

## **Lab Cycle 1**

### **Experiment No : 1**


**Date : 13/04/2022**

AIM : Write a program to display the message “WELCOME JAVA”.

#### SOURCE CODE :

```
class Welcome{  
    public static void main(String args[]){  
        System.out.println("Welcome java");  
    }  
}
```

#### OUTPUT :



```
21mca14@user:~$ javac Welcome.java  
21mca14@user:~$ java Welcome  
Welcome java
```

#### RESULT :

Program is successfully executed and output is verified.

**Experiment No : 2****Date : 13/04/2022**

AIM : Create a class Rectangle with instance variable length and breadth. Define a method setData for setting values of instance variables and a method getArea to return the area of Rectangle using the class. Find the area of the rectangle using the values length = 12.48 and breadth = 13.

ALGORITHM :

Step 1 : Define a class Rectangle with instance variables length and breadth.

Step 2 : Define methods setData() and getArea() as member functions of Rectangle class to set the values of instance variables and to calculate the area respectively.

Step 3 : Create an object to access the members of Rectangle class.

Step 4 : Invoke the method setData(12.48,13).

Step 5 : Invoke method getArea().

Step 6 : Print area of the rectangle.

SOURCE CODE :

```
public class Rectangle {  
    double length, breadth;  
    void setData(double l,double b) {  
        length = l; breadth = b;  
    }  
    double getArea() {  
        return length*breadth;  
    }  
}  
class Find_area{  
    public static void main(String args[]) {  
        Rectangle r = new Rectangle();  
        r.setData(12.48,13);  
        System.out.println("Area of Rectangle: "+ r.getArea());  
    }  
}
```

OUTPUT :

```
21mca14@user:~$ javac Rectangle.java  
21mca14@user:~$ java Rectangle  
Area of Rectangle: 162.24
```

RESULT :

Program is successfully executed and output is verified.

**Experiment No : 3****Date : 20/04/2022**

AIM : Write a program to read integer from keyboard and check whether the number is even or odd.


ALGORITHM :

- Step 1 : Import the Java 'Scanner' class.  
Step 2 : Read an integer (num) from user.  
Step 3 : If  $\text{num} \% 2 == 0$  , then print "num is even".  
Step 4 : Else, print "num id odd".

SOURCE CODE :

```
import java.util.Scanner;

class OddEven{
    public static void main(String args[]){
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the number:");
        int num = sc.nextInt();
        if(num%2==0)
            System.out.println(num + " is a Even number");
        else
            System.out.println( num + " is an Odd ");
    }
}
```

OUTPUT :

```
21mca14@user:~$ javac OddEven.java
21mca14@user:~$ java OddEven
Enter the number:
34
34 is a Even number
21mca14@user:~$ java OddEven
Enter the number:
33
33 is a Odd
```

RESULT :

Program is successfully executed and output is verified.

**Experiment No : 4****Date : 20/04/2022**

AIM : Define a class 'product' with data members pcode,pname and price. Create three objects of the class and find the product having the lowest price.

**ALGORITHM :**

Step 1 : Define a class Product with data members pcode,pname,price.

Step 2 : Read the details (product code,name,price,..etc) of three products from the user.

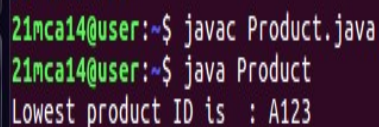
Step 3 : Create three objects for Product class and initialize it with the details read from the user.

Step 4 : Compare the prices of three products.

Step 5 : Print the product which has the lowest price.

**SOURCE CODE :**

```
import java.util.Scanner;
public class Product {
    String pcode, pname; int price;
    Product(String pcode_get, String pname_get, int price_get) {
        pcode = pcode_get; pname = pname_get; this.price = price_get;
    }
    void compare(Product p1,Product p2){
        if (price <= p1.price && price <= p2.price)
            System.out.println("Lowest product ID is : " + pcode);
        if (p2.price <= price && p2.price <= price)
            System.out.println("Lowest product ID is : " + p2.pcode);
        if (p1.price <= p2.price && p1.price <= price)
            System.out.println("Lowest product ID is : " + p1.pcode);
    }
}
class Product_details{
    public static void main(String[] args) {
        Product p_1 = new Product("A123", "TV", 2);
        Product p_2 = new Product("B123", "RADIO", 21);
        Product p_3 = new Product("C123", "DVD", 3);
        p_1.compare(p_2,p_3);}
}
```

**OUTPUT :**

```
21mca14@user:~$ javac Product.java
21mca14@user:~$ java Product
Lowest product ID is : A123
```

**RESULT :**

Program is successfully executed and output is verified.

**Experiment No : 5****Date : 22/04/2022**

AIM : Read two matrices from the console and perform matrix addition.

ALGORITHM :

Step 1 : Read the count of total number of elements in each matrix.

Step 2 : Declare two 2D array.

Step 3 : Read the matrices and store it into the array.

Step 4 : Perform addition between the elements of the two arrays which has the same index and store the result in another array.

Step 5 : Display the resultant matrix.

SOURCE CODE :

```
import java.util.Scanner;

class AddMatrix{
    public static void main(String args[]){
        int i,j,rows,cols;
        Scanner n=new Scanner(System.in);
        System.out.println("Enter the no of rows: ");
        rows=n.nextInt();
        System.out.println("Enter the no of cols: ");
        cols=n.nextInt();
        int A[][]= new int[rows][cols];
        int B[][]=new int[rows][cols];
        System.out.println("Enter the elements of Matrix A: ");
        for(i=0;i<rows;i++){
            for(j=0;j<cols;j++){
                A[i][j]=n.nextInt();
            }
        }
        System.out.println("Enter the elements of Matrix B: ");
        for(i=0;i<rows;i++){
            for(j=0;j<cols;j++){
                B[i][j]=n.nextInt();
            }
        }
        int C[][]=new int[rows][cols];
        System.out.println(" The sum of Matrix A and B: ");
        for(i=0;i<rows;i++){
            for(j=0;j<cols;j++){
                C[i][j]=A[i][j]+B[i][j];
                System.out.print(C[i][j]+" ");
            }
        }
    }
}
```

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

#### OUTPUT :

```
21mca14@user:~$ javac AddMatrix.java  
21mca14@user:~$ java AddMatrix  
Enter the no of rows  
2  
Enter the no of cols  
3  
Enter the elements of Matrix A  
1  
2  
3  
4  
5  
6  
Enter the elements of Matrix B  
6  
5  
4  
3  
2  
1  
the sum of Matrix A and B  
7 7 7  
7 7 7
```

#### RESULT :

Program is successfully executed and output is verified.

**Experiment No : 6****Date : 22/04/2022**

AIM : Write a program to perform complex number addition.

ALGORITHM :

Step 1 : Read the two complex numbers

Step 2 : Perform addition of real parts as well as imaginary parts of two numbers. Store the result.

Step 3 : Print the result.

SOURCE CODE :

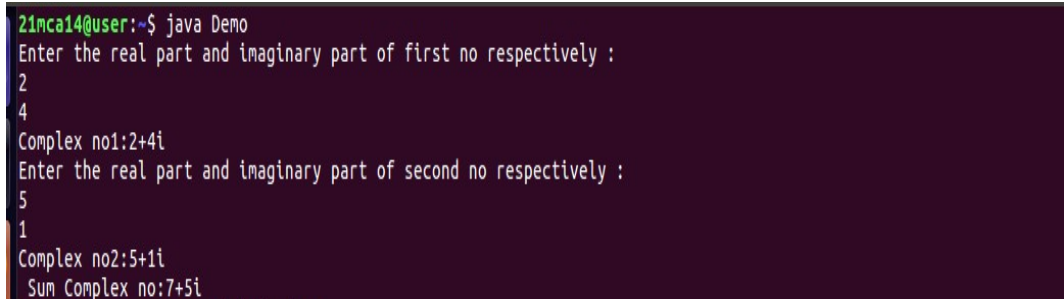
```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
class Complex{
    int real,img;
    Complex(int treal,int timg){
        real=treale;
        img=timg;
    }
    Complex add_complexno(Complex c1,Complex c2){
        Complex temp = new Complex();
        temp.real=c1.real+c2.real;
        temp.img=c1.img+c2.img;
        return temp;
    }
}
public class Demo{
    public static void main(String args[]) throws IOException{
        BufferedReader r =new BufferedReader(new
            InputStreamReader(System.in));
        System.out.println("Enter the real part and imaginary part of first no
            respectively : ");
```

```

        int a1=Integer.parseInt(r.readLine());
        int b1=Integer.parseInt(r.readLine());
        Complex c1 = new Complex(a1,b1);
        System.out.println("Complex no 1 : " +c1.real + "+" + c1.img + "i");
        System.out.println("Enter the real part and imaginary part of second no
        respectively : ");
        int a2=Integer.parseInt(r.readLine());
        int b2=Integer.parseInt(r.readLine());
        Complex c2 = new Complex(a2,b2);
        System.out.println("Complex no 2 : "+ c2.real + "+" + c2.img + "i");
        Complex c3 = new Complex();
        c3 = c3.add_complexno(c1,c2);
        System.out.println(" sum complex no: " + c3.real + "+" + c3.img + "i");
    }
}

```

#### OUTPUT :



```

21mca14@user:~$ java Demo
Enter the real part and imaginary part of first no respectively :
2
4
Complex no1:2+4i
Enter the real part and imaginary part of second no respectively :
5
1
Complex no2:5+1i
Sum Complex no:7+5i

```

#### RESULT :

Program is successfully executed and output is verified.



**Experiment No : 7**

**Date : 27/04/2022**

AIM : Read a matrix from the console and check whether it is symmetric/not

ALGORITHM :

Step 1 : Read the order of the matrix.

Step 2 : If the number of rows and columns are not same, then print matrix is not symmetric and exit.

Step 3 : Else, read the matrix.

Step 4 : Find the transpose of the matrix by interchanging rows and columns.

Step 5 : Check if the original matrix is same as its transpose. If it is same, then print matrix is symmetric. Else, print matrix is not symmetric.

