

## **LOYOLA INSTITUTE OF TECHNOLOGY**

**[Affiliated to Anna University]**

**Palanchur, Chennai – 600 123**



**DEPARTMENT OF INFORMATION TECHNOLOGY**

**CS8383- OBJECT ORIENTED PROGRAMMING LABORATORY**

**III SEMESTER**

**NOV/DEC-2020**

**LOYOLA INSTITUTE OF TECHNOLOGY**  
**Palanchur, Chennai – 600 123**



**UNIVERSITY REGISTER NO:**

210919205005

**CERTIFICATE**

Certified to be bonafide record of work done by ANCY.A of III Semester B.TECH(INFORMATION TECHNOLOGY) in the **CS8383 – OBJECT ORIENTED PROGRAMMING LABORATORY** during the Academic Year 2020 - 2021.

**STAFF-IN-CHARGE**

**HEAD OF THE DEPARTMENT**

Submitted for the University Practical Examination held on \_\_\_\_\_

**INTERNAL EXAMINER**

**EXTERNAL EXAMINER**

## INDEX

EX.NO	DATE	NAME OF THE EXPERIMENT	PAGE NO	SIGNATURE
1	12-8-20 13-8-20 14-8-20 19-8-20	GENERATION OF ELECTRICITY BILL	4	
2	20-8-20 21-8-20	IMPLEMENTATION OF PACKAGES	10	
3	26-8-20 27-8-20	IMPLEMENTATION OF INHERITANCE	20	
4	28-8-20 2-9-20	IMPLEMENTATION OF STACK ADT	29	
5	3.9.20 4.9.20	IMPLEMENTATION OF ARRAY LIST	37	
6	9.9.20 10.9.20	IMPLEMENTATION OF ABSTRACT CLASS	43	
7	11.9.20 16.9.20	IMPLEMENTATION OF EXCEPTION HANDLING	49	
8	17.9.20 18.9.10	DISPLAYING INFORMATION ABOUT A FILE	54	
9	23.9.20 24.9.20	IMPLEMENTATION OF MULTITHREADING	58	
10	25.9.20 30.9.20 1.10.20 7.10.20	IMPLEMENTATION OF GENERIC METHODS	63	
11	8.10.20 9.10.20 14.10.20 15.10.20	IMPLEMENTATION OF SCIENTIFIC CALCULATOR	67	

Ex>No: 1  
Date: 12/8/2020; 13/8/2020;  
11/8/2020; 19/8/2020  
AIM:

## Generation of Electricity Bill

TO Write a Java application to generate Electricity bill  
Algorithm;

1. Create a ~~file~~ class named Electricity Bill and a Main class.
2. Declare the necessary fields for generating a electricity Bill in both the classes
3. Create a constructor for the Electricity Bill class with necessary arguments.
4. Define a function for generating bill for either Commercial bill or domestic bill on the consumer's tariff.
5. For domestic consumer, the following conditions are verified.
6. For unit less than 100, Rs 1 is net amount.
7. For unit less than 100 and less than 200, add Rs. 2.50 to the net amount.
8. For units greater than 200 and less than 500, Rs. 4 is the net amount.
9. For units greater than 500, add Rs. 6 to the net amount.
10. For commercial user, the following conditions are verified.
11. For units less than 100 add Rs. 2 to the net amount.
12. For units greater than 200 and less than 200, Rs. 4.50 is the net amount.
13. For units greater than 200 and less than 500, add Rs. 6 to net amount.
14. For units greater than 500, add Rs. 7 to the net amount.
15. In the main class, read necessary input from the user.
16. Call the Electricity Bill class constructor and pass the input values as arguments.
17. Generate the electricity bill and print the details.

```
program
import java.util.*;
class ElectricityBill
{
    private String consumer_name;
    private int consumer_number;
    private int perviousMonthReading;
    private int currentMonthReading;
    private boolean isCommercial=false;
    private float netAmount=0f;
    public ElectricityBill(int cnum, String cname, int prev, int current,
boolean isCom)
    {
        this.consumer_name = cname;
        this.consumer_number = cnum;
        this.perviousMonthReading = prev;
        this.currentMonthReading = current;
        this.isCommercial = isCom;
    }

    public void generateBill()
    {
        if(isCommercial == false)
        {
            for(int i=1; i<=500currentMonthReading; i++)
            {
                if(i<=100)
                {
                    netAmount= netAmount+1.0f;
                }
                else if(i>=101 && i<=200)
                {
                    netAmount += 2.50f;
                }
            }
        }
    }
}
```

```

        }
        else if(i>=201 && i<=500)
        {
            netAmount += 4.0f;
        }
        else if(i>=501)
        {
            netAmount += 5.0f;
        }
    }

}

else
{
    for(int i=1; i<=currentMonthReading; i++)
    {
        if(i<=100)
        {
            netAmount+= 2.0f;
        }
        else if(i>=101 && i<=200)
        {
            netAmount += 4.50f;
        }
        else if(i>=201 && i<=500)
        {
            netAmount += 6.0f;
        }
        else if(i>=501)
        {
            netAmount += 7.0f;
        }
    }
}

System.out.println("\n\nName: "+consumer_name);

```

```
        System.out.println("Number: "+consumer_number);
        System.out.println("Previous Month Reading:
"+perviousMonthReading);
        System.out.println("Current Month Readin:
"+currentMonthReading);
        if(isCommercial)
        {
            System.out.println("Connection Type: Commercial");
        }
        else
        {
            System.out.println("Connection Type: Domestic");
        }
        System.out.println("Bill Amount(INR): "+netAmount);
    }

}

class Main
{

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

    public static void main(String[] args)
    {
        String consumer_name;
        int consumer_number;
        int perviousMonthReading;
        int currentMonthReading;
        boolean isCommercial=false;

        System.out.println("Enter the Consumer First Name:");
        consumer_name = sc.next();
        consumer_name += " ";
    }
}
```

```
System.out.println("Enter the Consumer Last Name");
consumer_name += sc.nextLine();

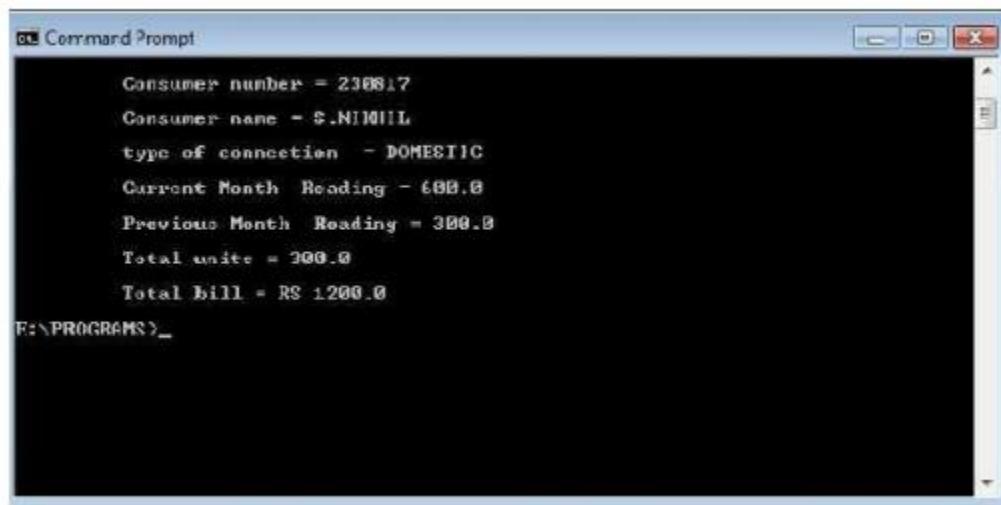
System.out.println("Enter the Consumer Number:");
consumer_number = sc.nextInt();

System.out.println("Enter the prev month reading:");
previousMonthReading = sc.nextInt();

System.out.println("Enter the current month reading:");
currentMonthReading = sc.nextInt();

while(true)
{
    System.out.println("Is it Commercial Connection?: Y/N");
    String type = sc.nextLine();
    if(type.equals("n") || type.equals("N"))
    {
        isCommercial = false;
        break;
    }
    else if(type.equals("y") || type.equals("Y"))
    {
        isCommercial = true;
        break;
    }
    else
    {
        System.out.println("Invalid choice, Enter a valid
choice.");
        continue;
    }
}
```

```
ElectricityBill eb = new ElectricityBill(consumer_number,  
consumer_name, perviousMonthReading, currentMonthReading,  
isCommercial);  
  
eb.generateBill();  
}  
}
```



Result:-

Thus, the Java application to generate the electricity bill was written and executed.

