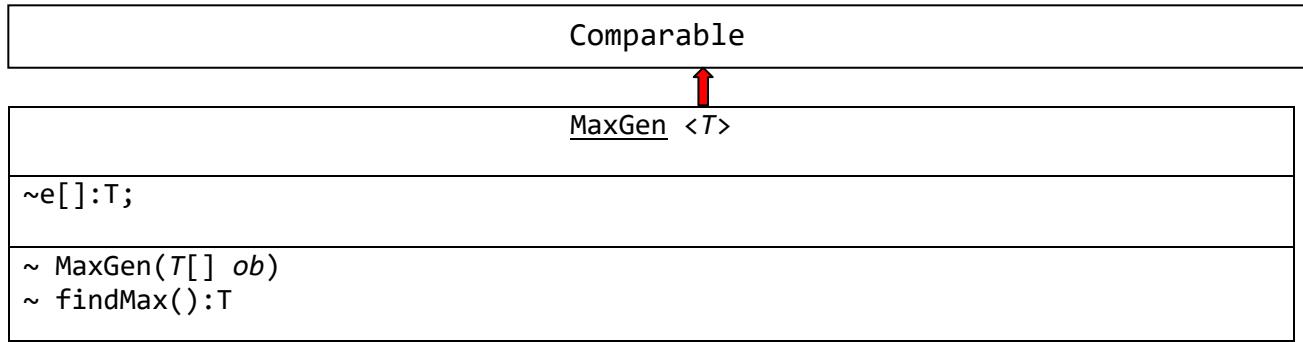
Enter the array element 2: java

Enter the array element 3: zebra

Enter the array element 4: student

Enter the array element 5: cat

MAXIMUM ELEMENT: zebra



**Code:**

```

import java.util.*;

class StackImpl<T>
{
    int top,size;
    T [] ob;

    StackImpl(T[] s,int size)
    {
        ob = s;
        top = -1;
        this.size = size;
    }

    public boolean push(T value)      //to insert to top of stack
    {
        if(!isFull())
        {
            ++top;
            ob[top] = value;
            return true;
        }
        else
        {
            System.out.println("STACK FULL");
            return false;
        }
    }

    public T pop()                  //to remove from top of stack
    {
        if(isEmpty())
        {
            System.out.println("STACK IS EMPTY");
        }
        else
        {
    
```

```

        T data = ob[top];
        top--;
        return data;
    }
    return null;
}

public boolean contains(T value)          //to search for a value in stack
{
    if(!isEmpty())
    {
        for(int i=0;i<top+1;i++)
        {
            if(value == ob[i])
            {
                System.out.println("FOUND");
                return true;
            }
        }
    }
    else if(isEmpty())
    {
        System.out.println("STACK EMPTY");
        return false;
    }
    System.out.println("NOT FOUND");
    return false;
}

public void display()
{
    if(!isEmpty())
    {
        for(int i=0;i<top+1;i++)
        {
            System.out.print(ob[i]+"\t");
        }
        System.out.println();
    }
    else
    {
        System.out.println("STACK IS EMPTY");
    }
}

public boolean isEmpty()
{
    if(top == -1)           //no element in stack
    {
        return true;
    }
}

```

```

        }
        return false;
    }

    public boolean isFull()
    {
        if(top<size-1)
        {
            return false;
        }
        return true;
    }
}

class StackMain
{
    private static Scanner sc = new Scanner(System.in);
    public static void main(String [] args)
    {
        System.out.print("MENU\n1.Integer\n2.Character\nEnter choice: ");
        int n,c,size;
        n = sc.nextInt();
        if(n==1)
        {
            Integer[] i = new Integer[100];
            System.out.print("Enter size of stack: ");
            size = sc.nextInt();
            StackImpl<Integer> ob = new StackImpl<Integer>(i,size);
            do
            {
                System.out.print("MENU\n1.Push\n2.Pop\n3.Search\n4.Display\n5
.Exit\nEnter choice: ");
                c = sc.nextInt();
                switch(c)
                {
                    case 1:
                        System.out.print("Enter element to push: ");
                        ob.push(sc.nextInt());
                        break;

                    case 2:
                        System.out.println("Popped element: "+ob.pop());
                        break;

                    case 3:
                        System.out.print("Enter element to search: ");
                        ob.contains(sc.nextInt());
                        break;

                    case 4:
                }
            }
        }
    }
}