SOURCE CODE :

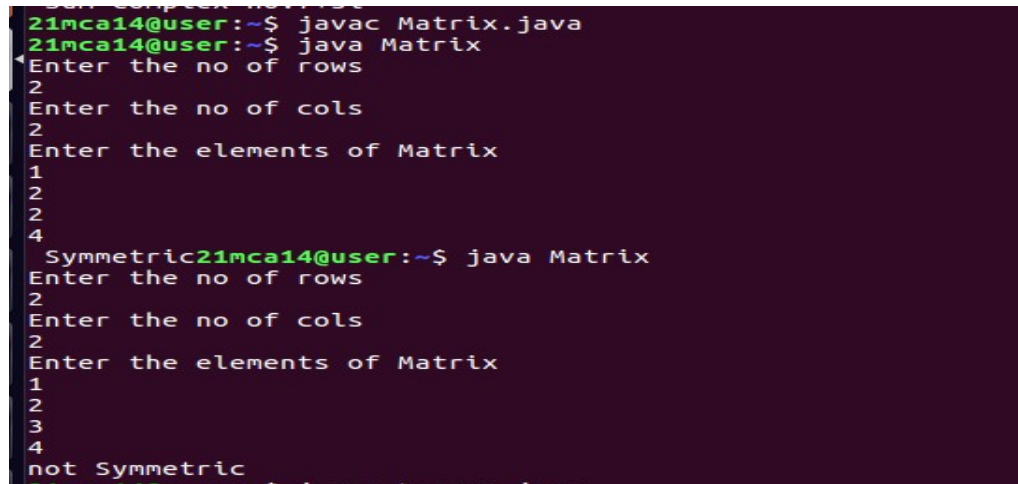
```
import java.util.Scanner;
class Matrix{
    public static void main(String args[]){
        int i,j,rows,cols,f=0;
        Scanner n=new Scanner(System.in);
        System.out.println("Enter the no of rows: ");
        rows=n.nextInt();
        System.out.println("Enter the no of cols: ");
        cols=n.nextInt();
        if(rows!=cols)
            System.out.print(" Not symmetric");
        else{
            int num[][]= new int[rows][cols];
            System.out.println("Enter the elements of Matrix: ");
            for(i=0;i<rows;i++){
                for(j=0;j<cols;j++){
                    num[i][j]=n.nextInt();
                }
            }
            for(i=0;i<rows;i++){
                for(j=0;j<cols;j++){
                    if(num[i][j]!=num[j][i]){
                        f=1;
                        break;
                    }
                }
            }
            if(f==1)
                System.out.println("Not Symmetric");
        }
    }
}
```

```

        else
            System.out.print(" Symmetric");
    }
}

```

#### OUTPUT :



```

21mca14@user:~$ javac Matrix.java
21mca14@user:~$ java Matrix
Enter the no of rows
2
Enter the no of cols
2
Enter the elements of Matrix
1
2
2
4
Symmetric
21mca14@user:~$ java Matrix
Enter the no of rows
2
Enter the no of cols
2
Enter the elements of Matrix
1
2
3
4
not Symmetric

```

#### RESULT :

Program is successfully executed and output is verified.

**Experiment No : 8****Date : 27/04/2022**

AIM : Write a program to print the leap years within the given range.

ALGORITHM :

Step 1 : Read the beginning and end years.

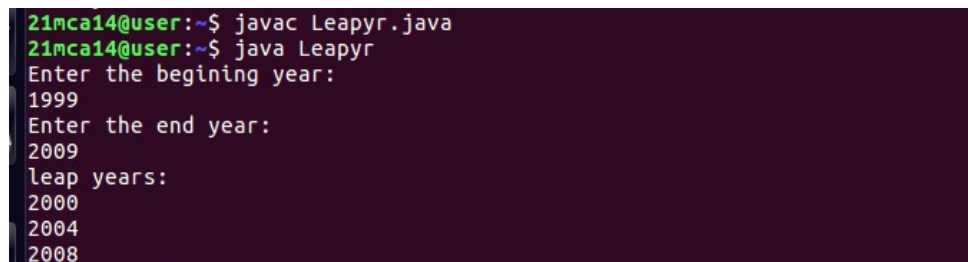
Step 2 : Repeat step3 between the beginning and end years.

Step 3 : Check if the year is divisible by 4 or not by 100 and check if the year is divisible by 400. If either of them true, then print the year .

SOURCE CODE :

```
import java.util.Scanner;
class Leapyr{
    public static void main(String args[]){
        Scanner sc=new Scanner(System.in);
        int start,end;
        System.out.println("Enter the beginning year: ");
        start = sc.nextInt();
        System.out.println("Enter the end year: ");
        end = sc.nextInt();
        System.out.println("Leap years: ");
        for(int i=start;i<=end;i++){
            if(i%4==0||(i%100!=0)&&(i%400==0))
                System.out.println(i);
        }
    }
}
```

OUTPUT :



```
21mca14@user:~$ javac Leapyr.java
21mca14@user:~$ java Leapyr
Enter the beginning year:
1999
Enter the end year:
2009
leap years:
2000
2004
2008
```

RESULT :

Program is successfully executed and output is verified.

**Experiment No : 9****Date : 27/04/2022**

AIM : Create CPU with attribute price. Create an inner class processor (no of cores,manufacturer)and static nested class RAM(memory,manufacturer).Create an object of CPU and print information of processor and RAM.

ALGORITHM :

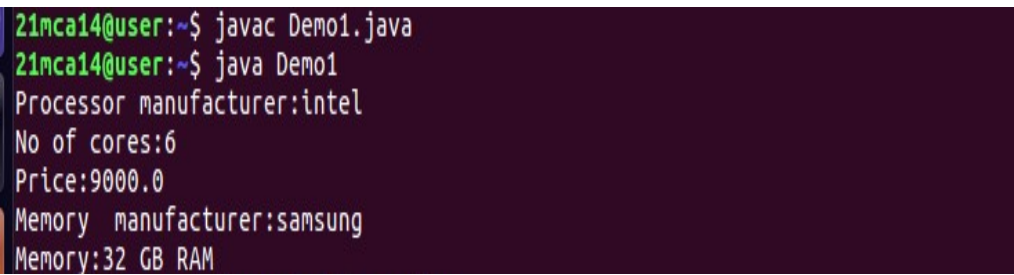
- Step 1 : Define a class CPU with data member price. Define an inner class called Processor with data members to hold the specifications of processor such as no.of cores, manufacturer. Also define a static class RAM(use static keyword) with member variables to hold the information about the memory like memory size, manufacturer etc.
- Step 2 : Initialize member variables of classes CPU,RAM and Processor.
- Step 3 : Create an object for CPU class and invoke a method to instantiate an object for inner class Processor then invoke another method to display the specification of the Processor.
- Step 4 : Create an object for static inner class RAM of Class CPU and invoke a method to display the specification of the class RAM

SOURCE CODE :

```
class CPU{
    double price=9000;
    void display_1(){
        Processor obj_2=new Processor();
        obj_2.display_2();
    }
    class Processor{
        int no_of_cores=6;
        String pross_manufact="intel";
        void display_2(){
            System.out.println("Processor manufacturer:" + pross_manufact);
            System.out.println("No of cores:" + no_of_cores);
            System.out.println("Price:" + price);
        }
    }
    static class RAM{
        String memory="32 GB RAM";
        String mmry_manufact="samsung";
        void display(){
            System.out.println("Memory manufacturer:" + mmry_manufact);
            System.out.println("Memory:" + memory);
        }
    }
}
```

```
}  
class Demo1{  
    public static void main(String args[]){  
        CPU obj1=new CPU();  
        obj1.display_1();  
        CPU.RAM obj=new CPU.RAM();  
        obj.display();  
    }  
}
```

OUTPUT :

A terminal window with a dark purple background and light green text. It shows the compilation and execution of a Java program. The commands and their outputs are as follows:  
21mca14@user:~\$ javac Demo1.java  
21mca14@user:~\$ java Demo1  
Processor manufacturer:intel  
No of cores:6  
Price:9000.0  
Memory manufacturer:samsung  
Memory:32 GB RAM  
The prompt character is a green '21mca14@user:~\$' and the output text is light green.

```
21mca14@user:~$ javac Demo1.java  
21mca14@user:~$ java Demo1  
Processor manufacturer:intel  
No of cores:6  
Price:9000.0  
Memory manufacturer:samsung  
Memory:32 GB RAM
```

RESULT :

Program is successfully executed and output is verified.

## Lab Cycle 2

### Experiment No : 10

Date : 29/04/2022

AIM: Write a Program which accepts the marks of a student into a 1D array from the keyboard. Calculate and display total marks & percentage obtained by the student.

### ALGORITHM :

Step 1 : Read the number of subjects.

Step 2 : Declare a 1D array to store the marks of the student.

Step 3 : Read the marks and store it into the array.

Step 4 : Calculate the total marks by adding the entries in the array and find the percentage of the total marks.

Step 5 : Print the total marks and percentage.

### SOURCE CODE :

```
import java.util.Scanner;

public class MarkArray{

    public static void main(String[] args){

        int n;
        float percentage,total = 0;
        Scanner s = new Scanner(System.in);
        System.out.print("Enter no. of subject:");
        n = s.nextInt();
        int marks[] = new int[n];
        System.out.println("Enter marks out of 100:");
        for(int i = 0; i < n; i++){
            marks[i] = s.nextInt();
            total += marks[i];
        }
        int p=n*100;
        percentage=(total/p)*100;
        System.out.println("Sum:"+total);
        System.out.println("Percentage:"+percentage);
    }
}
```

### OUTPUT :

```
21mca14@user:~$ javac MarkArray.java
21mca14@user:~$ java MarkArray
Enter no. of subject:5
Enter marks out of 100:
78
79
89
99
80
Sum:425.0
Percentage:85.0
```

### RESULT :

Program is successfully executed and output is verified.

**Experiment No : 11****Date : 29/04/2022**

AIM : Program to sort strings.

ALGORITHM :

- Step 1 : Read the total number of strings.
- Step 2 : Declare a string array to hold the strings.
- Step 3 : Read the strings and store it into the array.
- Step 4 : Use nested loops to sort the strings in the array in alphabetical order.
- Step 5 : Use outer loop to hold the elements.
- Step 6 : Use inner loop to compare with the remaining elements. ( Use compareTo() )
- Step 7 : Swap the array elements.
- Step 8 : Print the sorted array.

SOURCE CODE :

```
import java.util.Scanner;

class Sortstr{
    public static void main(String args[]){
        int i,j,n;
        String temp;
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the no of strings");
        n=sc.nextInt();
        String str[]= new String[n];
        Scanner s=new Scanner(System.in);
        System.out.println("Enter the strings");
        for(i=0;i<n;i++){
            str[i]=s.nextLine();
        }
        for(i=0;i<n;i++){
            for(j=i+1;j<n;j++){
                if(str[i].compareTo(str[j])>0){
                    temp=str[i];
                    str[i]=str[j];
                    str[j]=temp;
                }
            }
        }
        System.out.println("Strings in sorted order");
        for(i=0;i<n;i++){
            System.out.println(str[i]);
        }
    }
}
```



### OUTPUT :

```
21mca14@user:~$ javac Sortstr.java
21mca14@user:~$ java Sortstr
Enter the no of strings
3
Enter the strings
Tara
Alia
Kiara
Stings in sorted order
Alia
Kiara
Tara
```

### RESULT :

Program is successfully executed and output is verified.

**Experiment No : 12****Date : 29/04/2022**

AIM : Program to sort characters from a string.

ALGORITHM :

Step 1 : Read the string.

Step 2 : Convert the given string into character array elements.( Use toCharArray() )

Step 3 : Use nested loops to sort the characters in the array.

Step 4 : Swap the array elements.

Step 5 : Convert the sorted character array back to string. ( Use valueOf() )

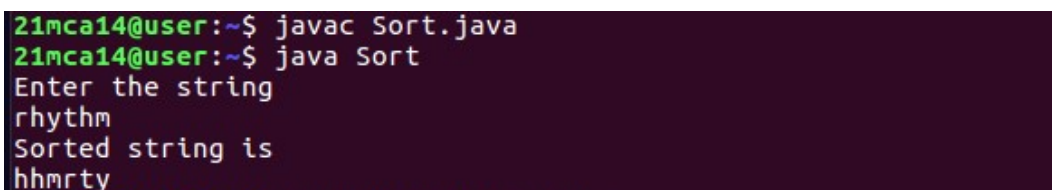
Step 6 : Print the sorted string.