Ex: No: 02  
Date: 20/8/20; 21/8/20

## Implementation of packages

Aim:

To write a Java application for conversion of units and currencies using packages.

Algorithm:-

1. Create 3 classes for currency conversion, distance conversion and time conversion in single package
2. In currency converter class, define a constructor that takes a number as its argument
3. Declare variables with values of 1 dollar, 1 euro and 1 yen equivalents Indian rupees.
4. Define a method that performs the conversion based on formulae as follows.
5. The user is prompted with options for currency conversion.
6. For dollar to rupees, Multiply the number by with 1 dollar equivalent rupees.
7. For Euro to rupees, Multiply the number by 1 euro equivalent rupees.
8. For yen to rupees, Multiply the number with 1 yen equivalent rupees.
9. For rupees to dollar, divide the number with 1 dollar equivalent rupees.
10. For rupees to euro, divide the number with 1 dollar equivalent rupees.
11. For rupees to yen, divide the number with 1 dollar equivalent rupees.
12. In distance converter class, define a constructor that takes a number as its argument
13. Declare variables with values of 1 mile and 1 meter equivalent kilometer.
14. Define a method that performs the conversion based on formulae as follows.

15. The user is prompted with options for distance conversion.
16. For meter to kilometer, multiply the number with 1 meter equivalent kilometer
17. For kilometer to meter, divide the number with 1 meter equivalent kilometer
18. For mile to kilometer, Multiply the number with 1 meter equivalent kilometer
19. For kilometer to mile, divide the number with 1 mile equivalent kilometer.
20. In timeconverter class, define a constructor that takes a number as its argument
21. For hour to minutes, Multiply the number with 60.0
22. For Minutes to hour, divide the number with 60.0
23. In the Main Method, the user is asked to enter a number and to choose a conversion is performed and that
24. The chosen conversion is performed and the result is displayed to the user.
25. Step (23 and 24) are performed until the user chooses to exit

**PROGRAM :**

```
//converter Package
//Currency Converter Class
package converter;
import java.util.Scanner;

public class CurrencyConverter
{
    private float oldValue, newValue = 0;

    Scanner sc = new Scanner(System.in);

    public CurrencyConverter(float value)
    {
        this.oldValue = value;
    }

    public void convertCurrency()
    {
        int choice;

        float perDollar = 68.79f;
        float perEuro = 80.82f;
        float perYen = 0.62f;

        while(true)
        {
            System.out.println("\nChoose:\n1.Dollar to Rupees\n2.Euro to
Rupees\n3.Yen to Rupees\n 4.Rupees to Dollar\n5.Rupees to Euro\n6.Rupees
to Yen\n7.Go Back");
            choice = sc.nextInt();
            System.out.println(); // one line space
```

```
if(choice == 7)
{
    break;
}

switch(choice)
{
case 1:
    newValue = oldValue * perDollar;
    System.out.println(oldValue + " Dollar = "+ newValue +" Rupee");
    break;
case 2:
    newValue = oldValue * perEuro;
    System.out.println(oldValue + " Euro= "+ newValue +" Rupee");
    break;
case 3:
    newValue = oldValue * perYen;
    System.out.println(oldValue + " Yen = "+ newValue +" Rupee");
    break;
case 4:
    newValue = oldValue / perDollar;
    System.out.println(oldValue + " Rupee = "+ newValue +" Dollar");
    break;
case 5:
    newValue = oldValue / perEuro;
    System.out.println(oldValue + " Rupee = "+ newValue +" Euro");
    break;
case 6:
    newValue = oldValue / perYen;
    System.out.println(oldValue + " Rupee = "+ newValue +" Yen");
    break;

default:
    System.out.println("Invalid Choice. Select a valid option.");
}
```

```
        break;
    }
}
}
//Distance converter class
package converter;
import java.util.Scanner;

public class DistanceConverter {

    private float oldValue, newValue = 0;

    Scanner sc = new Scanner(System.in);

    public DistanceConverter(float value){
        this.oldValue = value;
    }

    public void distanceConverter(){
        int choice;

        float perMile = 0.621371f;
        float perMeter = 0.001f;

        while(true)
        {
            System.out.println("\nChoose:\n1.Meter to Kilometer\n2.Kilometer to
Meter\n3.Miles to Kilometer\n"
+ "4.Kilometer to Miles\n5.Go Back");

            choice = sc.nextInt();

```

```
if(choice == 5)
{
    break;
}

switch(choice){
    case 1:
        newValue = oldValue * perMeter;
        System.out.println(oldValue + " Meter = "+ newValue +
Kilometer");
        break;
    case 2:
        newValue = oldValue / perMeter;
        System.out.println(oldValue + " Kilometer = "+ newValue + "
Meter");
        break;
    case 3:
        newValue = oldValue / perMile;
        System.out.println(oldValue + " Miles = "+ newValue + " Kilometer");
        break;
    case 4:
        newValue = oldValue * perMile;
        System.out.println(oldValue + " Kilometer = "+ newValue + "
Miles");
        break;
    case 5:
        break;
    default:
        System.out.println("Invalid Choice. Select a valid option.");
        break;
}
}
```

```
//Time convertor class
package converter;
import java.util.Scanner;

public class TimeConverter {
    private float oldValue, newValue = 0;

    Scanner sc = new Scanner(System.in);

    public TimeConverter(float value){
        this.oldValue = value;
    }

    public void timeConverter(){
        int choice;

        while(true){
            System.out.println("\nChoose:\n1.Hour to Minute\n2.Minute to
Hour\n3.Go Back\n");
            choice = sc.nextInt();
            if(choice == 3){
                break;
            }

            switch(choice){
                case 1:
                    newValue = oldValue * 60.0f;
                    System.out.println(oldValue + " Hour = "+newValue+"Minutes");
                    break;
                case 2:
                    newValue = oldValue / 60.0f;
                    System.out.println(oldValue + " Minutes = "+newValue+" Hours");
                    break;
                case 3:
```

```
        break;
    default:
        System.out.println("Invalid Choice. Select a valid option.");
        break;
    }

}

}

// com.company.tonypsr Package
//Main package
package com.company.tonypsr;

import converter.*;
import java.util.Scanner;

public class Main {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        float num; int choice;

        System.out.println("\t\tConverter");

        while(true){
            System.out.println("\n1.Currency\n2.Distance\n3.Time\n4.exit");
            choice = sc.nextInt();
            if(choice == 4){
                System.exit(1);
            }

            System.out.println("Enter a number:");
            num = sc.nextInt();
        }
    }
}
```

```
switch (choice){  
    case 1:  
        CurrencyConverter currency = new CurrencyConverter(num);  
        currency.convertCurrency();  
        break;  
    case 2:  
        DistanceConverter distance = new DistanceConverter(num);  
        distance.distanceConverter();  
        break;  
    case 3:  
        TimeConverter time = new TimeConverter(num);  
        time.timeConverter();  
        break;  
    default:  
        System.out.println("Invalid choice. Select a valid option");  
    }  
}  
}  
}
```

OUTPUT

```
1.Currency
2.Distance
3.Time
4.exit
Enter the choice
1
Enter a number:
74

Choose:
1.DollartoRupees
2.EurotoRupees
3.YentoRupees
4.RupeestоАollar
5.RupeestоЕuro
6.RupeestоЎен
7.GoBack
1

74.0Dollar=5090.46Rupee

1.Currency
2.Distance
3.Time
4.exit
Enter the choice
2
Enter a number:
1000

Choose:
1.MetertoKilometer
2.Kilometerto Meter
3.MilestoKilometer
4.KilometertoMiles
5.GoBack
2
1000.0Kilometer=999999.94Meter

1.Currency
2.Distance
3.Time
4.exit
Enter the choice
3
Enter a number:
127

Choose:
1.HourtoMinute
2.MinutetoHour
3.Go Back

1
127.0Hour=7620.0Minutes
```

Result:

Thus, the Java application for conversion of units and currencies using Packages has been written, executed and output is verified.