```

```

        ob.display();
    }
}while(c!=5);
}
else if(n==2)
{
    Character[] str = new Character[100];
    System.out.print("Enter size of stack: ");
    size = sc.nextInt();
    StackImpl<Character> s = new StackImpl<Character>(str,size);
    do
    {
        System.out.print("MENU\n1.Push\n2.Pop\n3.Search\n4.Display\n5
.Exit\nEnter choice: ");
        c = sc.nextInt();
        switch(c)
        {
            case 1:
                System.out.print("Enter element to push: ");
                s.push(sc.next().charAt(0));
                break;

            case 2:
                System.out.println("Popped element: "+s.pop());
                break;

            case 3:
                System.out.print("Enter element to search: ");
                s.contains(sc.next().charAt(0));
                break;

            case 4:
                s.display();
        }
    }while(c!=5);
}
}

```

**Output:**

```

MENU
1.Integer
2.Character
Enter choice: 2

```

Enter size of stack: 5

MENU

1.Push

2.Pop

3.Search

4.Display

5.Exit

Enter choice: 2

STACK IS EMPTY

Popped element: null

MENU

1.Push

2.Pop

3.Search

4.Display

5.Exit

Enter choice: 1

Enter element to push: e

MENU

1.Push

2.Pop

3.Search

4.Display

5.Exit

Enter choice: 1

Enter element to push: t

MENU

- 1.Push
- 2.Pop
- 3.Search
- 4.Display
- 5.Exit

Enter choice: 1

Enter element to push: w

MENU

- 1.Push
- 2.Pop
- 3.Search
- 4.Display
- 5.Exit

Enter choice: 1

Enter element to push: b

MENU

- 1.Push
- 2.Pop
- 3.Search
- 4.Display
- 5.Exit

Enter choice: 1

Enter element to push: q

MENU

- 1.Push

2.Pop  
3.Search  
4.Display  
5.Exit

Enter choice: 4

e t w b q

MENU

1.Push  
2.Pop  
3.Search  
4.Display  
5.Exit

Enter choice: 1

Enter element to push: m

STACK FULL

MENU  
1.Push  
2.Pop  
3.Search  
4.Display  
5.Exit

Enter choice: 2

Popped element: q

MENU

1.Push  
2.Pop

3.Search

4.Display

5.Exit

Enter choice: 3

Enter element to search: q

NOT FOUND

MENU

1.Push

2.Pop

3.Search

4.Display

5.Exit

Enter choice: 3

Enter element to search: t

FOUND

MENU

1.Push

2.Pop

3.Search

4.Display

5.Exit

Enter choice: 5

MENU

1.Integer

2.Character

Enter choice: 1

Enter size of stack: 5

MENU

1.Push

2.Pop

3.Search

4.Display

5.Exit

Enter choice: 2

STACK IS EMPTY

Popped element: null

MENU

1.Push

2.Pop

3.Search

4.Display

5.Exit

Enter choice: 3

Enter element to search: 4

STACK EMPTY

MENU

1.Push

2.Pop

3.Search

4.Display

5.Exit

Enter choice: 4

STACK IS EMPTY

MENU

1.Push

2.Pop

3.Search

4.Display

5.Exit

Enter choice: 1

Enter element to push: 4

MENU

1.Push

2.Pop

3.Search

4.Display

5.Exit

Enter choice: 1

Enter element to push: 6

MENU

1.Push

2.Pop

3.Search

4.Display

5.Exit

Enter choice: 1

Enter element to push: 5

MENU

- 1.Push
- 2.Pop
- 3.Search
- 4.Display
- 5.Exit

Enter choice: 1

Enter element to push: 8