SOURCE CODE :

```
import java.util.*;

class Sort{
    public static void main(String args[]){
        String temp;
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the string");
        temp=sc.nextLine();
        char str_arr[]=temp.toCharArray();
        Arrays.sort(str_arr);
        String sorted="";
        sorted=sorted.valueOf(str_arr);
        System.out.println("Sorted string is \n" +sorted);
    }
}
```

OUTPUT :



```
21mca14@user:~$ javac Sort.java
21mca14@user:~$ java Sort
Enter the string
rhythm
Sorted string is
hmrty
```

RESULT :

Program is successfully executed and output is verified.

**Experiment No : 13****Date : 4/05/2022**

AIM : Search an element in an array.

ALGORITHM :

- Step 1 : Read the total number of elements in the array.
- Step 2 : Declare an array to store the elements.
- Step 3 : Read the elements into the array.
- Step 4 : Read the element to be searched.
- Step 5 : Traverse through the array and check if search value matches the element.  
if it matches then print "value is found in respective position" else print "not found".

SOURCE CODE :

```
import java.util.Scanner;

class Mtrixsearch{
    public static void main(String args[]){
        int i,size,f=0;
        Scanner n=new Scanner(System.in);
        System.out.println("Enter the size of array");
        size=n.nextInt();
        int arr[]= new int[size];
        System.out.println("Enter the elements of array");
        for(i=0;i<size;i++){
            arr[i]=n.nextInt();
        }
        System.out.println("Enter the element to be searched");
        int a=n.nextInt();
        for(i=0;i<size;i++){
            if(arr[i]==a){
                f=1;
                break;
            }
        }
        if(f==1)
            System.out.println(a+" is found at position "+(i+1));
        else
            System.out.println("not found");
    }
}
```

### OUTPUT :

```
21mca14@user:~$ javac Mtrixsearch.java
21mca14@user:~$ java Mtrixsearch
Enter the size of array
5
Enter the elements of array
2
5
6
8
9
Enter the element to be searched
6
6 is found at position 3
```

### RESULT :

Program is successfully executed and output is verified.

**Experiment No : 14****Date : 4/05/2022**

AIM : Perform string manipulation (using Built-in methods of String Class and StringBuffer Class).

ALGORITHM :

Step 1 : Read two Strings str1 and str2.

Step 2 : Use various String manipulation built-in methods of String class

- a) concatenate 2 strings using concat() [str1.concat(str2)]
- b) find length of string using length() [str1.length()]
- c) convert string to uppercase using toUpperCase() [str1.toUpperCase()]
- d) replace a character by other using replace() [str1.replace('e','a')]
- e) convert a string to character array using toCharArray()
- f) find the index position of a substring using indexOf()

Step 3 : Use various String manipulation built-in methods of StringBuffer class

Step 4 : Create an object for StringBuffer class and initialize it with a string

- a) concatenate strings using append() [str.append(temp1)]
- b) find length of string using length() [str1.length()]
- c) read the position to insert a substring using insert() [str.insert(pos,temp2)]

SOURCE CODE :

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

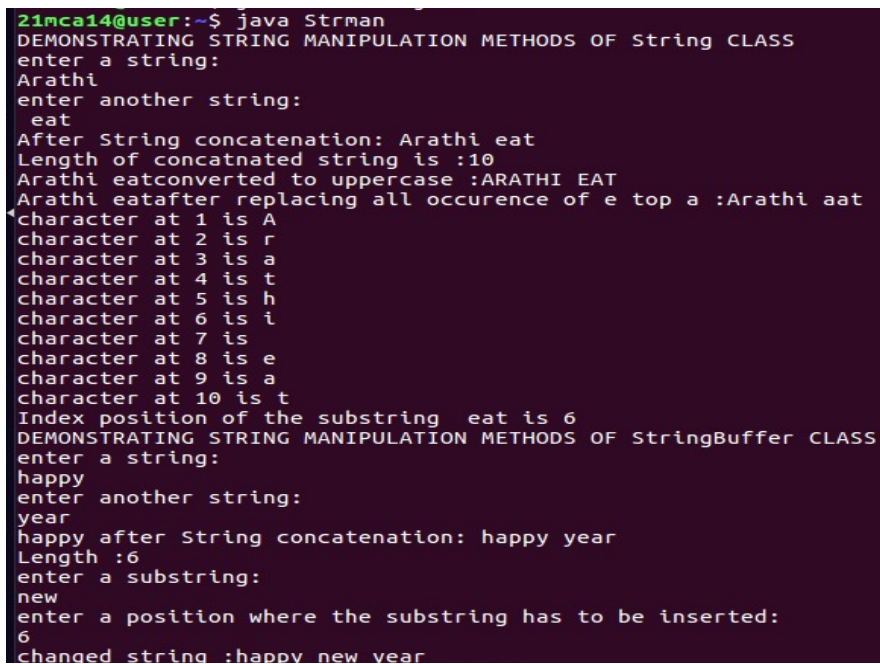
```
class Strman{
    public static void main(String args[]){
        Scanner sc =new Scanner(System.in);
        System.out.println("DEMONSTRATING STRING MANIPULATION
METHODS OF String CLASS");
        System.out.println("enter a string: ");
        String str1=sc.nextLine();
        System.out.println("enter another string: ");
        String str2=sc.nextLine();
        String str3=str1.concat(str2);
        System.out.println("After String concatenation: " + str3);
        System.out.println("Length of concatnated string is :"+ str3.length());
        System.out.println(str3+"converted to uppercase :"+ str3.toUpperCase());
        System.out.println(str3+"after replacing all occurence of e top a :"+
str3.replace('e','a'));
        char ch[]=str3.toCharArray();
        for(int i=0;i<ch.length;i++)
            System.out.println("character at "+(i+1)+" is "+ch[i]);
        System.out.println("Index position of the substring "+str2+" is "
+str3.indexOf(str2));
        System.out.println("DEMONSTRATING STRING MANIPULATION
METHODS OF StringBuffer CLASS");
```

```

        System.out.println("enter a string: ");
        String temp=sc.nextLine();
        StringBuffer str=new StringBuffer(temp);
        System.out.println("enter another string: ");
        String temp1=sc.nextLine();
        str.append(temp1);
        System.out.println(temp+"after String concatenation: " + str);
        System.out.println("Length :"+ str1.length());
        System.out.println("enter a substring: ");
        String temp2=sc.nextLine();
        System.out.println("enter a position where the substring has to be
        inserted: ");
        int pos =sc.nextInt();
        str.insert(pos,temp2);
        System.out.println("changed string :"+str);
    }
}

```

#### OUTPUT :



```

21mca14@user:~$ java Strman
DEMONSTRATING STRING MANIPULATION METHODS OF String CLASS
enter a string:
Arathi
enter another string:
eat
After String concatenation: Arathi eat
Length of concatnated string is :10
Arathi eatconverted to uppercase :ARATHI EAT
Arathi eatafter replacing all occurence of e top a :Arathi aat
character at 1 is A
character at 2 is r
character at 3 is a
character at 4 is t
character at 5 is h
character at 6 is i
character at 7 is
character at 8 is e
character at 9 is a
character at 10 is t
Index position of the substring eat is 6
DEMONSTRATING STRING MANIPULATION METHODS OF StringBuffer CLASS
enter a string:
happy
enter another string:
year
happy after String concatenation: happy year
Length :6
enter a substring:
new
enter a position where the substring has to be inserted:
6
changed string :happy new year

```

#### RESULT :

Program is successfully executed and output is verified.

**Experiment No : 15****Date : 4/05/2022**

AIM : Program to create a class for Employee having attributes eNo, eName,Salary.  
Read n employ information and search for an employee given eNo using the concept of array of objects.

**ALGORITHM :**

- Step 1 : Define a class Employee with instance variables eNo, eName, Salary.
- Step 2 : Define a parameterized constructor to initialize the instance variables.
- Step 3 : Read the count of total number of employees.
- Step 4 : Define an array of object , "obj".
- Step 5 : Read the details of each employee and store it in obj
- Step 6 : Read the eNo of the employee to be searched.
- Step 7 : Use loop to traverse through the array.
- Step 8 : Check whether the eNo to be searched is present in the array "obj".
- Step 9 : If yes, then print "Found". Else, print "Not found"

**SOURCE CODE :**

```
import java.util.*;

class Employee{
    int e_no;
    String e_name;
    float e_salary;
    Employee(int id,String name,float sal){
        e_no=id;
        e_name=name;
        e_salary=sal;
    }
}

class Main{
    public static void main(String args[]){
        Scanner sc = new Scanner(System.in);
        Scanner s = new Scanner(System.in);
        int id,i,flag=0;
        String name;
        float sal;
        System.out.println("enter the no.of employees: ");
        int n=sc.nextInt();
        Employee[] obj=new Employee[n];
        for(i=0;i<n;i++){
            System.out.println("Enter the details of employee: " + (i+1));
            System.out.println("employee no: ");
```

```

        id=sc.nextInt();
        System.out.println("employee name: ");
        name=s.nextLine();
        System.out.println("employee salary: ");
        sal=sc.nextFloat();
        obj[i]=new Employee(id,name,sal);
    }
    System.out.println("enter the employee id to be searched: ");
    int temp=sc.nextInt();
    for(i=0;i<n;i++){
        if(obj[i].e_no==temp){
            System.out.println("Result found!");
            flag=1;
            break;
        }
    }
    if(flag==1)
        System.out.println("Employee name: "+ obj[i].e_name);
    else
        System.out.println("not found" );
}
}

```

#### OUTPUT :

```

enter the no.of employees:
2
Enter the details of employee: 1
employee no:
1
employee name:
pooja
employee salary:
20000
Enter the details of employee: 2
employee no:
3
employee name:
Arya
employee salary:
25000
enter the employee id to be searched:
3
Result found!
Employee name: Arya

```

#### RESULT :

Program is successfully executed and output is verified.



### **Lab Cycle 3**

#### **Experiment No : 16**

**Date : 11/05/2022**

AIM : Using the concept of method overloading find the area of different shapes rectangle, circle and square.

#### ALGORITHM :

Step 1 : Define a class with three methods of same name but with different arguments in order to calculate the area of three different shapes such as rectangle, circle and square.

Step 2 : Read the particulars of the shapes and store it

Step 3 : Create three objects and invoke the methods defined in the step 1 by passing the stored values as parameters .

Step 4 : Display the calculated area of the three shapes.