Ex: NO: 3

Date: 26/8/20; 07/8/20

Aim:-

To develop a java application for generation of salary slip using inheritance

Algorithm:-

1. Create an Employee class with Emp name, Emp Id, address, mail id, mobile number and basic pay.
2. Create a parameterised constructor that takes basicpay as parameter.
3. Define a getDetails method to get details from the user.
4. Define a generate method to calculate gross, net salary and display the salary slip.
5. Create a class of Associate Professor, professor, programmer and Assistant professor that extends the Employee class.
6. The constructor of each of the sub class gets basic pay as parameter.
7. The values of DA, HRA, PF and staff club fund are calculated as follows.
8. DA is 9% of BP, HRA is 10% of BP, PF is 12% of BP and Staff club fund is 0.1% BP.
9. Define Method to call the generate method in the Super class with necessary parameters.
10. Create a main class with the main method
11. In the main method the user is provided with choice.
12. Call the getDetails and generate pay slip methods from object of class that agrees with user's selection.
13. Get the data from the user
14. Display the results.

## Implementation of Inheritance

**PROGRAM:**

```
import java.util.Scanner;

public class Employee {
    private String Emp_name;
    private int Emp_id;
    private String Address;
    private String Mail_id;
    private String Mobile_no;
    private float BasicPay;

    public Employee(float BasicPay){
        this.BasicPay = BasicPay;
    }

    public void getDetails(){
        Scanner sc = new Scanner(System.in);

        System.out.println("Enter the Name:");
        Emp_name = sc.nextLine();
        System.out.println("Enter the ID:");
        Emp_id = Integer.parseInt(sc.nextLine());
        System.out.println("Enter the Address:");
        Address = sc.nextLine();
        System.out.println("Enter the E-mail ID:");
        Mail_id = sc.nextLine();
        System.out.println("Enter the Mobile Number:");
        Mobile_no = sc.nextLine();
    }

    void generate(String profession, float DA,float HRA,float PF,float
staffClubFund){
        //net salary calculation
        float grossSalaray = BasicPay + (DA + HRA);
```

```

float netSalary = grossSalaray - (PF + staffClubFund);

System.out.println("\n\t\tSalary Slip");
System.out.println("ID: " + Emp_id);
System.out.println("Name: " + Emp_name);
System.out.println("Profession: " + profession);
System.out.println("Address: " + Address);
System.out.println("E-mail: " + Mail_id);
System.out.println("Mobile: " + Mobile_no);
System.out.println("Basic Pay: Rs." + BasicPay);
System.out.println("DA: Rs." + DA);
System.out.println("HRA: Rs." + HRA);
System.out.println("PF: Rs." + PF);
System.out.println("Staff Club Fund: Rs." + staffClubFund);
System.out.println("\nGross Salary: " + grossSalaray);
System.out.println("Net Salary: " + netSalary);
}

}

```

```

public class AssociateProfessor extends Employee {
    private float BasicPay;
    private float DA;
    private float HRA;
    private float PF;
    private float staffClubFund;
    public String profession = "Associate Professor";

    public AssociateProfessor(float BasicPay){
        super(BasicPay);

        this.BasicPay = BasicPay;
        this.DA = (97*BasicPay) /100;
        this.HRA = (10*BasicPay) / 100;
    }
}
```

```
        this.PF = (12*BasicPay) / 100;
        this.staffClubFund = (0.1f*BasicPay) / 100;
    }

    public void generatePaySlip(){
        super.generate(profession, DA, HRA, PF, staffClubFund);
    }

}

public class AssistantProfessor extends Employee {
    private float BasicPay;
    private float DA;
    private float HRA;
    private float PF;
    private float staffClubFund;
    public String profession = "Assistant Professor";

    public AssistantProfessor(float BasicPay){
        super(BasicPay);

        this.DA = (97*BasicPay) /100;
        this.HRA = (10*BasicPay) / 100;
        this.PF = (12*BasicPay) / 100;
        this.staffClubFund = (0.1f*BasicPay) / 100;
    }

    public void generatePaySlip(){
        super.generate(profession, DA, HRA, PF, staffClubFund);
    }

}
```

```
public class Programmer extends Employee{
    private float BasicPay;
    private float DA;
    private float HRA;
    private float PF;
    private float staffClubFund;
    public String profession = "Programmer";

    public Programmer(float BasicPay){
        super(BasicPay);

        this.DA = (97*BasicPay) /100;
        this.HRA = (10*BasicPay) / 100;
        this.PF = (12*BasicPay) / 100;
        this.staffClubFund = (0.1f*BasicPay) / 100;
    }

    public void generatePaySlip(){
        super.generate(profession, DA, HRA, PF, staffClubFund);
    }
}
```

```
public class Professor extends Employee {
    private float BasicPay;
    private float DA;
    private float HRA;
    private float PF;
    private float staffClubFund;
    public String profession = "Professor";
```

```
public Professor(float BasicPay){  
    super(BasicPay);  
  
    this.DA = (97*BasicPay) /100;  
    this.HRA = (10*BasicPay) / 100;  
    this.PF = (12*BasicPay) / 100;  
    this.staffClubFund = (0.1f*BasicPay) / 100;  
}  
  
public void generatePaySlip(){  
    super.generate(profession, DA, HRA, PF, staffClubFund);  
}  
}  
  
public class Main {  
  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
  
        System.out.println("1.Associate Professor\n2.Professor\n3.Assistant  
Professor\n4.Programmer");  
        System.out.println("Choose Profession:");  
  
        int choice = sc.nextInt();  
  
        System.out.println("\nEnter Basic Pay:");  
        float BasicPay = sc.nextFloat();  
  
        switch (choice){  
            case 1:  
                AssociateProfessor associateProfessor = new  
AssociateProfessor(BasicPay);  
                associateProfessor.getDetails();
```

```
associateProfessor.generatePaySlip();
break;
case 2:
    Professor professor = new Professor(BasicPay);
    professor.getDetails();
    professor.generatePaySlip();
    break;
case 3:
    AssistantProfessor assistantProfessor = new
AssistantProfessor(BasicPay);
    assistantProfessor.getDetails();
    assistantProfessor.generatePaySlip();
    break;
case 4:
    Programmer programmer = new Programmer(BasicPay);
    programmer.getDetails();
    programmer.generatePaySlip();
    break;
default:
    System.out.println("Invalid Selection!!!!");
}
}
}
```

## OUTPUT:

```
cmd C:\Windows\system32\cmd.exe - java employees

D:\IICSE_IIIT\Java Lab>New folder>javac employees.java
D:\IICSE_IIIT\Java Lab>New folder>java employees
enter employee detials/n
enter vname of the employee:
yyyyyy
enter mail_id of the employee:
xxxxxx
enter adress of the employee:
aaaaaa
enter Emp_id:
44444
enter mobile_no:
785645
Employee id:44444
Employee Name:yyyyyy
phone number:785645
Mialid:xxxxxx
Address:aaaaaa
enter vname of the employee:
-
```

Result:

Thus, the Java application for generation of salary slip using inheritance has been written, executed and output is verified.

Exno 4

Date: 28/6/20; 2/9/20

## Implementation of stack ADT

Aim:-

To write a java application stack ADT with necessary exception handling.

Algorithm:-

1. Create a interface to implement stack ADT with necessary abstract methods.

2. Create a class Stack that implements stack interface.

3. In Stack class, declare an array for stack, top pointer and MaxSize Variable.

4. Define a constructor that takes size as parameter and initialize the Stack array with size.

5. The following methods are implemented on class Stack.

6. push - It takes value parameter. If the stack is not full, append the value to array. otherwise issue StackOverflow Error.

