WRITE A PROGRAM TO FIND THE AREA OF SQUARE, RECTANGLE AND CIRCLE USING METHOD OVERLOADING.

AIM:

To write a program to find the area of Square, Rectangle and cide circle using method overloading.

PROCEDURE:

Step 1: Start
Step 2: Define area 1 = a*a in method void area (int a) for area of Square.
Step 3: Define area 2 = a*b in the method void area (int l, int b) for area of rectangle.
Step 4: Define area 3 = 3.14*r*r in method void area (float r) for area of circle.
Step 5: Read the values of a, l, b & r
Step 6: call the methods
area (a), area (l, b) & area (r)
Step 7: Print the results.
Step 8: Stop.

PROGRAM :

```
import java.io.*;
import java.io.DataInputStream;
class methodoverloading
{
    void area (int a)
    {
    float area1 = a*a;
    System.Out.println("Area of SQuare is"+ area1
                +"SQ units");
    }
    void area (int 1, float b)
    {
    float area2 = 1*b;
    system.out.println("Area of the rectangle is"+
                area2+" SQ units");
    }
     void area (float r)
    {
      double area3 = 3.14*r*r;
      System.out.println("Area of the circle is"+
                area3+" SQ units");
      }
}
class overload
{
   Print static void main (string args[])throws
                IO Exception
```

```
{
int a,l,b;
float r;
Data Input Stream in= new Data input Stream
                        (system.in);
System.Out.println("Enter the a, l, b, r values:");
a= Integer.parseInt(in.read line());
l = Integer.parseInt(in.read line());
b=Integer.parseInt(in.read line());
r=Integer.parse Float(in.read line());
Method overloading a1=new method overloading()
a1.area(a);
a1.area(l,b);
a1.area(r);
}
}.
```

OUTPUT:

```
D:\n java > set path = c:\jdk 1.5.0\bin
D:\n Java > javac Overload.java
D:\n java > java overload
Enter the a, l, b, r values:
5
4
3
5.0
Area of the Square is 25.0 Sq units
Area of the Rectangle is 12.0 Sq units
Area of the Circle is 78.5 Sq units
```

RESULT:

Thus, the above java Programming has been executed Successfully and the output are verified.

Expt.No.: 2
Date :

Page No.: 7

WRITE A PROGRAM TO SORT THE LIST OF NUMBERS USING COMMAND LINE ARGUMENTS.

AIM:

To write a program to sort the list of Numbers using command line arguments.

PROCEDURE:

Step 1: Start
Step 2: Declare a[]=new int a[20], i, temp as integer.
Step 3: Print the given arguments using for loop
Step 4: Using the loop with i=0 until i<args.length repeat step 5.
Step 5: using for loop with j=i+1 until j<args.length Repeat step 6.
Step 6: if (a[i]>a[j]) then
temp = a[i]; a[i]= a[j]; a[j]=temp;
End if
j++; end for
i++; end for
Step 7: Print the Sorted list of Numbers using for loop.
Step 8: Stop.

PROGRAM:

```
import java.io.*;
class sort
{
    Public static void main (String args[])
    {
        int a[] = new int [20];
        int j, temp, i;
        System.out.println("Given arguments:");
        for (i=0; i<args.length; i++)
        {
        System.out.println (+a[i]);
        }
        for (i=0; i< args.length ; i++)
        {
            for (j=i+1; j< args.length; j++)
            {
            if (a[i] > a[j])
              {
                    temp = a[i];
                    a[i] = a[j];
                    a[j] = temp;
              }
            }
        }
        system.out.println("sorted result:");
        for (i=0; i<args.length ; i++)
        {
          System.out.println (+a[i]);
```

3

3

OUTPUT:

```
D:\njava> set path = c:\jdk1.5.0\bin
D:\njava>java c Sort.java
D:\njava>java Sort 30 10 50 20 40
Given arguments:
30
10
50
20
40
Sorted Result:
10
20
30
40
50
```

RESULT:

Thus, the Java Programming has been created Successfully and the output are verified.

Expt.No.: 3
Date :

Page No.:11

WRITE A PROGRAM IN MULTIPLY THE GIVEN TWO MATRICES.