#### SOURCE CODE :

```
import java.util.*;
class Area1{
    void area(int x){
        System.out.println("the area of the square is "+x*x+" sq units");
    }
    void area(int x, int y){
        System.out.println("the area of the rectangle is "+x*y+" sq units");
    }
    void area(double x){
        double z = 3.14 * x * x;
        System.out.println("the area of the circle is "+z+" sq units");
    }
}
class Areaoverload{
    public static void main(String args[]){
        Scanner sc = new Scanner(System.in);
        Area1 obj = new Area1();
        System.out.println("enter the side of square");
        int side = sc.nextInt();
        obj.area(side);
        System.out.println("enter the radius of circle");
        double r = sc.nextDouble();
        obj.area(r);
        System.out.println("enter the length and breadth of rectangle");
        int l = sc.nextInt();
        int b = sc.nextInt();
        obj.area(l,b);
    }
}
```

```
}  
}
```

OUTPUT :

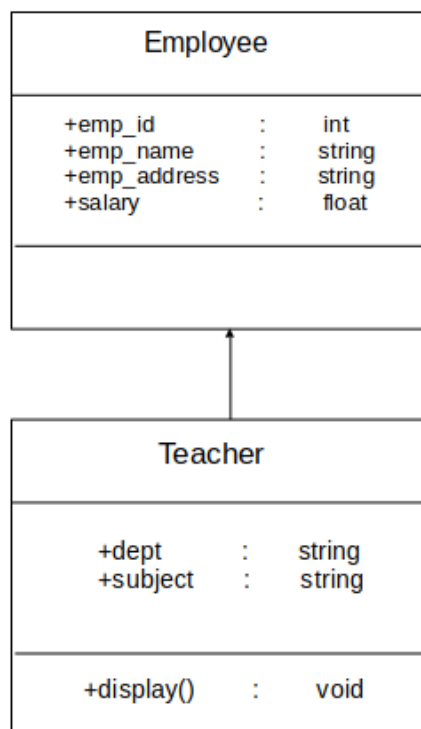
```
21mca14@user:~$ java Areaoverload  
enter the side of square  
4  
the area of the square is 16 sq units  
enter the radius of circle  
10  
the area of the circle is 314.0 sq units  
enter the length and breadth of rectangle  
3  
4  
the area of the rectangle is 12 sq units
```

RESULT :

Program is successfully executed and output is verified.

**Experiment No : 17****Date : 11/05/2022**

AIM : Create a class 'Employee' with data members Empid, Name, Salary, Address and constructors to initialize the data members. Create another class 'Teacher' that inherit the properties of class employee and contain its own data members department, Subjects taught and constructors to initialize these data members and also include display function to display all the data members. Use array of objects to display details of N teachers.

**CLASS DIAGRAM :****SOURCE CODE :**

```
import java.util.*;
class Employee{
    public int emp_id;
    public String emp_name,emp_address;
    public float salary;
    Employee(int id,String name,float sal,String add){
        emp_id=id;
        emp_name=name;
        salary=sal;
        emp_address=add;
    }
}
```

```

class Teacher extends Employee{
    public String dept,subject;
    Teacher(int id,String name,float sal,String add,String dep,String subj){
        super(id,name,sal,add);
        dept = dep;
        subject = subj;
    }
    public void display(){
        System.out.println("emp_id is"+emp_id);
        System.out.println("emp_name is"+emp_name);
        System.out.println("emp_salary is"+Salary);
        System.out.println("emp_address is"+emp_address);
        System.out.println("emp_separtment is"+dept);
        System.out.println("Teacher subjects taught is"+subject);
    }
}
class SimpleIn{
    public static void main(String args[]){
        int i;
        Scanner sc =new Scanner(System.in);
        Scanner s=new Scanner(System.in);
        System.out.println("enter the no of employee");
        int n=sc.nextInt();
        Teacher[] obj=new Teacher[n];
        for(i=0;i<n;i++){
            System.out.println("enter the details of Employee"+(i+1));
            System.out.println("Employee id:");
            int id=sc.nextInt();
            System.out.println("Employee name:");
            String name=s.nextLine();
            System.out.println("Employee Salary:");
            float sal=sc.nextFloat();
            System.out.println("Employee Address:");
            String add=s.nextLine();
            System.out.println("Teacher Department:");
            String dept=s.nextLine();
            System.out.println("Subject taught by teacher:");
            String subj=s.nextLine();
            obj[i] = new Teacher(id,name,sal,add,dept,subj);
        }
        for(i=0;i<n;i++)
            obj[i].display();
    }
}

```

### OUTPUT :

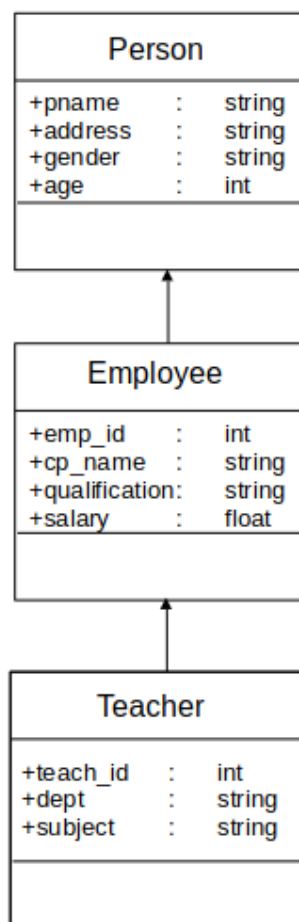
```
21mca14@user:~$ javac SimpleIn.java
21mca14@user:~$ java SimpleIn
enter the no of employee
2
enter the details of Employee1
Employee id:
1
Employee name:
Arathi
Employee Salary:
23400
Employee Address:
calicut
Teacher Department:
mca
Subject taught by teacher:
ACN
enter the details of Employee2
Employee id:
2
Employee name:
Pooja
Employee Salary:
30000
Employee Address:
palakkad
Teacher Department:
cs
Subject taught by teacher:
OOPS
emp_id is1
emp_name isArathi
emp_salary is23400.0
emp_address iscalicut
emp_separtment ismca
Teacher subjects taught isACN
emp_id is2
emp_name isPooja
emp_salary is30000.0
emp_address ispalakkad
emp_separtment iscs
Teacher subjects taught isOOPS
```

### RESULT :

Program is successfully executed and output is verified.

**Experiment No : 18****Date : 13/05/2022**

AIM : Create a class 'Person' with data members Name, Gender, Address, Age and a constructor to initialize the data members and another class 'Employee' that inherits the properties of class Person and also contains its own data members like Empid, Company name, Qualification, Salary and its own constructor. Create another class 'Teacher' that inherits the properties of class Employee and contains its own data members like Subject, Department, Teacher id and also contain constructors and methods to display the data members. Use array of objects to display details of N teachers.

CLASS DIAGRAM:SOURCE CODE :

```
import java.util.*;
class Person{
    public int age;
    public String pname,gender,address;
```

```

        Person(String name,String sex,String add,int age){
            pname=name;
            this.age=age;
            gender=sex;
            address=add;
        }
    }
    class Employee extends Person{
        public int emp_id;
        public String cp_name,qualification;
        public float salary;
        Employee(int id,String cname,float sal,String qualifctn,String name,String
            sex,String add,int age){
            super(name,sex,add,age);
            emp_id=id;
            cp_name=cname;
            salary=sal;
            qualification=qualifctn;
        }
    }
    class Teacher extends Employee{
        public int teach_id;
        public String dept,subject;
        Teacher(int id,String cname,float sal,String quali,String name,String sex,String
            add,int age,String dep,String subj,int tid){
            super(id,cname,sal,quali,name,sex,add,age);
            teach_id=tid;
            dept = dep;
            subject = subj;
        }
        public void display(){
            System.out.println();
            System.out.println("person_name is : "+pname);
            System.out.println("person_age is : "+age);
            System.out.println("person Gender is : "+gender);
            System.out.println("person Address is : " +address);
            System.out.println("emp_id is : "+emp_id);
            System.out.println("cp_name is : "+cp_name);
            System.out.println("emp_salary is : "+Salary);
            System.out.println("emp_Qualificatio is : "+qualification);
            System.out.println("teacher_id is : "+teach_id);
            System.out.println("emp_separtment is : "+dept);
            System.out.println("Teacher subjects taught is : "+subject);
        }
    }
    class Multilevel{

```

```

public static void main(String args[]){
    int i;
    Scanner sc =new Scanner(System.in);
    Scanner s =new Scanner(System.in);
    System.out.println("enter the no of Persons");
    int n=sc.nextInt();
    Teacher[] obj=new Teacher[n];
    for(i=0;i<n;i++){
        System.out.println("enter the details of Persons"+(i+1));
        System.out.println("Person name:");
        String name=s.nextLine();
        System.out.println("Person age:");
        int age=sc.nextInt();
        System.out.println("Person Gender:");
        String sex=s.nextLine();
        System.out.println("Person Address:");
        String add=s.nextLine();
        System.out.println("Employee id:");
        int id=sc.nextInt();
        System.out.println("Company name:");
        String cname=s.nextLine();
        System.out.println("Employee Salary:");
        float sal=sc.nextFloat();
        System.out.println("Employee Qualification:");
        String quali=s.nextLine();
        System.out.println("Teacher id:");
        int tid=sc.nextInt();
        System.out.println("Teacher Department:");
        String dep=s.nextLine();
        System.out.println("Subject taught by teacher:");
        String subj=s.nextLine();
        obj[i] = new Teacher
(id,cname,sal,quali,name,sex,add,age,dep,subj,tid);
    }
    for(i=0;i<n;i++)
        obj[i].display();
    }
}

```



### OUTPUT :

```
21mca14@user:~$ java Multilevel
enter the no of Persons
2
enter the details of Persons1
Person name:
Arathi
Person age:
34
Person Gender:
Female
Person Address:
calicut
Employee id:
23
Company name:
TCS
Employee Salary:
24000
Employee Qualification:
PG
Teacher id:
2
Teacher Department:
CS
Subject taught by teacher:
ACN
enter the details of Persons2
Person name:
Maya
Person age:
39
Person Gender:
Female
Person Address:
pala
Employee id:
33
Company name:
Infosys
Employee Salary:
34000
Employee Qualification:
PHP
Teacher id:
3
Teacher Department:
MCA
Subject taught by teacher:
DS
person_name is : Arathi
person_age is : 34
person Gender is :Female
person Address is :calicut
emp_id is :23
cp_name is :TCS
emp_salary is :24000.0
emp_Qualificatio is :PG
teacher_id is :2
emp_separment is :CS
Teacher subjects taught is :ACN

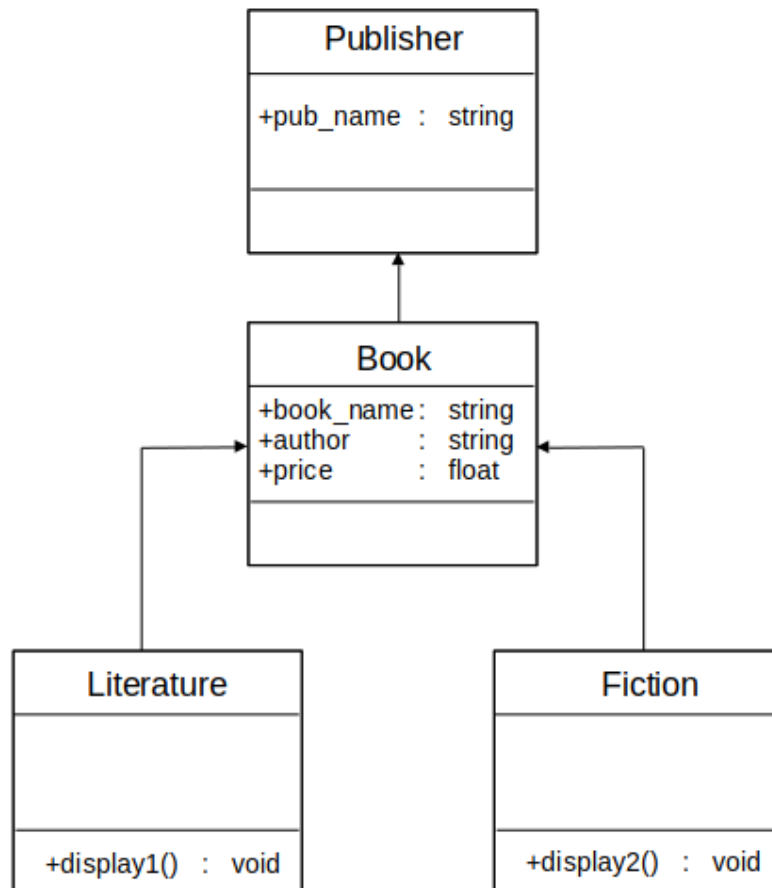
person_name is : Maya
person_age is : 39
person Gender is :Female
person Address is :pala
emp_id is :33
cp_name is :Infosys
emp_salary is :34000.0
emp_Qualificatio is :PHP
teacher_id is :3
emp_separment is :MCA
Teacher subjects taught is :DS
```

### RESULT :

Program is successfully executed and output is verified.