7. pop - If the stack is not empty, store the last value to temp variable. decrement the top variable by 1. Return value of temp.

8. IsFull - If the stack is full, return true. Otherwise return false.

9. IsEmpty - If the stack is empty, return true. Otherwise return false.

10. Peak - Return last value from the stack array.

11. PrintStack - print the values stored in the stack of reverse order

12. Create a Main class With Main Method

13. The user is asked to enter the size of the stack.

14. The user is provided with options to choose from.

15. The selected operation will be performed on the stack.

16. Repeat step 14 and 15 until the user opts to exit the program

## **PROGRAM**

```
// Interface Stack  
package stack;  
  
public interface IStack {  
    void push(int a);  
    int pop();  
    boolean isFull();  
    boolean isEmpty();  
    int peak();  
    void printStack();  
}
```

```
//Class Stack  
package stack;
```

```
public class Stack implements IStack{  
    private int[] array;  
    private int top;  
    private int maxSize;  
  
    public Stack(int size){  
        maxSize = size;  
        array = new int[size];  
        top = -1;  
    }  
  
    public void push(int a){  
        try{  
            top += 1;  
            array[top] = a;  
        }catch (Exception e){  
            System.out.println("Stack Overflow!");  
        }  
    }  
}
```

```
        top--;
    }
}

public int pop(){
    int temp = -95456154;
    try{
        temp = array[top];
    }catch (Exception e){
        top++;
    }
    top--;
    return temp;
}

public boolean isFull(){
    return (top >= (maxSize-1));
}

public boolean isEmpty(){
    return (top == -1);
}

public int peak(){
    return array[top];
}

public void printStack(){
    System.out.println();
    if(top != -1) {
        for (int i = top; i >= 0; i--) {
            System.out.println(array[i]);
        }
    }
}
```

```
        }else{
            System.out.println("Stack is empty");
        }
    }
}

//Main Class
package stack;

import java.util.Scanner;

public class Main {

    public static void main(String[] args) {
        Stack myStack;
        Scanner sc = new Scanner(System.in);
        int num;
        System.out.println("Enter the number of slots:");
        myStack = new Stack(sc.nextInt());

        while(true){
            System.out.println("\n1.Push to Stack\n2.Pop from Stack\n3.Peak from
Stack\n4.Print Stack\n"
                + "5.Is the Stack full\n6.Is the Stack empty\n7.Exit");
            System.out.println("Enter your choice:");
            int choice = sc.nextInt();
            switch(choice){
                case 1:
                    System.out.println("Enter the number:");
                    num = sc.nextInt();
                    myStack.push(num);
                    break;
                case 2:
                    num = myStack.pop();

```

```
if(num == -95456154) {
    System.out.println("Stack UnderFlow!");
}else{
    System.out.println("Popped " + num + "from the Stack");
}
break;
case 3:
    num = myStack.peak();
    System.out.println("Peak "+ num + "from the Stack");
    break;
case 4:
    myStack.printStack();
    break;
case 5:
    if(myStack.isFull()){
        System.out.println("The Stack full");
    }else{
        System.out.println("The Stack is not full");
    }
    break;
case 6:
    if(myStack.isEmpty()){
        System.out.println("The Stack is empty");
    }else{
        System.out.println("The Stack is not empty");
    }
    break;
case 7:
    return;
}
}
}
```

## OUTPUT

```
Enter the number of slots:  
3  
  
1.Push to Stack  
2.Pop from Stack  
3.Peak from Stack  
4.Print Stack  
5.Is the Stack full  
6.Is the Stack empty  
7.Exit  
Enter your choice:  
1  
Enter the number:  
1
```

```
1.PushToStack
2.PopFromStack
3.PeekFromStack
4.PrintStack
5.IsTheStackFull
6.IsTheStackEmpty
7.Exit
Enter your choice:
1
Enter the number:
2
|
1.PushToStack
2.PopFromStack
3.PeekFromStack
4.PrintStack
5.IsTheStackFull
6.IsTheStackEmpty
7.Exit
Enter your choice:
1
Enter the number:
3
-----
1.PushToStack
2.PopFromStack
3.PeekFromStack
4.PrintStack
5.IsTheStackFull
6.IsTheStackEmpty
7.Exit
Enter your choice:
4

3
2
1

1.PushToStack
2.PopFromStack
3.PeekFromStack
4.PrintStack
5.IsTheStackFull
6.IsTheStackEmpty
7.Exit
Enter your choice:
2
Popped3fromtheStack

1.PushToStack
2.PopFromStack
3.PeekFromStack
4.PrintStack
5.IsTheStackFull
6.IsTheStackEmpty
7.Exit
Enter your choice:
4

2
1
|
1.PushToStack
2.PopFromStack
3.PeekFromStack
4.PrintStack
5.IsTheStackFull
6.IsTheStackEmpty
7.Exit
Enter your choice:
5
TheStackisnotfull

1.PushToStack
2.PopFromStack
3.PeekFromStack
4.PrintStack
5.IsTheStackFull
6.IsTheStackEmpty
7.Exit
Enter your choice:
6
TheStackisnotempty
```

Result:

Thus, Java application for implementation of stack ADT with necessary exception handling has been written, executed and output is verified.

EN015

Anti 3/9/20; 4/9/20

## Implementation of ArrayList

AIM:-

To write a Java application to perform some ArrayList operations using some userdefined methods.

Algorithm:

1. Create a class ArrayList operation with an ArrayList as a instance variable

2. Define a parameterised constructor with size of the array list as parameter.

3. Create the following Methods:

4. Append - Insert the element to end of the ArrayList with built in add Method.

5. Insert - Insert the element at a particular index in the array list using built in method set.

6. Search - Search for a string in the ArrayList using Linear Search.

7. Strings starting with letter - search for strings that whose first character matches with the given character.

8. Print ArrayList - print all the strings in the array using loop.

9. Create a Main class with Main Method.

10. The user is asked to enter the size of the ArrayList.

11. The user is provided with options to choose from.

12. The selected operation is performed on the ArrayList of strings.

13. Repeat step 11 and 12 until the user stop the exit.

## **Program**

```
package exp4;
```

```
import java.util.ArrayList;
import java.util.Scanner;
```

```
public class ArrayListOperation {
    ArrayList<String> myStringList;
```

```
    public ArrayListOperation(int num){
        myStringList = new ArrayList<>(num);
    }
    public ArrayListOperation(){
        myStringList = new ArrayList<>();
    }
```

```
    public void append(String temp){
        myStringList.add(temp);
    }
```

```
    public void addAtIndex(String temp, int index){
        myStringList.set(index, temp);
    }
```

```
    public void search(String temp){
        for(int i=0; i< myStringList.size(); i++){
            String current = myStringList.get(i);
            if(current.equals(temp)){
                System.out.println(temp + " is found at index " + i);
            }
        }
```

```
        }
    }

public void startWithLetter(char character){
    for(int i=0; i<myStringList.size(); i++){
        String t = myStringList.get(i);
        if(t.charAt(0) == character){
            System.out.println(t);
        }
    }
}

public void printArrayList(){
    System.out.println();
    for(int i =0; i<myStringList.size(); i++){
        System.out.println(myStringList.get(i));
    }
    System.out.println();
}

}

public class Exp4 {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the size of the ArrayList:");
        int size = sc.nextInt();
        ArrayListOperation myList = new ArrayListOperation(size);
        while(true){
            System.out.println("1.Append\n2.Insert\n3.Search\n4.Strings starting
with a letter\n5.Print ArrayList\n6.Exit");
            int choice = sc.nextInt();
            switch(choice){
                case 1:
```

```
System.out.println("Enter the String to append:");
myList.append(sc.next());
break;
case 2:
    System.out.println("Enter the String to insert:");
    String temp = sc.next();
    System.out.println("Enter the index to insert in:");
    int index = sc.nextInt();
    myList.addAtIndex(temp, index);
    break;
case 3:
    System.out.println("Enter the String to search:");
    myList.search(sc.next());
    break;
case 4:
    System.out.println("Enter the character to check for:");
    String a = sc.next();
    char l = a.charAt(0);
    myList.startsWithLetter(l);
    break;
case 5:
    myList.printArrayList();
    break;
case 6:
    return;
default:
    System.out.println("Invalid Choice!");
}

}
}

}
```

## OUTPUT

```
Enter the size of the Array List:  
1  
1.Append  
2.Insert  
3.Search  
4.Stringsstartingwith aletter  
5.PrintArrayList  
6.Exit  
1  
Enter the String to append:  
College  
1.Append  
2.Insert  
3.Search  
4.Stringsstartingwith aletter  
5.PrintArrayList  
6.Exit  
1  
Enter the String to append:  
Department-IT  
1.Append  
2.Insert  
3.Search  
4.Stringsstartingwith aletter  
5.PrintArrayList  
6.Exit  
3  
Enter the String to search:  
College  
Collegeisfoundatindex0  
1.Append  
2.Insert  
3.Search  
4.Stringsstartingwith aletter  
5.PrintArrayList  
6.Exit  
5
```

Result:-

Thus, java application for performing Array list operation has been written, executed and output is verified.