MENU

- 1.Push
- 2.Pop
- 3.Search
- 4.Display
- 5.Exit

Enter choice: 1

Enter element to push: 9

MENU

- 1.Push
- 2.Pop
- 3.Search
- 4.Display
- 5.Exit

Enter choice: 1

Enter element to push: 7

STACK FULL

MENU

1.Push

2.Pop

3.Search

4.Display

5.Exit

Enter choice: 4

4        6        5        8        9

MENU

1.Push

2.Pop

3.Search

4.Display

5.Exit

Enter choice: 2

Popped element: 9

MENU

1.Push

2.Pop

3.Search

4.Display

5.Exit

Enter choice: 4

4        6        5        8

MENU

1.Push

2.Pop

3.Search

4.Display

5.Exit

Enter choice: 5

<u>StackImpl&lt;T&gt;</u>
<pre>~top:int ~size:int ~ob[]:T</pre>
<pre>~ StackImpl(T[] s,int size) + push(T value):Boolean + pop():T + contains(T value): boolean + display():void + isEmpty(): boolean + isFull(): boolean</pre>

Code:

```
import java.util.*;

class Sort<T extends Comparable> {

    T [] obj;
    Sort(T[] e)
    {
        obj = e;
    }

    void sortElements()
    {
        for(int i=0;i<obj.length;i++)
        {
            for(int j=i;j<obj.length;j++)
            {
                if(obj[i].compareTo(obj[j])>0) //sorting using swap
                {
                    T x = obj[i];
                    obj[i] = obj[j];
                    obj[j] = x;
                }
            }
        }
    }
}
```

```

        }
    }
}

void displaySort()
{
    this.sortElements();
    for(int i=0;i<obj.length;i++)
    {
        System.out.print(obj[i]+"\t");
    }
    System.out.println();
}
}

class SortTest
{
    private static Scanner sc = new Scanner(System.in);
    public static void main(String [] args)
    {
        int c,n;
        System.out.print("MENU\n1.Integer\n2.Double\n3.String\nEnter choice:");
    );
        c = sc.nextInt();
        System.out.print("Enter size: ");
        n = sc.nextInt();
        sc.nextLine();
        if(c == 1)
        {
            Integer[] iob = new Integer[n];
            for(int i=0;i<n;i++)
            {
                System.out.print("Enter the array element "+(i+1)+": ");
                iob[i] = (Integer)sc.nextInt();
            }
            Sort<Integer> l1 = new Sort<Integer>(iob);
            l1.displaySort();
        }
        else if(c == 2)
        {
            Double[] dob = new Double[n];
            for(int i=0;i<n;i++)
            {
                System.out.print("Enter the array element "+(i+1)+": ");
                dob[i] = (Double)sc.nextDouble();
            }
            Sort<Double> inums = new Sort<Double>(dob);
            inums.displaySort();
        }
    }
}
```

```
        else if(c == 3)
    {
        String[] ob = new String[n];
        for(int i=0;i<n;i++)
        {
            System.out.print("Enter the array element "+(i+1)+": ");
            ob[i] = (String)sc.nextLine();
        }
        Sort<String> inums = new Sort<String>(ob);
        inums.displaySort();
    }
}
```