**Experiment No : 19****Date : 13/05/2022**

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

**CLASS DIAGRAM :****SOURCE CODE :**

```
import java.util.Scanner;
class Publisher{
    public String pub_name;
    Publisher(String name){
        pub_name=name;
    }
}
class Book extends Publisher{
    public String book_name,author;
```

```

        public float price;
        Book(String pname,String aname,String bname,float price){
            super(pname);
            book_name=bname;
            author=aname;
            this.price=price;
        }
    }
    class Literature extends Book{
        Literature(String bname,String aname,String pname,float p1){
            super(pname,aname,bname,p1);
        }
        public void display1(){
            System.out.println("Publisher name: " + pub_name);
            System.out.println("Name of the book: : " + book_name);
            System.out.println("Author name: " + author);
            System.out.println("Price of book: " + price);
        }
    }
    class Fiction extends Book{
        Fiction(String bname,String aname,String pname,float p2){
            super(pname,aname,bname,p2);
        }
        public void display2(){
            System.out.println("Publisher name: " + pub_name);
            System.out.println("Name of the book: " + book_name);
            System.out.println("Author name: " + author);
            System.out.println("Price of book: " + price);
        }
    }
    class hierarcial{
    public static void main(String args[]){
        Scanner sc=new Scanner(System.in);
        Scanner s=new Scanner(System.in);
        System.out.println("Enter details of the book(literature): ");
        System.out.println("Name of book: ");
        String b1name=sc.nextLine();
        System.out.println("Name of author: ");
        String a1name=sc.nextLine();
        System.out.println("Publisher name: ");
        String p1name=sc.nextLine();
        System.out.println("Price: ");
        float p1=s.nextFloat();
        System.out.println("Enter details of the book(fiction): ");
        System.out.println("Name of book: ");
        String b2name=sc.nextLine();
    }
    }

```

```

        System.out.println("Name of author: ");
        String a2name=sc.nextLine();
        System.out.println("Publisher name: ");
        String p2name=sc.nextLine();
        System.out.println("Price: ");
        float p2=s.nextFloat();
        System.out.println(" ");
        System.out.println("Book Details : ");
        Literature obj1=new Literature(b1name,a1name,p1name,p1);
        obj1.display1();
        System.out.println(" ");
        System.out.println("Book Details : ");
        Fiction obj2=new Fiction(b2name,a2name,p2name,p2);
        obj2.display2();
    }
}

```

#### OUTPUT :

```

lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ javac Hierarcial.java
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ java Hierarcial
Enter details of the book(literature):
Name of book:
Great Expectations
Name of author:
Charles Dicken
Publisher name:
Simon & Schuster
Price:
1200
Enter details of the book(fiction):
Name of book:
Harry Potter and the Philosopher's Stone
Name of author:
J K Rowling
Publisher name:
Bloomsbury
Price:
299

Book Details :
Publisher name: Simon & Schuster
Name of the book: : Great Expectations
Author name: Charles Dicken
Price of book: 1200.0

Book Details :
Publisher name: Bloomsbury
Name of the book: Harry Potter and the Philosopher's Stone
Author name: J K Rowling
Price of book: 299.0

```

#### RESULT :

Program is successfully executed and output is verified.

**Experiment No : 20****Date : 18/05/2022**

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

ALGORITHM :

Step 1 : Define two interfaces Student and Sports. Define the function prototype of method which sets the value of the instance variables of class which implements it.

Step 2 : Define a class Result which implements the interfaces Student and Sports.

Step 3 : Define the methods specified in the interfaces Student and Sports in class Result in order to store the academic and sports details of a student. Also define method to display the academic and sports details.

Step 4 : Read the details(name,academic and sports) of a student.

Step 5 : Invoke the methods(inherit methods) of class Result to initialize the variables.

Step 6 : Display the details.

SOURCE CODE :

```
import java.util.*;
interface Student{
    void set_value1(String sname,int s,String regno);
}
interface Sports{
    void set_value2(float p);
}
class Result implements Student, Sports{
    public String name,reg_no;
    public int total;
    public float score_point;
    public void set_value1(String sname,int s,String regno){
        name=sname;
        total=s;
        reg_no=regno;
    }
    public void set_value2(float pt){
        score_point=pt;
    }
    public void display(){
        System.out.println("Register number: "+ reg_no);
        System.out.println("Name: " + name);
        System.out.println("Academic score: " + total);
        System.out.println("Score obtained in sports: " + score_point);
    }
}
```

```

}
class Interfaceprgrm{
    public static void main(String args[]){
        Scanner sc=new Scanner(System.in);
        Scanner s=new Scanner(System.in);
        int total;float pt;
        System.out.print("Enter the register number: ");
        String reg_no=sc.nextLine();
        System.out.print("Name of the student: ");
        String name=sc.nextLine();
        System.out.println("Enter the academic and sports details of a student: ");
        System.out.print("Enter the total marks obtained(max marks 1200): ");
        total=s.nextInt();
        System.out.println("Enter the overall score point obtained in sports(max
point 10): ");
        pt=s.nextFloat();
        Result obj=new Result();
        obj.set_value1(name,total,reg_no);
        obj.set_value2(pt);
        obj.display();
    }
}

```

#### OUTPUT :

```

lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ javac Interfaceprgrm.java
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ java Interfaceprgrm
Enter the register number: 112AH23
Name of the student: Jim
Enter the academic and sports details of a student:
Enter the total marks obtained(max marks 1200): 1111
Enter the overall score point obtained in sports(max point 10):
6.5
Register number: 112AH23
Name: Jim
Academic score: 1111
Score obtained in sports: 6.5

```

#### RESULT :

Program is successfully executed and output is verified.

**Experiment No : 21****Date : 18/05/2022**

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

**ALGORITHM** :

Step 1 : Create an interface to define the prototypes of functions, area() and perimeter() which are used to find the area and perimeter of circle and rectangle.

Step 2 : Define two classes, Circle having instance variable radius and Rectangle having instance variables length and breadth. Implement the interface with the classes. That is, the classes must define the prototypes declared in the interface.

[Use the formulas,

a) Circle

1.  $\text{area} = 3.14 * \text{radius} * \text{radius}$

2.  $\text{perimeter} = 2 * 3.14 * \text{radius}$

b) Rectangle

1.  $\text{area} = \text{length} * \text{breadth}$

2.  $\text{perimeter} = 2 * (\text{length} + \text{breadth})$ ]

Step 3 : Define a menu to display the options (circle and rectangle).

Step 4 : Read the choice of the user.

Step 5 : If the choice is to find the area and perimeter of circle then,

a) Read the radius and store it.

b) Invoke the methods of class Circle by instantiating an object of the class.

c) Display the calculated result.

Step 6 : If the choice is to find the area and perimeter of rectangle then,

a) Read length and breadth and store it

b) Invoke the methods of class Circle by instantiating an object of the class.

c) Display the result.

**SOURCE CODE** :

```
import java.util.*;
interface interface1{
    void area();
    void perimeter();
}
class Circle implements interface1{
    public float radius;
    Circle(float rad){
        radius=rad;
    }
    public void area(){
```

```

        System.out.println("Area of the circle is: " + (3.14*radius*radius));
    }
    public void perimeter(){
        System.out.println("Perimeter of the circle is: " + (2*3.14*radius));
    }
}
class Rectangle implements interface1{
    public float len,bdth;
    Rectangle(Float l, Float b){
        len=l;
        bdth=b;
    }
    public void area(){
        System.out.println("Area of the rectangle is: " + (len*bdth));
    }
    public void perimeter(){
        System.out.println("Perimeter of the circle is: " + (2*(len+bdth)));
    }
}
class Area_perimeter{
    public static void main(String[]args){
        int ch=1;
        while(ch == 1){
            Scanner sc=new Scanner(System.in);
            System.out.println("Choose either 1 or 2 to find the area and
            perimeter of,");
            System.out.println("\n 1.circle 2. Rectangle ");
            System.out.print("Enter your choice: ");
            int choice=sc.nextInt();
            switch(choice){
                case 1: System.out.print("Enter the radius of the circle: ");
                    float r=sc.nextFloat();
                    Circle obj1=new Circle(r);
                    obj1.area();
                    obj1.perimeter();
                    break;
                case 2: System.out.print("Enter the length of the rectangle:
                    ");
                    float l=sc.nextFloat();
                    System.out.print("Enter the breadth of the
                        rectangle: ");
                    float b=sc.nextFloat();
                    Rectangle obj2=new Rectangle(l,b);
                    obj2.area();
                    obj2.perimeter();
                    break;
            }
        }
    }
}

```



```

    }
    System.out.print("Want to continue?(1(yes) or 0(no)) ");
    ch=sc.nextInt();
}
}
}

```

### OUTPUT :

```

lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ javac Area_perimeter.java
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ java Area_perimeter
Choose either 1 or 2 to find the area and perimeter of,

    1.circle 2. Rectangle
Enter your choice: 1
Enter the radius of the circle: 5.6
Area of the circle is: 98.47039664611819
Perimeter of the circle is: 35.16799940109253
Want to continue?(1(yes) or 0(no)) 1
Choose either 1 or 2 to find the area and perimeter of,

    1.circle 2. Rectangle
Enter your choice: 2
Enter the length of the rectangle: 8
Enter the breadth of the rectangle: 6.6
Area of the rectangle is: 52.8
Perimeter of the circle is: 29.2
Want to continue?(1(yes) or 0(no)) 0

```

### RESULT :

Program is successfully executed and output is verified.

**Experiment No : 22****Date : 20/05/2022**

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

Order No.