Ex>No: 06

Date: 9/9/20; 10/9/20

### Implementation of Abstract class

Aim:-

To develop a java application for calculation of area of shapes using abstract class.

Algorithm:-

1. Create an abstract class Shape with two member variables of type float and an abstract method called printArea.
2. Create a class Rectangle that extends Shape abstract class.
3. In Rectangle class, define a parameterised constructor and printArea Method to calculate the Area of rectangle using the formula  $A = l \times b$ .
4. Create a class Triangle that extends Shape abstract class.
5. In Triangle class, define a parameterised constructor and printArea Method to calculate the area of rectangle using the formula  $A = \frac{1}{2} b \times h$ .
6. Create a class Circle that extends Shape abstract class.
7. In Circle class, define a parameterised constructor and printArea Method to calculate the area of rectangle using the formula  $A = \pi r \times r$ .
8. Create a Main class and define the Main Method.
9. The user is prompted to choose the shape whose area is to be calculated.
10. Get the values from the user.
11. Initiate the values using the values provided by the user.
12. Call the printArea Method from the class to appropriate shape.
13. The result is displayed to the user.
14. Repeat Step 8 to 12 till the user exits out of the application.

## **Program**

```
import java.util.Scanner;

abstract class Shape{
    float a;
    float b;
    abstract void printArea();
}

public class Rectangle extends Shape{
    public Rectangle(float a, float b){
        this.a = a;
        this.b = b;
    }

    void printArea() {
        float area = a * b;
        System.out.println("The area of the Rectangle is " + area);
    }
}

public class Circle extends Shape {
    public Circle(float a){
        this.a = a;
    }

    void printArea() {
        float area = 3.14f*a*a;
        System.out.println("The area of the Circle is " + area);
    }
}
```

```
public class Triangle extends Shape{  
  
    public Triangle(float a, float b){  
        this.a = a;  
        this.b = b;  
    }  
  
    void printArea() {  
        float area = 0.5f*a*b;  
        System.out.println("The area of the Triangle is " + area);  
    }  
  
}
```

```
public class Main {  
  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
  
        while(true) {  
            System.out.println("\n\t\tArea Calculator");  
            System.out.println("1.Rectangle\n2.Triangle\n3.Circle\n4.Exit");  
            System.out.println("Enter your choice: ");  
            int choice = sc.nextInt();  
            switch (choice) {  
                case 1:  
                    System.out.println("Enter the Length in mm: ");  
                    float l = sc.nextFloat();  
                    System.out.println("Enter the Width in mm: ");  
                    float w = sc.nextFloat();  
                    Shape rectangle = new Rectangle(l, w);  
                    rectangle.printArea();  
                    break;  
            }  
        }  
    }  
}
```

```
case 2:  
    System.out.println("Enter the Base in mm: ");  
    float b = sc.nextFloat();  
    System.out.println("Enter the Height in mm:");  
    float h = sc.nextFloat();  
    Shape triangle = new Triangle(b, h);  
    triangle.printArea();  
    break;  
case 3:  
    System.out.println("Enter the Radius in mm: ");  
    float r = sc.nextFloat();  
    Shape circle = new Circle(r);  
    circle.printArea();  
    break;  
case 4:  
    return;  
default:  
    System.out.println("Invalid choice!!!");  
}  
  
}  
}  
}
```

OUTPUT

```
        AreaCalculator
1.Rectangle
2.Triangle
3.Circle
4.Exit
Enter your choice:
1
Enter the Length in (mm):
220
Enter the Width in (mm):
55
The area of the Rectangle is 122100.0
```

```
        AreaCalculator
1.Rectangle
2.Triangle
3.Circle
4.Exit
Enter your choice:
2
Enter the Base in (mm):
90
Enter the Height in (mm):
150
The area of the Triangle is 6750.0
```

```
        AreaCalculator
1.Rectangle
2.Triangle
3.Circle
4.Exit
Enter your choice:
3
Enter the Radius in (mm):
244
The area of the Circle is 186943.05
```

Result:

Thus, the java application for calculation of area of shapes using abstract class has been written, executed and output is verified.

EX: No: 07

Date: 11/9/20; 16/9/20

## Implementation of Exception handling

AIM:

To develop a java application for the implementation of exception handling.

Algorithm:-

1. Create a class MarkException that extends Exception to represent an exception.
2. In MarkException class, define a parameterized constructor and a string Method to return a string representation of the Exception.
3. Create a class student In student class, define a parameterized constructor and read Method to read the input values from the user.
4. In student class, define method check to check the value of Mark and throw an exception if the value is less than 0 or greater than 100.
5. Create a Main class and define the main method.
6. call the read Method to read the values from the user.
7. call the check Method.
8. The results is displayed to the user by calling the print Method.

## Program

```
public class MarkException extends Exception{
    int m;
    public MarkException(int x)
    {
        m=x;
    }
    public String toString()
    {
        return "invalid mark"+m;
    }
}
import java.util.Scanner;
public class Student {
    String name;
    long rollno;
    int mark;
    void read()
    {
        Scanner s=new Scanner(System.in);
        System.out.println("Enter name");
        name=s.next();
        System.out.println("Enter rollno");
        rollno=s.nextLong();
        System.out.println("Enter mark");
        mark=s.nextInt();
    }
    void check()
    {
        try
        {
            if(mark <0 || mark>100)
```

```
        throw new MarkException(mark);
    }
    catch(MarkException e)
    {
        System.out.println(e);
        mark=0;
    }

}
void print()
{
    System.out.println("Name : "+name);
    System.out.println("Rollno : "+rollno);
    System.out.println("Mark : "+mark);
}
public class Example {
    public static void main(String[] args) {
        Student s=new Student();
        s.read();
        s.check();
        s.print();
    }
}
```

## OUTPUT

- 1. Enter name : Ancy  
Enter rollno:5005  
Enter mark :76

Result:-

Thus, the Java application for the implementation of exception handling has been written, executed and output is verified.

EX-NO:8

Date: 17/12/20; 18/12/20

AIM:-

To Write a java program that reads a file name from the user, display information about whether the file exists, whether the file is readable, or writable, the type of file and the length of the file in bytes.

Algorithm:-

1. Create a class File pgm.
2. Get the file name from the user.
3. Use the file name functions and display the information about the file.
  - a. getName() - displays the name of the file.
  - b. getPath() - displays the path name of the file.
  - c. getParent() - This Method returns the pathname string of this abstract pathname's parent, or null if this pathname does not name a parent directory.
  - d. exists() - checks whether the file exists or not.
  - e. canRead() - basically a check if the file can be read
  - f. canWrite() - verifies the application can write to the file.
  - g. isDirectory() - displays whether it is a directory or not.
  - h. isFile() - displays whether it is a file or not.
  - i. lastModified() - displays the last modified information.
  - j. Length() - displays the size of the file.
  - k. delete() - deletes the file.
4. Invoke the predefined functions and display the information about the file.