**Output:**

MENU

1.Integer

2.Double

3.String

Enter choice: 1

Enter size: 4

Enter the array element 1: 2

Enter the array element 2: 1

Enter the array element 3: 5

Enter the array element 4: 3

1        2        3        5

MENU

1.Integer

2.Double

3.String

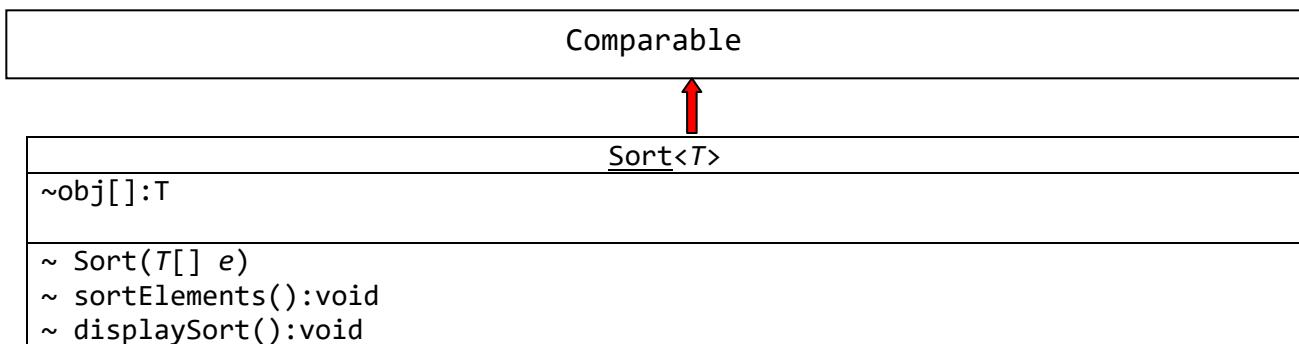
Enter choice: 2

```
Enter size: 3  
Enter the array element 1: 4.1  
Enter the array element 2: 1.3  
Enter the array element 3: 4.5  
1.3      4.1      4.5
```

MENU

- 1.Integer
- 2.Double
- 3.String

```
Enter choice: 3  
Enter size: 4  
Enter the array element 1: hello  
Enter the array element 2: world  
Enter the array element 3: java  
Enter the array element 4: zebra  
hello    java    world    zebra
```



**SSN College of Engineering**  
**Department of Computer Science and Engineering**  
**UCS1313 – Object Oriented Programming Using Java Lab**  
**II Year CSE - A, B Section ( III Semester)**  
**Academic Year 2020-21**  
**Batch: 2019- 2023**  
**Faculty Incharge : S.Rajalakshmi / S.Lakshmi Priya**

---

**Exercise - 8 – Collection Framework**

**Objective:**

1. To perform operations on ArrayList and LinkedList in Java

**Sample Learning Outcome:**

1. Usage of ArrayList and LinkedList in Java
2. Perform operations using Collection Framework
3. Perform merge, union, intersection and comparison on ArrayLists

**Best Practices:**

1. Class Diagram usage
2. Naming convention – for file names, variables
3. Comment usage at proper places
4. Prompt messages during reading input and displaying output
5. Incremental program development
6. Modularity
7. All possible test cases in output

1. Write a program to perform string operations using ArrayList. Write functions for the following

  - a. Append - add at end
  - b. Insert – add at particular index
  - c. Find the index of a particular element (Search)
  - d. Display the list
  - e. List all string starts with given letter
  - f. List of all string contains the Substring
  - g. Sort the elements in ArrayList
  - h. Remove a particular element
  - i. Replace one string with another string in ArrayList
  - j. Remove duplicate elements
2. Write a program to get two integer arraylist. Perform the following operations

  - a. Merge the two lists
  - b. Find Union of two lists
  - c. Find Intersection of two lists
  - d. Compare two lists
3. Using Collection framework, create a doubly linked list of integers and perform the following operations.

  - a. Insert element on both sides
  - b. Delete element on both sides
  - c. Insert an element at a particular position
  - d. Delete a particular element
  - e. Search for a particular element
  - f. Display list in forward order and backward order
  - g. Sort the elements in LinkedList
  - h. Replace one element in the list with another list
  - i. Remove duplicate elements