Date :

Product Id	Name	Quantity	unit price	Total
101	A	2	25	50
102	B	1	100	100
Net. Amount				150

ALGORITHM:

Step 1 : Create an interface which contains a prototype of function method().

Step 2 : Define a class Product which implements the interface. Also,

a) define the instance variables to store the values product ID, product name, quantity, unit price and total price.

b) define the function method () inorder to display the details of products in a bill format. [Use format() of String class to format the output.].

Step 3 : Define an array of object to store the details of all the products.

Step 4 : Invoke the methods defined in the class Product to generate the bill.

SOURCE CODE :

```
import java.util.*;
import java.text.SimpleDateFormat;

interface bill{
    void method();
}

class Product implements bill{
    String p_name;
    int p_id,p_qty,order_no;
    float p_price,total_price;
    Product(){
    }
    Product(String name,int id,int qty,float price,float total){
        p_name=name;
        p_id=id;
        p_qty=qty;
        p_price=price;
        total_price=total;
    }
    void display(){
        Date date = new Date();
        SimpleDateFormat formatter = new SimpleDateFormat("dd/MM/yy");
```

```

        String str = formatter.format(date);
        Random rand = new Random();
        int upperbound = 25;
        int int_random = rand.nextInt(upperbound);
        System.out.println("Order no:"+ int_random);
        System.out.println("Date: " + str);
        System.out.println("\n Product Id  Name  Quantity  unit price  Total");
    }
    public void method(){
        System.out.format("%8d %7s %7d %8.2f %8.2f\n ", p_id, p_name,
        p_qty, p_price, total_price);
    }
}
class create_bill{
    public static void main(String args[]){
        Scanner sc=new Scanner(System.in);
        Scanner s=new Scanner(System.in);
        float net_amt=0,total;
        System.out.print("Enter the total number of items: ");
        int n=sc.nextInt();
        Product [] obj=new Product[n];
        for(int i=0;i<n;i++){
            System.out.println("Enter product" + " " +(i+1) + " " + "details:");
            System.out.print("Name: ");
            String name = s.nextLine();
            System.out.print("ID: ");
            int id= sc.nextInt();
            System.out.print("Quantity: ");
            int qty = sc.nextInt();
            System.out.print("Price (per item): ");
            float price = sc.nextFloat();
            total = price * qty;
            net_amt+=total;
            obj[i]=new Product(name,id,qty,price,total);
        }
        Product prdct=new Product();
        prdct.display();
        for(int i=0;i<n;i++)
            obj[i].method();
        String temp="Net Amount";
        System.out.format("%38s %8.2f ",temp,net_amt);
        System.out.println();
    }
}

```

### OUTPUT :

```
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ javac create_bill.java
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ java create_bill
Enter the total number of items: 2
Enter product 1 details:
Name: A
ID: 101
Quantity: 2
Price (per item): 25
Enter product 2 details:
Name: B
ID: 102
Quantity: 1
Price (per item): 100
Order no:13
Date: 08/06/22

Product Id  Name  Quantity  unit price  Total
    101      A      2    25.00    50.00
    102      B      1   100.00   100.00
                        Net Amount   150.00
```

### RESULT :

Program is successfully executed and output is verified.

**Experiment No : 23****Date : 20/05/2022**

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

ALGORITHM :

Step 1 : Define three child classess Rectangle, Circle and Square which inherits the class Shape (parent class).

Step 2 : Define a method in parent class to calculate the area of different shapes.

Step 3 : Define this method (defined in the parent class) in child classess ; the name and the parameter must be same (method overriding).

Step 4 : Read and store the required values of different shapes inorder to calculate the area.

Step 5 : Invoke the overridddden method in each class which calculate and display the area.

SOURCE CODE :

```
import java.util.*;
class Shapes{
    float a,b;
    Shapes(){
    }
    Shapes(float value){
        a=value;
    }
    Shapes(float val1,float val2){
        a=val1;
        b=val2;
    }
    double area(){
        System.out.println("Area of different shapes");
        return 0;
    }
}
class Rectangle extends Shapes{
    Rectangle(float a,float b){
        super(a,b);
    }
    double area(){
        return a*b;
    }
}
class Circle extends Shapes{
    Circle(float a){
```

```

        super(a);
    }
    double area(){
        return 3.14*a*a;
    }
}
class Square extends Shapes{
    Square(float a){
        super(a);
    }
    double area(){
        return a*a;
    }
}
class Find_area{
    public static void main(String args[]){
        Scanner sc=new Scanner(System.in);
        Shapes obj=new Shapes();
        obj.area();
        System.out.print("Enter the radius of the circle: ");
        float r=sc.nextFloat();
        Circle obj1=new Circle(r);
        System.out.println("Area of circle is " + " " + obj1.area());
        System.out.print("Enter the length of the rectangle: ");
        float l=sc.nextFloat();
        System.out.print("Enter the breadth of the rectangle: ");
        float b=sc.nextFloat();
        Rectangle obj2=new Rectangle(l,b);
        System.out.println("Area of rectangle is " + " " + obj2.area());
        System.out.print("Enter the side length of the square: ");
        float a=sc.nextFloat();
        Square obj3=new Square(a);
        System.out.println("Area of square is " + " " + obj3.area());
    }
}

```

#### OUTPUT :

```

lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ javac Find_area.java
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ java Find_area
Area of different shapes
Enter the radius of the circle: 5
Area of circle is 78.5
Enter the length of the rectangle: 12
Enter the breadth of the rectangle: 8.5
Area of rectangle is 102.0
Enter the side length of the square: 5
Area of square is 25.0

```

#### RESULT :

Program is successfully executed and output is verified.

**Experiment No : 24****Date : 20/05/2022**

AIM : Create an Abstract Class 'Shape' with an abstract method find Area to find the area of different shapes. Create subclasses Rectangle, Circle and Square from Shape. Calculate and display area of Rectangle, Circle and Square.

**ALGORITHM** :

Step 1 : Define an abstract class Shape with abstract method area(). [Use "abstract" keyword].

Step 2 : Define three sub classess Rectangle, Circle and Square of class Shape.

Step 3 : Each class should define the abstract method area() in such a way to calculate the area of different shapes.

a) area() of class Rectangle finds the area of rectangle.[area=length\*breadth]

b) area() of class Circle finds the area of circle.[area=3.14\*rad\*rad]

c) area() of class Square finds the area of square.[area=a\*a]

Step 4 : Read the parameters of the different shapes and invoke the method area() of corresponding classess.

Step 5 : Display the results.

**SOURCE CODE** :

```
import java.util.*;
abstract class Shape{
    abstract void calculate_area();
}
class Rectangle extends Shape{
    float len,bdth;
    Rectangle(float l,float b){
        len=l;bdth=b;
    }
    void calculate_area(){
        System.out.println("Area of rectangle is " + " " + (len*bdth));
    }
}
class Circle extends Shape{
    float radius;
    Circle(float r){
        radius=r;
    }
    void calculate_area(){
        System.out.println("Area of circle is " + " " + (3.14*radius*radius));
    }
}
class Square extends Shape{
```

```

        float side;
        Square(float a){
            side=a;
        }
        void calculate_area(){
            System.out.println("Area of square is " + " " + (side*side));
        }
    }
    class Abstraction{
        public static void main(String args[]){
            Scanner sc=new Scanner(System.in);
            System.out.print("Enter the length of the rectangle: ");
            float l=sc.nextFloat();
            System.out.print("Enter the breadth of the rectangle: ");
            float b=sc.nextFloat();
            Shape obj1=new Rectangle(l,b);
            obj1.calculate_area(); System.out.print("Enter the radius of the circle: ");
            float r=sc.nextFloat();
            Shape obj2=new Circle(r);
            obj2.calculate_area();
            System.out.print("Enter the side length of the square: ");
            float a=sc.nextFloat();
            Shape obj3=new Square(a);
            obj3.calculate_area();
        }
    }
}

```

#### OUTPUT :

```

lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ javac Abstraction.java
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ java Abstraction
Enter the length of the rectangle: 20
Enter the breadth of the rectangle: 15
Area of rectangle is 300.0
Enter the radius of the circle: 8
Area of circle is 200.96
Enter the side length of the square: 5
Area of square is 25.0

```

#### RESULT :

Program is successfully executed and output is verified.



## Lab Cycle 4

**Experiment No : 25**

**Date : 27/05/2022**

AIM : Write a Java program to perform the following operations in a HashSet.

- a) Append the specified element to the end of a hash set.
- b) Iterate through all elements in a Hash Set
- c) Check whether a hash set is empty or not
- d) Get the number of elements in a hash set.

### ALGORITHM :

Step 1 : Start

Step 2 : Declare a HashSet collection hashSet with Integer as element type.

Step 3 : Perform step 4,5,6 & 7 using menu.

Step 4 : Read an element and add to collection using hashSet.add().

Step 5 : for Integer i in hashSet print i.

Step 6 : Check emptiness of hashSet using hashSet.isEmpty().

Step 7 : Print length of hashSet using hashSet.size().

Step 8 : Stop

### SOURCE CODE :

```
import java.util.HashSet;
import java.util.Iterator;
import java.util.Scanner;
```

```
public class hashSet {
    public static void main(String[] args) {
        boolean flag = true;
        HashSet<Integer> hs = new HashSet<Integer>();
        Scanner inp = new Scanner(System.in);
        do{
            System.out.println("***** Menu *****");
            System.out.println("1 - Append an element");
            System.out.println("2 - Traverse");
            System.out.println("3 - Check empty");
            System.out.println("4 - Check size");
            System.out.println("0 - Exit");
            System.out.print("Enter your choice : ");
            int c = inp.nextInt();
            switch (c) {
                case 1: {
                    System.out.println("Enter element to be added : ");
                    hs.add(inp.nextInt());
                    // System.out.println(hs);
                }
            }
        } while (flag);
    }
}
```

```

        break;
    }
    case 2: {
        Iterator<Integer> iterator = hs.iterator();
        while(iterator.hasNext())
            System.out.println(iterator.next());
        break;
    }

    case 3: {
        if(hs.isEmpty())
            System.out.println("Hash set is empty");
        else
            System.out.println("Hash set is not
            empty");
        break;}
    case 4: {
        System.out.println("Length of hash set is
        "+hs.size());
        break;}
    case 0: {
        System.out.println("BYE...");
        flag = false;
        break;
    }
    }}
    }while(flag);
    inp.close();
}
}

```

OUTPUT :

```

lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)
***** Menu *****
1 - Append an element
2 - Traverse
3 - Check empty
4 - Check size
0 - Exit
Enter your choice : 1
Enter element to be added :
10
***** Menu *****
1 - Append an element
2 - Traverse
3 - Check empty
4 - Check size
0 - Exit
Enter your choice : 1
Enter element to be added :
20

```

```

***** Menu *****
1 - Append an element
2 - Traverse
3 - Check empty
4 - Check size
0 - Exit
Enter your choice : 1
Enter element to be added :
30
***** Menu *****
1 - Append an element
2 - Traverse
3 - Check empty
4 - Check size
0 - Exit
Enter your choice : 1
Enter element to be added :
40
***** Menu *****
1 - Append an element
2 - Traverse
3 - Check empty
4 - Check size
0 - Exit
Enter your choice : 2
20
40
10
30
***** Menu *****
1 - Append an element
2 - Traverse
3 - Check empty
4 - Check size
0 - Exit
Enter your choice : 3
Hash set is not empty
***** Menu *****
1 - Append an element
2 - Traverse
3 - Check empty
4 - Check size
0 - Exit
Enter your choice : 4
Length of hash set is 4
***** Menu *****
1 - Append an element
2 - Traverse
3 - Check empty
4 - Check size
0 - Exit
Enter your choice : 0
BYE...