## Displaying Information about a file

```
program
import java.io.*;
import java.util.*;
class filepgm
{
    public static void main(String args[])
    {
        String filename;
        Scanner s=new Scanner(System.in);
        System.out.println("Enter the file name ");
        filename=s.nextLine();
        File f1=new File(filename);
        System.out.println("*****");
        System.out.println("FILE INFORMATION");
        System.out.println("*****");
        System.out.println("NAME OF THE FILE "+f1.getName());
        System.out.println("PATH OF THE FILE "+f1.getPath());
        System.out.println("PARENT"+f1.getParent());
        if(f1.exists())
            System.out.println("THE FILE EXISTS ");
        else
            System.out.println("THE FILE DOES NOT EXISTS ");
        if(f1.canRead())
            System.out.println("THE FILE CAN BE READ ");
        else
            System.out.println("THE FILE CANNOT BE READ ");
        if(f1.canWrite())
            System.out.println("WRITE OPERATION IS PERMITTED");
        else
            System.out.println("WRITE OPERATION IS NOT
PERMITTED");
        if(f1.isDirectory())
            System.out.println("IT IS A DIRECTORY ");
        else
            System.out.println("NOT A DIRECTORY");
        if(f1.isFile())
            System.out.println("IT IS A FILE ");
        else
            System.out.println("NOT A FILE");
```

```
        System.out.println("File last modified "+ f1.lastModified());
        System.out.println("LENGTH OF THE FILE "+f1.length());
        System.out.println("FILE DELETED "+f1.delete());
    }
}
```

## Output

```
Enter the filename
Ace.txt
*****
FILE INFORMATION
*****
NAME OF THE FILEAce.txt
PATH OF THE FILEAce.txt
PARENTnull
THE FILE DOES NOT EXISTS
THE FILE CANNOT BE READ
WRITE OPERATION IS NOT PERMITTED
NOT A DIRECTORY
NOT A FILE
File last modified@0
LENGTH OF THE FILE@0
FILE DELETEDfalse
```

Result:

Thus, the java application for display the file information has been written, executed and output is verified.

Ex No: 9  
Date: 23/9/20; 24/9/20

## Implementation of Multithreading

Aim:-

To write a Java program to implement a Multithread application.

Algorithm:-

1. Create a class even which implements first thread that computes the square of the number.
2. run() Method implements the code to be executed when thread gets executed.
3. Create a class odd which implements second thread that computes the cube of the number.
4. Create a third thread that generates random number. If the random number is even, it displays the square of the number.
5. If the random number generated is odd, it displays the cube of the given number.
6. The Multithreading is performed and the task is switched between Multiple threads.
7. The sleep() method makes the thread to suspend for the specified time.

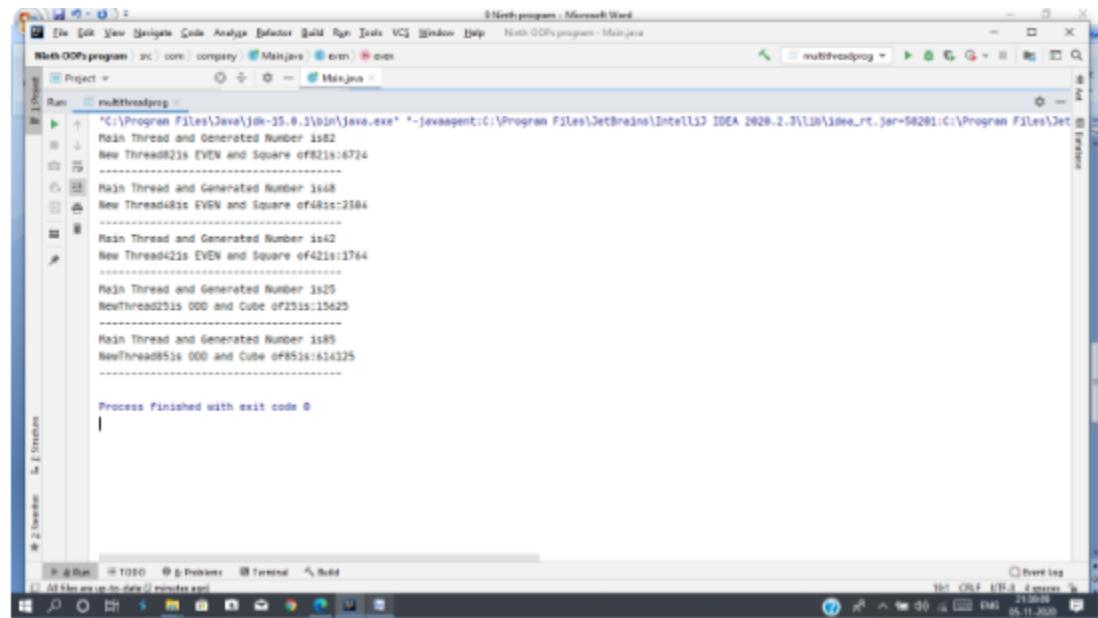
## **Program**

```
import java.util.*;
class even implements Runnable
{
    public int x;
    public even(int x)
    {
        this.x = x;
    }
    public void run()
    {
        System.out.println("New Thread "+ x +" is EVEN and Square of " + x + " is: " + x * x);
    }
}
class odd implements Runnable
{
    public int x;
    public odd(int x)
    {
        this.x = x;
    }
    public void run()
    {
        System.out.println("New Thread "+ x +" is ODD and Cube of " + x + " is: " + x * x * x);
    }
}
class A extends Thread
{
    public void run()
    {
        int num = 0;
        Random r = new Random();
        try
        {
            for (int i = 0; i < 5; i++)
            {
                num = r.nextInt(100);
                System.out.println("Main Thread and Generated Number is " + num);
                if (num % 2 == 0)

```



## Output



The screenshot shows the IntelliJ IDEA interface with the code editor open. The code is a Java program named Main.java, which runs multiple threads to print generated numbers and their properties. The output window displays the following text:

```
*C:\Program Files\Java\jdk-15.0.1\bin\java.exe* "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2020.2.3\lib\idea_rt.jar=54281:C:\Program Files\JetBrains\IntelliJ IDEA 2020.2.3\lib\idea_rt.jar" Main.OOPS.program - Main.java
Run multithreadprog
Main Thread and Generated Number is82
New Thread021is EVEN and Square of82is:6724
-----
Main Thread and Generated Number is48
New Thread421is EVEN and Square of48is:2304
-----
Main Thread and Generated Number is42
New Thread421is EVEN and Square of42is:1764
-----
Main Thread and Generated Number is20
NewThread021is ODD and Cube of25is:15625
-----
Main Thread and Generated Number is85
NewThread851is ODD and Cube of85is:614325
-----
Process finished with exit code 0
```

Result:-

Thus, the java application to implement multithreading has been written, executed and output is verified.

EX: NO: 10

## Implementation of Generic methods

Date: 25/9/20, 30/9/20, 01/10/20,  
7/10/20

AIM:-

To Write a Java program to find the Maximum value  
from the given type of elements using generic Methods.

Algorithm:

1. Create a class named Maxclass
2. In the Max class, create a generic Method, Max to find the Maximum of the give type of elements
3. In the Main Method, call the Max Method with integer, Double string, Boolean and byte elements to find the maximum.