## Exercise - 8 - Collection Framework

### Code:

```
import java.util.*;  
  
public class StringList{  
  
    static ArrayList<String> alist = new ArrayList<String>();  
  
    static void displayByLetter(char c) {  
        for (String s : alist) { //using for each loop to iterate  
            if (s.charAt(0) == c) {  
                System.out.print(s + "\t");  
            }  
        }  
    }  
  
    static void subList(String elt) {  
        int flag = -1;  
        for (String s : alist) {  
            for (int i = 0; i < s.length(); i++) {  
                if (s.charAt(i) == elt.charAt(0)) {  
                    // System.out.print(s.substring(i,i+elt.length()));  
                    if (s.substring(i, i + elt.length()).equals(elt)) {  
                        flag = 1;  
                        break;  
                    } else {  
                        continue;  
                    }  
                } else {  
                    flag = 0;  
                }  
            }  
            // System.out.print(flag);  
            if (flag == 1) {  
                System.out.print(s + "\t");  
            }  
        }  
    }  
  
    static void removeDuplicates() {  
        Collections.sort(alist);  
        for (int i = 1; i < alist.size(); i++) {  
            if (alist.get(i).equals(alist.get(i - 1))) { //comparing prev  
                alist.remove(alist.get(i));  
            }  
        }  
    }  
}
```

```

static int searchString(String elt) {
    int pos = -1;
    for (int i = 0; i < alist.size(); i++) {
        if (elt.equals(alist.get(i))) {
            pos = i;
            //System.out.println(pos);
            break;
        }
    }
    return pos;
}

static void display()
{
    for(String s:alist)
    {
        System.out.print(s+"\t");
    }
    System.out.println();
}

private static Scanner sc = new Scanner(System.in);

public static void main(String args[])
{
    int n,pos;
    String elt;
    System.out.println("MENU\n1.Insert end\n2.Insert at a position\n3.List strings that start with a letter\n4.List of all strings which contain a sub string\n5.Sort\n6.Remove an element\n7.Replace\n8.Remove duplicates\n9.Search for an element\n10.Exit\n");
    do {
        System.out.println();
        System.out.print("\nEnter option: ");
        n = sc.nextInt();
        sc.nextLine();
        switch (n) {
            case 1:
                System.out.print("Enter string to insert at end: ");
                elt=sc.nextLine();
                alist.add(elt);
                display();
                break;

            case 2:
                System.out.print("Enter position: ");
                pos = sc.nextInt();
                sc.nextLine();
                System.out.print("Enter string: ");
                elt = sc.nextLine();
        }
    }
}

```

```
        alist.add(pos, elt);
        display();
        break;

    case 3:
        System.out.print("Enter starting letter: ");
        char c = sc.nextLine().charAt(0);
        displayByLetter(c);
        break;

    case 4:
        System.out.print("Enter substring: ");
        elt = sc.nextLine();
        subList(elt);
        break;

    case 5:
        Collections.sort(alist);
        display();
        break;

    case 6:
        System.out.print("Enter element to remove: ");
        elt=sc.nextLine();
        alist.remove(elt);
        display();
        break;

    case 7:
        System.out.print("Enter position: ");
        pos=sc.nextInt();
        sc.nextLine();
        System.out.print("Enter string: ");
        elt=sc.nextLine();
        alist.set(pos,elt);
        display();
        break;

    case 8:
        removeDuplicates();
        display();
        break;

    case 9:
        System.out.print("Enter string to search: ");
        elt = sc.nextLine();
        pos=searchString(elt);
        if (pos != -1) {
            System.out.println("FOUND AT " + pos);
        } else {
```

```
        System.out.println("NOT FOUND");
    }
    break;
}
}while(n!=10);
}
}
```

### **Output:**

MENU

- 1.Insert end
- 2.Insert at a position
- 3.List strings that start with a letter
- 4.List of all strings which contain a substring
- 5.Sort
- 6.Remove an element
- 7.Replace
- 8.Remove duplicates
- 9.Search for an element
- 10.Exit

Enter option: 1

Enter string to insert at end: cat

cat

Enter option: 1

Enter string to insert at end: dog

cat dog

Enter option: 1

Enter string to insert at end: zebra

cat      dog      zebra

Enter option: 1

Enter string to insert at end: deer

cat      dog      zebra      deer

Enter option: 1

Enter string to insert at end: parrot

cat      dog      zebra      deer      parrot

Enter option: 2

Enter position: 3

Enter string: sheep

cat      dog      zebra      sheep      deer      parrot

Enter option: 3

Enter starting letter: d

dog      deer

Enter option: 4

Enter substring: ee

sheep      deer

Enter option: 5

cat      deer      dog      parrot    sheep    zebra

Enter option: 6

Enter element to remove: zebra

cat      deer      dog      parrot    sheep

Enter option: 7

Enter position: 3

Enter string: cat

cat      deer      dog      cat      sheep

Enter option: 8

cat      deer      dog      sheep

Enter option: 9

Enter string to search: sheep

FOUND AT 3

Enter option: 9

Enter string to search: parrot

NOT FOUND

Enter option: 10

## Collection



### StringList

```
~alist : ArrayList<String>
~displayByLetter(char c) : void
~subList(String elt): void
~removeDuplicates(): void
~searchString(String elt) :int
~ display():void
```