```

#### RESULT :

Program is successfully executed and output is verified.

**Experiment No : 26****Date : 1/06/2022**

AIM : Write a Java program to create a reverse order view of the elements contained in a given tree set.

ALGORITHM :

Step 1 : Start

Step 2 : Declare a TreeSet&lt;Integer&gt; collection object treeSet

Step 3 : Add some integer to treeSet using add() function.

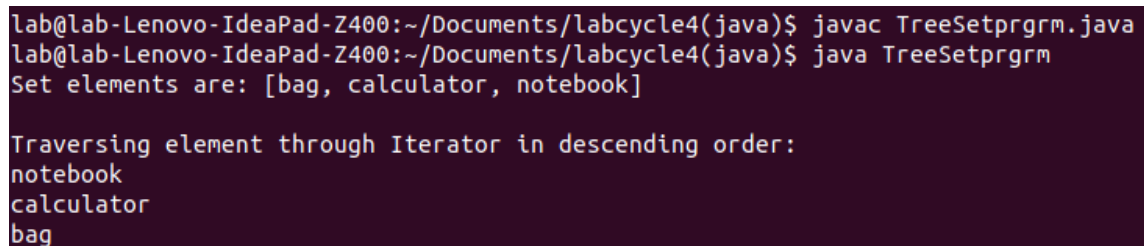
Step 4 : for int i in treeSet.descendingSet() perform step5

Step 5 : Print value of i

Step 6 : Stop

SOURCE CODE :

```
import java.util.*;
class TreeSetprgrm{
    public static void main(String args[]){
        TreeSet <String>set=new TreeSet<>();
        set.add("notebook");
        set.add("bag");
        set.add("calculator");
        set.add("calculator");//treeset contains only unique elements
        System.out.println("Set elements are: "+ set);
        System.out.println();
        System.out.println("Traversing element through Iterator in
        descending order: ");
        Iterator i=set.descendingIterator();
        while(i.hasNext())
            System.out.println(i.next());
    }
}
```

OUTPUT :

```
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ javac TreeSetprgrm.java
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ java TreeSetprgrm
Set elements are: [bag, calculator, notebook]

Traversing element through Iterator in descending order:
notebook
calculator
bag
```

RESULT :

Program is successfully executed and output is verified.

**Experiment No : 27****Date : 1/06/2022**

AIM : Program to demonstrate the creation of a Set object using the LinkedHashSet class.

ALGORITHM :

Step 1 : Start

Step 2 : Declare a LinkedHashSet linkedHashSet with String as object type

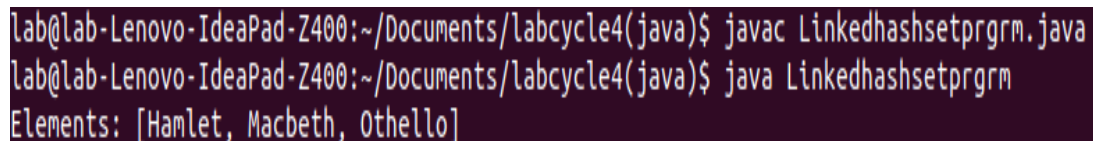
Step 3 : Add Strings to linkedHashSet using add() fn

Step 4 : Display linkedHashSet

Step 5 : Stop

SOURCE CODE :

```
import java.util.*;
class Linkedhashsetprgrm{
    public static void main(String args[]){
        LinkedHashSet<String> set=new LinkedHashSet<>();
        set.add("Hamlet");
        set.add("Macbeth");
        set.add("Othello");
        set.add("Othello");//linkedhashset contains only unique elements
        System.out.println("Elements: " + set);
    }
}
```

OUTPUT :

```
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ javac Linkedhashsetprgrm.java
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ java Linkedhashsetprgrm
Elements: [Hamlet, Macbeth, Othello]
```

RESULT :

Program is successfully executed and output is verified.

**Experiment No : 28****Date : 1/06/2022**

AIM : Maintain a list of Strings using ArrayList from the collection framework, perform built-in operations.

ALGORITHM :

Step 1 : Start

Step 2 : Declare a ArrayList names with with String as object type

Step 3 : Add some String values to "names" using add() function

Step 4 : Display a menu to perform the following steps 5,6,7,8 & 9

Step 5 : Read index value and name from user, then add to names using  
add(index,name)

Step 6 : Read name from user and print its position using indexOf(name) fn

Step 7 : Print the size of ArrayList "names" using size() function

Step 8 : Check emptiness of names using isEmpty()

Step 9 : Remove name in names by reading index value from user, using remove(index)

Step 10 : Stop

SOURCE CODE :

```
import java.util.*;
class Arraylistprgrm{
    public static void main(String args[]){
        try{
            ArrayList<String> names=new ArrayList<>();
            Scanner sc=new Scanner(System.in);
            Scanner s=new Scanner(System.in);
            int p; boolean result; String name;
            names.add("J K ROWLING");
            names.add("LEO TOLSTOY");
            names.add("WILLIAM SHAKESPEARE");
            System.out.println("Names of famous authors: " + names + "\n");
            System.out.println("*****Enter the strings in
            uppercase*****");
            int i=1;
            while(i==1){
                System.out.println(" 1.Insertion at specified position\n 2.Find index
                position of an element\n 3.Get the size of the array list\n 4.Check
                whether list is empty or not\n 5.Remove an element by its index");
                System.out.println("Enter your choice: ");
                int ch=s.nextInt();
                Switch(ch){
                    case 1: System.out.println("Enter the name of your favourite
                    author:");
                        name=sc.nextLine();
```

```

        System.out.println("Enter the index position:");
        p=s.nextInt();
        names.add(p-1,name);
        System.out.println("Array list after insertion: " + names);
        break;
    case 2: System.out.println("Enter the name of an author:");
        name=sc.nextLine();
        System.out.println("Index position of the given element
        is : " + " " + (names.indexOf(name)+1));
        break;
    case 3: System.out.println("Size of the array list is: " +
        names.size());
        break;
    case 4: result=names.isEmpty();
        System.out.println("List is empty or not? " + result);
        break;
    case 5: System.out.println("Enter the index position:");
        p=s.nextInt();
        names.remove(p-1);
        System.out.println("Array list after deletion: " + names);}
    System.out.println("Do you want to continue?(1 or 0)");
    i=s.nextInt();}
}
catch(Exception e)
    System.out.println("something went wrong" + e);}
}

```

### OUTPUT :

```

lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ javac ArrayListprgrm.java
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ java ArrayListprgrm
Names of famous authors: [J K ROWLING, LEO TOLSTOY, WILLIAM SHAKESPEARE]

*****Enter the strings in uppercase*****
1.Insertion at specified position
2.Find index position of an element
3.Get the size of the array list
4.Check whether list is empty or not
5.Remove an element by its index
Enter your choice:
1
Enter the name of your favourite author:
MAYA ANGELOU
Enter the index position:
3
Array list after insertion: [J K ROWLING, LEO TOLSTOY, MAYA ANGELOU, WILLIAM SHAKESPEARE]
Do you want to continue?(1 or 0)
1
1.Insertion at specified position
2.Find index position of an element
3.Get the size of the array list
4.Check whether list is empty or not
5.Remove an element by its index
Enter your choice:
2
Enter the name of an author:
LEO TOLSTOY
Index position of the given element is : 2

```

```

Do you want to continue?(1 or 0)
1
1.Insertion at specified position
2.Find index position of an element
3.Get the size of the array list
4.Check whether list is empty or not
5.Remove an element by its index
Enter your choice:
3
Size of the array list is: 4
Do you want to continue?(1 or 0)
1
1.Insertion at specified position
2.Find index position of an element
3.Get the size of the array list
4.Check whether list is empty or not
5.Remove an element by its index
Enter your choice:
4
List is empty or not? false
Do you want to continue?(1 or 0)
1
1.Insertion at specified position
2.Find index position of an element
3.Get the size of the array list
4.Check whether list is empty or not
5.Remove an element by its index
Enter your choice:
5
Enter the index position:
3
Array list after deletion: [J K ROWLING, LEO TOLSTOY, WILLIAM SHAKESPEARE]
Do you want to continue?(1 or 0)
0

```

### RESULT :

Program is successfully executed and output is verified.



**Experiment No : 29****Date : 3/06/2022**

AIM : Program to remove all the elements from a linked list.

ALGORITHM :

Step 1 : Start

Step 2 : Define a Class linkedlist prgrm

Step 3 : Create an ArrayList of string type called "fruits".

Step 4 : Add elements into ArrayList using add() method.

Step 5 : Declare a linkedlist ("fruit-list) object with parameter 'fruits'

Step 6 : Display the LinkedList elements

Step 7 : Delete all elements in linkedlist using clear() method.

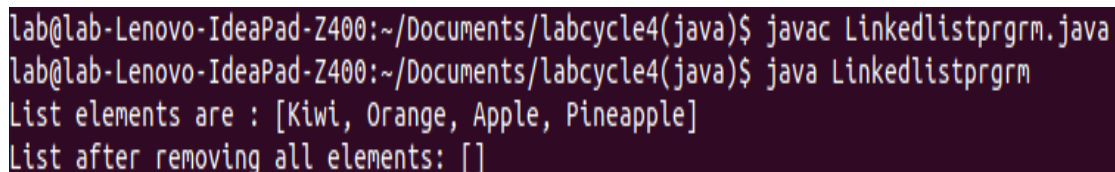
Step 8 : Display the linked list after removing all elements

Step 9 : Stop

SOURCE CODE :

```
import java.util.*;
class LinkedListprgrm{
    public static void main(String args[]){
        ArrayList<String> fruits=new ArrayList<>();
        fruits.add("Kiwi");
        fruits.add("Orange");
        fruits.add("Apple");
        fruits.add("Pineapple");
        LinkedList<String> fruits_list=new LinkedList<>(fruits);
        System.out.println("List elements are : " + fruits_list);
        fruits_list.clear();
        System.out.println("List after removing all elements: " + fruits_list);
    }
}
```

OUTPUT :



```
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ javac LinkedListprgrm.java
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ java LinkedListprgrm
List elements are : [Kiwi, Orange, Apple, Pineapple]
List after removing all elements: []
```

RESULT :

Program is successfully executed and output is verified.

## **Experiment No : 30**

**Date :**

AIM : Program to remove an object from the Stack when the position is passed as parameter.

### ALGORITHM :

Step 1 : Start

Step 2 : Define a Create class Stackprym & stack object "numbers"

Step 3 : Add elements into stack object using push().

Step 4 : Print the shot elements.

Step 5 : Accept position of object to be deleted from user

Step 6 : use remove() method of stack to remove element of (position -1).