## Program

```
public class MaxClass {  
    public static <T extends Comparable<T>> T max(T... elements) {  
        T max = elements[0];  
        for (T element : elements) {  
            if (element.compareTo(max) > 0) {  
                max = element;  
            }  
        }  
        return max;  
    }  
}  
  
public class Example {  
    public static void main(String[] args) {  
        MaxClass m=new MaxClass();  
        System.out.println("Integer Max: " + m.max(Integer.valueOf(32),  
Integer.valueOf(56), Integer.valueOf(89), Integer.valueOf(3),  
Integer.valueOf(456), Integer.valueOf(78), Integer.valueOf(45)));  
        System.out.println("Double Max: " + m.max(Double.valueOf(5.6),  
Double.valueOf(7.8), Double.valueOf(2.9), Double.valueOf(18.6),  
Double.valueOf(10.25), Double.valueOf(18.6001)));  
        System.out.println("String Max: " + m.max("Strawberry", "Mango",  
"Apple", "Pomegranate", "Guava", "Blackberry", "Cherry", "Orange",  
"Date"));  
        System.out.println("Boolean Max: " + m.max(Boolean.TRUE,  
Boolean.FALSE));  
        System.out.println("Byte Max: " + m.max(Byte.MIN_VALUE,  
Byte.MAX_VALUE));  
    }  
}
```

## Output

The screenshot shows the IntelliJ IDEA interface with the following details:

- Project Bar:** Shows the project structure with "Tenth OOPS program" selected.
- Run Tab:** Set to "Example".
- Output Window:** Displays the program's output:

```
"C:\Program Files\Java\jdk-15.0.1\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2020.2.3\lib\idea_rt.jar=61599:C:\Program Files\JetBrains\IntelliJ IDEA 2020.2.3\lib\idea_rt.jar" -Dfile.encoding=UTF-8 com.company.Mainjava MaxClass@max
IntegerMax:456
DoubleMax:18.6801
StringMax:Strawberry
BooleanMax:true
ByteMax:127

Process finished with exit code 8
```
- Bottom Navigation Bar:** Includes tabs for Run, TODO, Problems, Terminal, and Build.
- Status Bar:** Shows "All files are up-to-date (2 minutes ago)" and file paths like "C:\Users\91\IdeaProjects\Tenth OOPS program\src\com\company\Mainjava.java".

**Result:-**

Thus, the java application to find the maximum value from the given type of elements using generic methods has been written, executed and output is Verified.

Ex: NO: 11

Date: 8/10/20; 9/10/20; 14/10/20  
AIM: 15/10/20

Implementation of Scientific calculator.

To write a Java program to implement scientific calculator.

Algorithm:-

1. Import the Swing packages and awt packages
2. Create the class Scientific calculator that extends Container
3. In the Scientific calculator class, create the container and add controls for digits, scientific calculations and decimal Manipulations.
4. Use appropriate layout manager to lay the controls in the Container
5. Add event handlers to all the controls to handle the events
6. perform the corresponding decimal, numeric and scientific calculations and display the results.
7. In the Main Method, create an object of the Scientific calculator class and Make the frame visible on the screen.

**Program:**

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
public class ScientificCalculator extends JFrame implements ActionListener {
    JTextField tfield;
    double temp, temp1, result, a;
    static double m1, m2;
    int k = 1, x = 0, y = 0, z = 0;
    char ch;
    JButton b1, b2, b3, b4, b5, b6, b7, b8, b9, zero, clr, pow2, pow3, exp,
           fac, plus, min, div, log, rec, mul, eq, addSub, dot, mr, mc, mp,
           mm, sqrt, sin, cos, tan;
    Container cont;
    JPanel textPanel, buttonpanel;

    ScientificCalculator() {
        cont = getContentPane();
        cont.setLayout(new BorderLayout());
        JPanel textpanel = new JPanel();
        tfield = new JTextField(25);
        tfield.setHorizontalAlignment(SwingConstants.RIGHT);
        tfield.addKeyListener(new KeyAdapter() {
            public void keyTyped(KeyEvent keyevent) {
                char c = keyevent.getKeyChar();
                if (c >= '0' && c <= '9') {
                } else {
                    keyevent.consume();
                }
            }
        });
        textpanel.add(tfield);
        buttonpanel = new JPanel();
        buttonpanel.setLayout(new GridLayout(8, 4, 2, 2));
```

```
boolean t = true;
mr = new JButton("MR");
buttonpanel.add(mr);
mr.addActionListener(this);
mc = new JButton("MC");
buttonpanel.add(mc);
mc.addActionListener(this);

mp = new JButton("M+");
buttonpanel.add(mp);
mp.addActionListener(this);

mm = new JButton("M-");
buttonpanel.add(mm);
mm.addActionListener(this);

b1 = new JButton("1");
buttonpanel.add(b1);
b1.addActionListener(this);
b2 = new JButton("2");
buttonpanel.add(b2);
b2.addActionListener(this);
b3 = new JButton("3");
buttonpanel.add(b3);
b3.addActionListener(this);

b4 = new JButton("4");
buttonpanel.add(b4);
b4.addActionListener(this);
b5 = new JButton("5");
buttonpanel.add(b5);
b5.addActionListener(this);
b6 = new JButton("6");
buttonpanel.add(b6);
```

```
b6.addActionListener(this);

b7 = new JButton("7");
buttonpanel.add(b7);
b7.addActionListener(this);
b8 = new JButton("8");
buttonpanel.add(b8);
b8.addActionListener(this);
b9 = new JButton("9");
buttonpanel.add(b9);
b9.addActionListener(this);

zero = new JButton("0");
buttonpanel.add(zero);
zero.addActionListener(this);

plus = new JButton("+");
buttonpanel.add(plus);
plus.addActionListener(this);

min = new JButton("-");
buttonpanel.add(min);
min.addActionListener(this);

mul = new JButton("*");
buttonpanel.add(mul);
mul.addActionListener(this);

div = new JButton("/");
div.addActionListener(this);
buttonpanel.add(div);

addSub = new JButton("+/-");
buttonpanel.add(addSub);
```

```
addSub.addActionListener(this);

dot = new JButton(".");
buttonpanel.add(dot);
dot.addActionListener(this);

eq = new JButton("=");
buttonpanel.add(eq);
eq.addActionListener(this);

rec = new JButton("1/x");
buttonpanel.add(rec);
rec.addActionListener(this);
sqrt = new JButton("Sqrt");
buttonpanel.add(sqrt);
sqrt.addActionListener(this);
log = new JButton("log");
buttonpanel.add(log);
log.addActionListener(this);

sin = new JButton("SIN");
buttonpanel.add(sin);
sin.addActionListener(this);
cos = new JButton("COS");
buttonpanel.add(cos);
cos.addActionListener(this);
tan = new JButton("TAN");
buttonpanel.add(tan);
tan.addActionListener(this);
pow2 = new JButton("x^2");
buttonpanel.add(pow2);
pow2.addActionListener(this);
pow3 = new JButton("x^3");
buttonpanel.add(pow3);
```

```
pow3.addActionListener(this);
exp = new JButton("Exp");
exp.addActionListener(this);
buttonpanel.add(exp);
fac = new JButton("n!");
fac.addActionListener(this);
buttonpanel.add(fac);

clr = new JButton("AC");
buttonpanel.add(clr);
clr.addActionListener(this);
cont.add("Center", buttonpanel);
cont.add("North", textpanel);
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

}

public void actionPerformed(ActionEvent e) {
    String s = e.getActionCommand();
    if (s.equals("1")) {
        if (z == 0) {
            tfield.setText(tfield.getText() + "1");
        } else {
            tfield.setText("");
            tfield.setText(tfield.getText() + "1");
            z = 0;
        }
    }
    if (s.equals("2")) {
        if (z == 0) {
            tfield.setText(tfield.getText() + "2");
        } else {
            tfield.setText("");
            tfield.setText(tfield.getText() + "2");
            z = 0;
        }
    }
}
```

```
        }
    }
    if (s.equals("3")) {
        if (z == 0) {
            tfield.setText(tfield.getText() + "3");
        } else {
            tfield.setText("");
            tfield.setText(tfield.getText() + "3");
            z = 0;
        }
    }
    if (s.equals("4")) {
        if (z == 0) {
            tfield.setText(tfield.getText() + "4");
        } else {
            tfield.setText("");
            tfield.setText(tfield.getText() + "4");
            z = 0;
        }
    }
    if (s.equals("5")) {
        if (z == 0) {
            tfield.setText(tfield.getText() + "5");
        } else {
            tfield.setText("");
            tfield.setText(tfield.getText() + "5");
            z = 0;
        }
    }
    if (s.equals("6")) {
        if (z == 0) {
            tfield.setText(tfield.getText() + "6");
        } else {
            tfield.setText("");
        }
    }
}
```