### Code:

```
import java.util.*;

public class IntegerList {

    static ArrayList<Integer> i1 = new ArrayList<Integer>();
    static ArrayList<Integer> i2 = new ArrayList<Integer>();
    static ArrayList<Integer> i3 = new ArrayList<Integer>();

    static void mergeList() {
        i3.addAll(i1);
        i3.addAll(i2);
    }

    static void unionList() {
        i3.addAll(i1);
        for(int i:i2)
        {
            if(!i3.contains(i))      //eliminating repeated elements
            {
                i3.add(i);
            }
        }
    }

    static void interList()
    {
        for(Integer i:i1)
        {
            if(i2.contains(i) && !i3.contains(i)) //rmv repeated elements
            {
                i3.add(i);
            }
        }
    }
}
```

```

static void compareList()
{
    int f=0;
    if(i1.size()==i2.size()) //comparing sizes else comparing elements
    {
        for(int i=0;i<i1.size();i++)
        {
            if(!i1.get(i).equals(i2.get(i)))
            {
                f=0;
            }
        }
        f=1;
    }
    if(f==1)
    {
        System.out.println("Equal lists");
    }
    else
    {
        System.out.println("Non equal lists");
    }
}

static void display(ArrayList<Integer> i)
{
    for(Integer j:i)
    {
        System.out.print(j+"\t");
    }
    System.out.println();
}

private static Scanner sc = new Scanner(System.in);
public static void main(String [] args)
{
    int n;
    System.out.println("MENU\n1.Insert to list 1\n2.Insert to list 2\n3.Merge\n4.Union\n5.Intersection\n6.Compare\n7.Exit\n");
    do
    {
        System.out.print("\nEnter option: ");
        n = sc.nextInt();
        i3.clear();           //to modify i3 each time
        switch(n)
        {
            case 1:
                System.out.print("Enter integer to insert to list 1: ");
                i1.add(sc.nextInt());
                System.out.println("List i1");

```

```

        display(i1);
        break;

        case 2:
        System.out.print("Enter integer to insert to list 2: ");
        i2.add(sc.nextInt());
        System.out.println("List i2");
        display(i2);
        break;

        case 3:
        mergeList();
        System.out.println("Merged list");
        display(i3);
        break;

        case 4:
        unionList();
        System.out.println("Union");
        display(i3);
        break;

        case 5:
        interList();
        System.out.println("Intersection");
        display(i3);
        break;

        case 6:
        compareList();
        break;

    }
}while(n!=7);
}
}

```

### Output:

MENU

1.Insert to list 1

2.Insert to list 2

3.Merge

- 4.Union
- 5.Intersection
- 6.Compare
- 7.Exit

Enter option: 1

Enter integer to insert to list 1: 3

List i1

3

Enter option: 1

Enter integer to insert to list 1: 4

List i1

3        4

Enter option: 1

Enter integer to insert to list 1: 6

List i1

3        4        6

Enter option: 1

Enter integer to insert to list 1: 8

List i1

3        4        6        8

Enter option: 1

Enter integer to insert to list 1: 7

List i1

3        4        6        8        7

Enter option: 1

Enter integer to insert to list 1: 0

List i1

3        4        6        8        7        0

Enter option: 1

Enter integer to insert to list 1: 2

List i1

3        4        6        8        7        0        2

Enter option: 2

Enter integer to insert to list 2: 4

List i2

4

Enter option: 2

Enter integer to insert to list 2: 7

List i2

4        7

Enter option: 2

Enter integer to insert to list 2: 0

List i2

4      7      0

Enter option: 2

Enter integer to insert to list 2: 2

List i2

4      7      0      2

Enter option: 2

Enter integer to insert to list 2: 1

List i2

4      7      0      2      1

Enter option: 2

Enter integer to insert to list 2: 9

List i2

4      7      0      2      1      9

Enter option: 3

Merged list

3      4      6      8      7      0      2      4      7      0  
2      1      9