Step 7 : Display the removed element

Step 8 : Print the list of elements after deletion

Step 9 : Stop

### SOURCE CODE :

```
import java.util.*;
class Stackprgrm{
    public static void main(String args[]){
        Scanner n=new Scanner(System.in);
        Stack<Integer> numbers=new Stack<>();
        numbers.push(12);
        numbers.push(14);
        numbers.push(48);
        numbers.push(90);
        System.out.println("List elements are: " + numbers);
        System.out.println("Enter the position of the object to be deleted: ");
        int pos=n.nextInt();
        int top=numbers.remove(pos-1);
        System.out.println("Removed object is: " + top);
        System.out.println("List after deletion : " + numbers);
    }
}
```

### OUTPUT :

```
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ javac Stackprgrm.java
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ java Stackprgrm
List elements are: [12, 14, 48, 90]
Enter the position of the object to be deleted:
3
Removed object is: 48
List after deletion : [12, 14, 90]
```

### RESULT :

Program is successfully executed and output is verified.

**Experiment No : 31**

**Date : 3/06/2022**

AIM : Program to demonstrate the creation of queue objects using the PriorityQueue class.

ALGORITHM :

Step 1 : Start

Step 2 : Define a class PriorityQueue prgrm

Step 3 : Create an object 'art' of Priority Queue of string type

Step 4 : Add elements into priority queue using add() method.

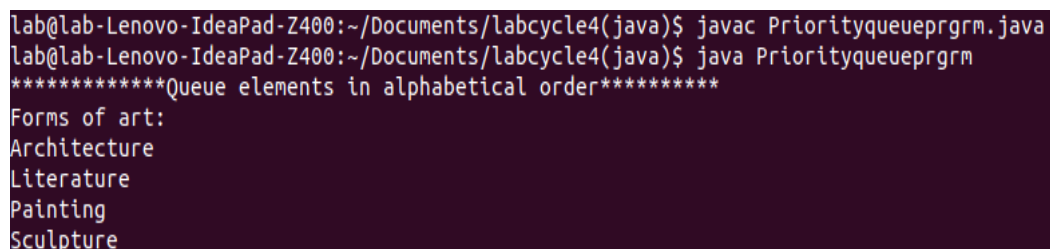
Step 5 : Define the iterator to i for priority queue to iterate through the priority queue & display its elements.

Step 6 : Stop

SOURCE CODE :

```
import java.util.*;
class PriorityQueueprgrm{
    public static void main(String args[]){
        PriorityQueue<String> art = new PriorityQueue<>();
        System.out.println("*****Queue elements in alphabetical
order*****");
        art.add("Painting");
        art.add("Sculpture");
        art.add("Literature");
        art.add("Architecture");
        Iterator i = art.iterator();
        System.out.println("Forms of art: ");
        while (i.hasNext())
            System.out.print(i.next() + " \n");
    }
}
```

OUTPUT :



```
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ javac PriorityQueueprgrm.java
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ java PriorityQueueprgrm
*****Queue elements in alphabetical order*****
Forms of art:
Architecture
Literature
Painting
Sculpture
```

RESULT :

Program is successfully executed and output is verified.

**Experiment No : 32****Date : 3/06/2022**

AIM : Program to demonstrate the addition and deletion of elements in deque.

ALGORITHM :

Step 1 : Start

Step 2 : Declare a object deque of Deque type with String type element

Step 3 : Assign an instance of LinkedList collection with String type element to deque

Step 4 : Perform the following steps 5,6,7,8,9,10,11 & 12 using menu

Step 5 : Read an String from user and add it to deque using add() fn

Step 6 : Read an String from user and add at begining of deque using addFirst() fn

Step 7 : Read an String from user and add at end of deque using addLast() fn

Step 8 : Read an String from user and perform push operation using push() fn

Step 9 : Display elements of deque using iterator

Step 10 : Display elements of deque after performing pop operation on deque using pop() fn

Step 11 : Display elements of deque after removing first element of deque using removeFirst() fn

Step 12 : Display elements of deque after removing last element of deque using removeLast() fn

Step 13 : Stop

SOURCE CODE :

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

```
public class Dequeprgrm {
```

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

```
        Deque<String> deque = new LinkedList<String>();
```

```
        Scanner s=new Scanner(System.in);
```

```
        Scanner sc=new Scanner(System.in);
```

```
        int i=1;
```

```
        while(i==1){
```

```
            System.out.println(" 1.Insertion \n 2.Deletion");
```

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

```
            int ch=s.nextInt();
```

```
            switch(ch){
```

```
                case 1: System.out.println("Add an element(string) to deque : ");
```

```
                    String name=sc.nextLine();
```

```
                    deque.add(name);
```

```
                    System.out.println("Add an element(string) to the  
beginning of deque : ");
```

```
                    String first=sc.nextLine();
```

```
                    deque.addFirst(first);
```

```
                    System.out.println("Add an element(string) to the end of  
deque : ");
```

```
                    String end=sc.nextLine();
```

```

        deque.addLast(end);
        System.out.println("push an element(string) to th deque
        :");
        String p=sc.nextLine();
        deque.push(p);
        System.out.println("\nDeque contents :");
        Iterator iterator = deque.iterator();
        while (iterator.hasNext())
            System.out.println(" " + iterator.next());
        break;
    case 2: System.out.println("\nPop:" + deque.pop());
        System.out.println("\nDeque after pop:" + deque);
        deque.removeFirst();
        deque.removeLast();
        System.out.println("\nDeque, after removing " + "first and
        last elements: " + deque);
        break;}
    System.out.print("Do you want to continue?(1 or 0)");
    i=s.nextInt();}}

```

#### OUTPUT :

```

lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ javac Dequeprgrm.java
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ java Dequeprgrm
1.Insertion
2.Deletion
Enter your choice:
1
Add an element(string) to deque :
Python
Add an element(string) to the beginning of deque :
Java
Add an element(string) to the end of deque :
SQL
push an element(string) to the deque :
Perl
Deque contents :
Perl
Java
Python
SQL
Do you want to continue?(1 or 0)1
1.Insertion
2.Deletion
Enter your choice:
2
Pop:Perl
Deque after pop:[Java, Python, SQL]
Deque, after removing first and last elements: [Python]
Do you want to continue?(1 or 0)0

```

#### RESULT :

Program is successfully executed and output is verified.

**Experiment No : 33****Date : 2/06/2022**

AIM : Program to demonstrate the working of Map interface by adding, changing and removing elements.

ALGORITHM :

Step 1 : Start

Step 2 : Declare a object hashMap of HashMap type with key-value type as Integer-String

Step 3 : Add some key-value pair to hashMap using put(key,value) fn

Step 4 : Display the elements of hashMap to show initial state.

Step 5 : Perform the following steps using menu

Step 6 : Read key and value from user and add them to hashMap using put(key,value) fn

Step 7 : Read key for element to be deleted and remove it from hashMap using remove(key) fn

Step 8 : Perform the following to update hashMap -

i) Read key of element to be replaced and new value

ii) add them to hashMap using replace(key,value) fn

Step 9 : Stop

SOURCE CODE :

```
import java.util.*;
class Mapprgrm{
    public static void main(String args[]){
        HashMap<Integer,String> hm=new HashMap<Integer,String>();
        Scanner s=new Scanner(System.in);
        Scanner sc=new Scanner(System.in);
        int k; String val;
        hm.put(101,"Vijay");
        hm.put(102,"Rahul");
        System.out.println("Initial list of elements:");
        for(Map.Entry m:hm.entrySet())
            System.out.println(m.getKey()+" "+m.getValue());
        System.out.println(" ");
        int i=1;
        while(i==1){
            System.out.println(" 1.Insertion \n 2.Deletion\n 3.Replace");
            System.out.println("Enter your choice: " );
            int ch=s.nextInt();
            switch(ch){
                case 1: System.out.println("Enter a data as key(integer)-
                    value(string) pair into the given list:");
                    System.out.print("Key: ");
                    k=s.nextInt();
                    System.out.print("Value: ");
```

```

        val=sc.nextLine();
        hm.put(k,val);
        System.out.println("Updated list of elements: " + hm);
        break;
    case 2: System.out.print("Enter the key to be deleted: ");
        k=s.nextInt();
        hm.remove(k);
        System.out.println("Updated list of elements: " + hm);
        break;
    case 3: System.out.print("Enter the key of the key-value pair to be
        replaced: ");
        k=s.nextInt();
        System.out.print("Enter the new value: ");
        val=sc.nextLine();
        hm.replace(k,val);
        System.out.println("Updated list of elements: " + hm);
        break;}
    System.out.print("Do you want to continue?(1 or 0)");
    i=s.nextInt();}}

```

#### OUTPUT :

```

lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ javac Mapprgrm.java
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ java Mapprgrm
Initial list of elements:
101 Vijay
102 Rahul

1.Insertion
2.Deletion
3.Replace
Enter your choice:
1
Enter a data as key(integer)- value(string) pair into the given list:
Key: 103
Value: Sheela
Updated list of elements: {101=Vijay, 102=Rahul, 103=Sheela}
Do you want to continue?(1 or 0)1
1.Insertion
2.Deletion
3.Replace
Enter your choice:
2
Enter the key to be deleted: 102
Updated list of elements: {101=Vijay, 103=Sheela}
Do you want to continue?(1 or 0)1
1.Insertion
2.Deletion
3.Replace
Enter your choice:
3
Enter the key of the key-value pair to be replaced: 101
Enter the new value: Anju
Updated list of elements: {101=Anju, 103=Sheela}
Do you want to continue?(1 or 0)0

```

#### RESULT :

Program is successfully executed and output is verified.

**Experiment No : 34****Date : 2/06/2022**

AIM : Program to Convert HashMap to TreeMap.

ALGORITHM :

Step 1 : Start

Step 2 : Declare a object hashMap of type HashMap collection with key-value type as String.

Step 3 : Add some key-value pairs to hashMap using put(key,value).

Step 4 : Declare a object treeMap of type TreeMap collection with key-value type as String.

Step 5 : Convert hashMap to TreeMap by add all elements of hashMap to treeMap using putAll()fn.

Step 6 : Display the TreeMap.

Step 7 : Stop

SOURCE CODE :

```
import java.util.*;
public class Conversion{
    public static void main(String args[]) {
        Map<String, String> map = new HashMap<>();
        Scanner s=new Scanner(System.in);
        Scanner sc=new Scanner(System.in);
        int k; String val;
        map.put("Red", "#FF0000");
        map.put("Green", "#00FF00");
        map.put("Blue", "#0000FF");
        System.out.println("\n List contains:" + map);
        Map<String, String> treeMap = new TreeMap<>();
        treeMap.putAll(map);
        System.out.println("\n After conversion (HashMap to TreeMap), the list
        will be:\n " + treeMap);
    }
}
```

OUTPUT :

```
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ javac Conversion.java
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ java Conversion

List contains: {Red=#FF0000, Blue=#0000FF, Green=#00FF00}

After conversion (HashMap to TreeMap), the list will be:
{Blue=#0000FF, Green=#00FF00, Red=#FF0000}
```

RESULT :

Program is successfully executed and output is verified.