```
        tfield.setText(tfield.getText() + "6");
        z = 0;
    }
}
if (s.equals("7")) {
    if (z == 0) {
        tfield.setText(tfield.getText() + "7");
    } else {
        tfield.setText("");
        tfield.setText(tfield.getText() + "7");
        z = 0;
    }
}
if (s.equals("8")) {
    if (z == 0) {
        tfield.setText(tfield.getText() + "8");
    } else {
        tfield.setText("");
        tfield.setText(tfield.getText() + "8");
        z = 0;
    }
}
if (s.equals("9")) {
    if (z == 0) {
        tfield.setText(tfield.getText() + "9");
    } else {
        tfield.setText("");
        tfield.setText(tfield.getText() + "9");
        z = 0;
    }
}
if (s.equals("0")) {
    if (z == 0) {
        tfield.setText(tfield.getText() + "0");
    }
}
```

```

    } else {
        tfield.setText("");
        tfield.setText(tfield.getText() + "0");
        z = 0;
    }
}

if (s.equals("AC")) {
    tfield.setText("");
    x = 0;
    y = 0;
    z = 0;
}

if (s.equals("log")) {
    if (tfield.getText().equals("")) {
        tfield.setText("");
    } else {
        a = Math.log(Double.parseDouble(tfield.getText()));
        tfield.setText("");
        tfield.setText(tfield.getText() + a);
    }
}

if (s.equals("1/x")) {
    if (tfield.getText().equals("")) {
        tfield.setText("");
    } else {
        a = 1 / Double.parseDouble(tfield.getText());
        tfield.setText("");
        tfield.setText(tfield.getText() + a);
    }
}

if (s.equals("Exp")) {
    if (tfield.getText().equals("")) {
        tfield.setText("");
    } else {

```

```

        a = Math.exp(Double.parseDouble(tfield.getText()));
        tfield.setText("");
        tfield.setText(tfield.getText() + a);
    }
}

if (s.equals("x^2")) {
    if (tfield.getText().equals("")) {
        tfield.setText("");
    } else {
        a = Math.pow(Double.parseDouble(tfield.getText()),
2);
        tfield.setText("");
        tfield.setText(tfield.getText() + a);
    }
}

if (s.equals("x^3")) {
    if (tfield.getText().equals("")) {
        tfield.setText("");
    } else {
        a = Math.pow(Double.parseDouble(tfield.getText()),
3);
        tfield.setText("");
        tfield.setText(tfield.getText() + a);
    }
}

if (s.equals("+/-")) {
    if (x == 0) {
        tfield.setText("-" + tfield.getText());
        x = 1;
    } else {
        tfield.setText(tfield.getText());
    }
}

if (s.equals(".")) {

```

```
if (y == 0) {
    tfield.setText(tfield.getText() + ".");
    y = 1;
} else {
    tfield.setText(tfield.getText());
}
}

if (s.equals("+")) {
    if (tfield.getText().equals("")) {
        tfield.setText("");
        temp = 0;
        ch = '+';
    } else {
        temp = Double.parseDouble(tfield.getText());
        tfield.setText("");
        ch = '+';
        y = 0;
        x = 0;
    }
    tfield.requestFocus();
}
if (s.equals("-")) {
    if (tfield.getText().equals("")) {
        tfield.setText("");
        temp = 0;
        ch = '-';
    } else {
        x = 0;
        y = 0;
        temp = Double.parseDouble(tfield.getText());
        tfield.setText("");
        ch = '-';
    }
    tfield.requestFocus();
}
```

```
        }
        if (s.equals("/")) {
            if (tfield.getText().equals("")) {
                tfield.setText("");
                temp = 1;
                ch = '/';
            } else {
                x = 0;
                y = 0;
                temp = Double.parseDouble(tfield.getText());
                ch = '/';
                tfield.setText("");
            }
            tfield.requestFocus();
        }
        if (s.equals("*")) {
            if (tfield.getText().equals("")) {
                tfield.setText("");
                temp = 1;
                ch = '*';
            } else {
                x = 0;
                y = 0;
                temp = Double.parseDouble(tfield.getText());
                ch = '*';
                tfield.setText("");
            }
            tfield.requestFocus();
        }
        if (s.equals("MC")) {
            m1 = 0;
            tfield.setText("");
        }
        if (s.equals("MR")) {
```

```

        tfield.setText("");
        tfield.setText(tfield.getText() + m1);
    }
    if (s.equals("M+")) {
        if (k == 1) {
            m1 = Double.parseDouble(tfield.getText());
            k++;
        } else {
            m1 += Double.parseDouble(tfield.getText());
            tfield.setText("'" + m1);
        }
    }
    if (s.equals("M-")) {
        if (k == 1) {
            m1 = Double.parseDouble(tfield.getText());
            k++;
        } else {
            m1 -= Double.parseDouble(tfield.getText());
            tfield.setText("'" + m1);
        }
    }
    if (s.equals("Sqrt")) {
        if (tfield.getText().equals("")) {
            tfield.setText("");
        } else {
            a = Math.sqrt(Double.parseDouble(tfield.getText()));
            tfield.setText("");
            tfield.setText(tfield.getText() + a);
        }
    }
    if (s.equals("SIN")) {
        if (tfield.getText().equals("")) {
            tfield.setText("");
        } else {

```

```

        a = Math.sin(Double.parseDouble(tfield.getText()));
        tfield.setText("");
        tfield.setText(tfield.getText() + a);
    }
}

if (s.equals("COS")) {
    if (tfield.getText().equals("")) {
        tfield.setText("");
    } else {
        a = Math.cos(Double.parseDouble(tfield.getText()));
        tfield.setText("");
        tfield.setText(tfield.getText() + a);
    }
}

if (s.equals("TAN")) {
    if (tfield.getText().equals("")) {
        tfield.setText("");
    } else {
        a = Math.tan(Double.parseDouble(tfield.getText()));
        tfield.setText("");
        tfield.setText(tfield.getText() + a);
    }
}

if (s.equals("=")) {
    if (tfield.getText().equals("")) {
        tfield.setText("");
    } else {
        temp1 = Double.parseDouble(tfield.getText());
        switch (ch) {
        case '+':
            result = temp + temp1;
            break;
        case '-':
            result = temp - temp1;

```

```

        break;
    case '/':
        result = temp / temp1;
        break;
    case '*':
        result = temp * temp1;
        break;
    }
    tfield.setText("");
    tfield.setText(tfield.getText() + result);
    z = 1;
}
if (s.equals("n!")) {
    if (tfield.getText().equals(""))
        tfield.setText("");
    } else {
        a = fact(Double.parseDouble(tfield.getText()));
        tfield.setText("");
        tfield.setText(tfield.getText() + a);
    }
}
tfield.requestFocus();
}

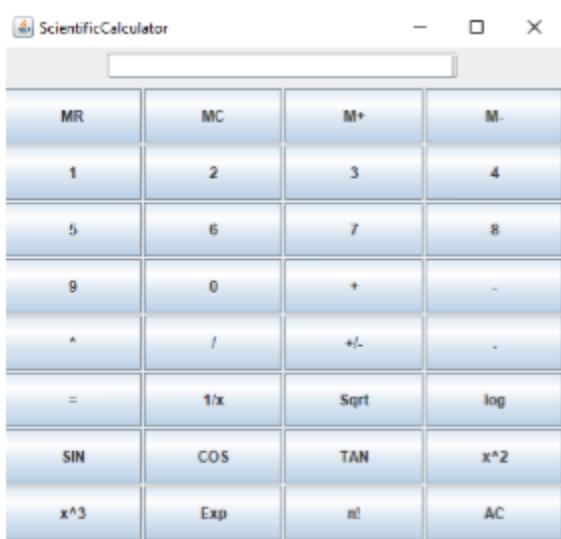
double fact(double x) {
    int er = 0;
    if (x < 0) {
        er = 20;
        return 0;
    }
    double i, s = 1;
    for (i = 2; i <= x; i += 1.0)
        s *= i;
}

```

```
        return s;
    }
}

public class Example {
    public static void main(String[] args) {
        ScientificCalculator f = new ScientificCalculator();
        f.setTitle("ScientificCalculator");
        f.pack();
        f.setVisible(true);
    }
}
```

## Output:



Result:-

Thus, the java application to implement scientific calculator has been written, executed and output is verified.