Enter option: 4

Union

3      4      6      8      7      0      2      1      9

Enter option: 5

Intersection

4            7            0            2

Enter option: 6

Non equal lists

Enter option: 7

---

MENU

1.Insert to list 1

2.Insert to list 2

3.Merge

4.Union

5.Intersection

6.Compare

7.Exit

Enter option: 1

Enter integer to insert to list 1: 3

List i1

3

Enter option: 1

Enter integer to insert to list 1: 4

List i1

3 4

Enter option: 1

Enter integer to insert to list 1: 6

List i1

3 4 6

Enter option: 1

Enter integer to insert to list 1: 8

List i1

3 4 6 8

Enter option: 1

Enter integer to insert to list 1: 9

List i1

3 4 6 8 9

Enter option: 2

Enter integer to insert to list 2: 3

List i2

3

Enter option: 2

Enter integer to insert to list 2: 4

List i2

3        4

Enter option: 2

Enter integer to insert to list 2: 6

List i2

3        4        6

Enter option: 2

Enter integer to insert to list 2: 8

List i2

3        4        6        8

Enter option: 2

Enter integer to insert to list 2: 9

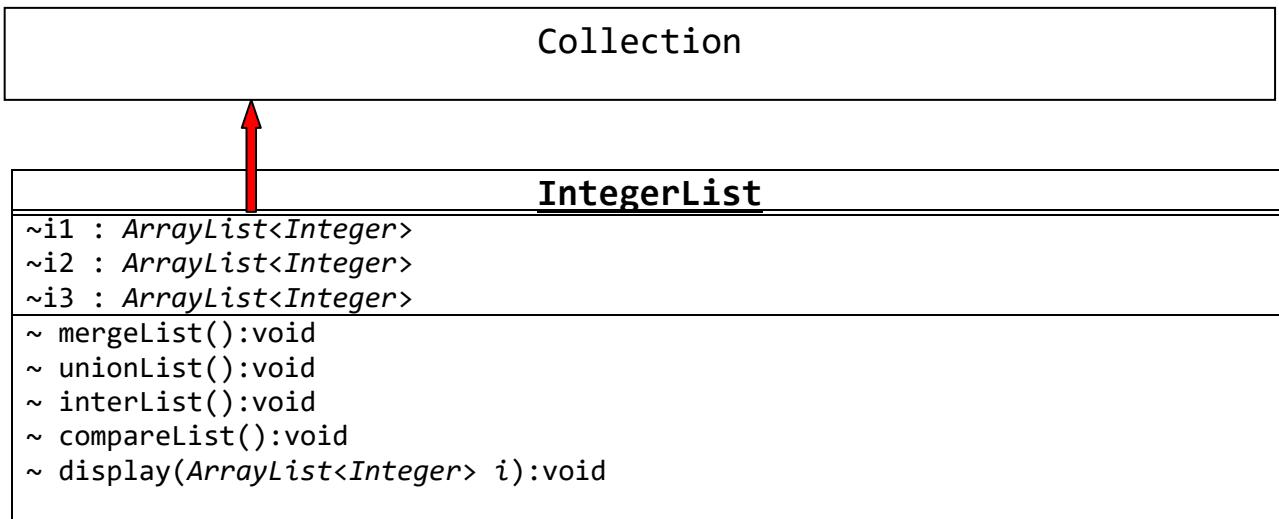
List i2

3        4        6        8        9

Enter option: 6

Equal lists

Enter option: 7



### Code:

```

import java.util.*;

public class Doublell {

    static LinkedList<Integer> linklist = new LinkedList<Integer>();
    static void displayForw()
    {
        for(int s:linklist)
        {
            System.out.print(s+"\t");
        }
        System.out.println();
    }

    static void displayBack()
    {
        Collections.reverse(linklist);      //reversing a linked list
        for(int s:linklist)
        {
            System.out.print(s+"\t");
        }
        System.out.println();
    }

    static void removeDuplicates()
    {
        Collections.sort(linklist);
        for (int i = 1; i < linklist.size(); i++) {
            if (linklist.get(i).equals(linklist.get(i - 1))) {

```

```

        linklist.remove(linklist.get(i));
    }
}

static void removeElt(int j)
{
    int pos=-1;
    for(int i=0;i<linklist.size();i++)
    {
        if(linklist.get(i).equals(j)) //if element in linklist equals elt
        {
            pos=i;
            break;
        }
    }
    if(pos!=-1)
    {
        linklist.remove(pos);
    }
    else
    {
        System.out.println("Not found");
    }
}

private static Scanner sc = new Scanner(System.in);

public static void main(String args[])
{
    int n;
    System.out.println("MENU\n1.Insert front\n2.Insert end\n3.Delete firs
t\n4.Delete end\n5.Insert at position\n6.Delete element\n7.Search\n8.Display
forward\n9.Display backward\n10.Sort\n11.Replace\n12.Remove duplicate\n13.Exi
t");
    do
    {
        System.out.print("\nEnter option: ");
        n = sc.nextInt();
        switch(n)
        {
            case 1:
                System.out.print("Enter element: ");
                linklist.addFirst(sc.nextInt());
                displayForw();
                break;

            case 2:
                System.out.print("Enter element: ");
                linklist.add(sc.nextInt());
        }
    }
}

```

```
        displayForw();
        break;

    case 3:
        System.out.println("Deleted element: "+linklist.removeFirst())
    );
        displayForw();
        break;

    case 4:
        System.out.println("Deleted element: "+linklist.removeLast())
    ;
        displayForw();
        break;

    case 5:
        System.out.print("Enter position and element: ");
        linklist.add(sc.nextInt(),sc.nextInt());
        displayForw();
        break;

    case 6:
        System.out.print("Enter element: ");
        removeElt(sc.nextInt());
        displayForw();
        break;

    case 7:
        System.out.print("Enter element: ");
        System.out.println("Found: "+linklist.contains(sc.nextInt()))
    ;
        break;

    case 8:
        displayForw();
        break;

    case 9:
        displayBack();
        break;

    case 10:
        Collections.sort(linklist);
        displayForw();
        break;

    case 11:
        System.out.print("Enter position and element: ");
        linklist.set(sc.nextInt(),sc.nextInt());
        displayForw();
```

```
        break;

    case 12:
        removeDuplicates();
        displayForw();
        break;
    }
}while(n!=13);
}
```

### **Output:**

MENU

- 1.Insert front
- 2.Insert end
- 3.Delete first
- 4.Delete end
- 5.Insert at position
- 6.Delete element
- 7.Search
- 8.Display forward
- 9.Display backward
- 10.Sort
- 11.Replace
- 12.Remove duplicate
- 13.Exit

Enter option: 1

Enter element: 3

3

Enter option: 1

Enter element: 5

5 3

Enter option: 1

Enter element: 2

2 5 3

Enter option: 1

Enter element: 6

6 2 5 3

Enter option: 1

Enter element:

9

9 6 2 5 3

Enter option: 1

Enter element: 0

0 9 6 2 5 3

Enter option: 1

Enter element: 7

7 0 9 6 2 5 3

Enter option: 2

Enter element: 4

7 0 9 6 2 5 3 4

Enter option: 2

Enter element: 7

7 0 9 6 2 5 3 4 7

Enter option: 2

Enter element: 9

7 0 9 6 2 5 3 4 7 9

Enter option: 3

Deleted element: 7

0 9 6 2 5 3 4 7 9

Enter option: 4

Deleted element: 9

0 9 6 2 5 3 4 7

Enter option: 5

Enter position and element: 2 3

0 9 3 6 2 5 3 4 7

Enter option: 6

Enter element: 0

9        3        6        2        5        3        4        7

Enter option: 7

Enter element: 2

Found: true

Enter option: 7

Enter element: 0

Found: false

Enter option: 8

9        3        6        2        5        3        4        7

Enter option: 9

7        4        3        5        2        6        3        9

Enter option: 10

2        3        3        4        5        6        7        9

Enter option: 11

Enter position and element: 6 4

2        3        3        4        5        6        4        9

Enter option: 12

2        3        4        5        6        9

Enter option: 13