AIM:

To write a program in multiply the given two matrices.

PROCEDURE:

Step 1: Start

Step 2: Declare m and n integer variables and read Number of rows and columns of the first matrix.

Step 3: Declare p and q integer variables and real number of rows and columns of second matrix.

Step 4: Declare initialize

int a[][] = new int[m][n]; int b[][] = new int[p][q];

int c[][] = new int[m][n];

Step 5: if (n==p) then, Read value of first matrix by for loop

Read the value of second matrix, for loop

Step 6: using for loop with i=0 until i<m repeat Step 7.

Step 7: using for loop with j=0 until j<q repeat Step 8.

Step 8: Compute c[i][j]=0;

using for loop with k=0 until k<n repeat Step 9.

Step 9: Compute c[i][j] += a[i][k] * b[k][j];
Print c[i][j];
++k; end for
Print "\n"
++j; end for
++i; end for

Step 10: else
Print "Matrix cannot be multiplied!!"
end if

Step 11: Stop.

Output:

D:\njava>set path=c:\jdk1.5.0\bin
D:\njava>javac matrix.java
D:\njava>java matrix
Enter the rows and columns of first matrix:
2
2
Enter the rows and columns of second matrix:
2
2
Enter the first matrix:
1
2
3
4

PROGRAM:

```
import java.io.*;
import java.io.DataInputstream;
class matrix
{
  Public static void main (string args[] )throws
IO Exception.
  {
    int m,n,P, Q, I,J,K;
    DataInputstream in=new DataInputstream (system.in);
    System.out.println ("Enter the rows and columns of
                         first matrix:");
    m =Integer.Parse Int (in.read line c);
    n =Integer.Parse Int (in.read line c);
    System.out.println ("Enter rows and columns of
                         Second matrix");
    P= Integer.Parse Int(in.read line c);
    Q = Integer.Paise Int(in.read line c);
    int a[][]=new int [m][n];
    int b[][]=new int [P][Q];
    int c[][]=new int [m][n];
   if(n==P)
     System.out.println ("In Enter first matrix :In");
     for(i=0; i<m; i++)
       for(j=0; j<n ;j++)
         a[i][j] =integer.Parse int (in.read line c));
   System.out.println("In The multiplication result
matrix is:In");
    for(i=0; i<m; ++i)
    {
```

```
        for (j=0; j<q ; ++j)
        {
            c[i][j]=0;
        for (k=0 ; k<n; ++k)
            c[i][j] = c[i][j] + (a[i][k]*b[k][j]);
        System.out.print ("\t" + c[i][j]);
        }
        System.out.println ("\n");
        }
    }
    else
    System.out.println ("\n Sorry!!! matrix multiplication
cant be done");
    }
}.
```

Enter the Second matrix :
5
6
7
8
The multiplication result matrix is

19 22
43 50

RESULT:

Thus, the above Java Programming has been created successfully and the output are verified.

Expt.No.: 4
Date :

Page No.: 17

WRITE A PROGRAM TO DESIGN A CLASS TO REPRESENT A BANK ACCOUNT. INCLUDE THE FOLLOWING: DATA MEMBERS: NAME OF THE DEPOSITER, ACCOUNT NUMBER, TYPE OF ACCOUNT, AND BALANCE AMOUNT IN ACCOUNT. METHODS: TO ASSIGN INITIAL VALUES. TO DEPOSIT IN AMOUNT. TO WITHDRAW AN ACCOUNT AFTER CHECKING BALANCE AND TO DISPLAY THE NAME AND BALANCE.

AIM:

To write a program to design a class to represent a Bank account. Include the following: Data members: Name of the depositer, Account number, type of Account. And Balance amount in account. Methods: To Assign initial values. To deposit in amount. To withdraw an account after checking Balance and to display the name and Balance.

PROCEDURE:

Step 1: Start

Step 2: Define a class Bank with required variables and methods to read required values from user. To Print menu and Switch between the methods by calling and Passing arguments and to exit.

Step 3: Define another class bank Acc with the methods deposit, withdraw and display to perform respective operation Initialize balance to 500.

Step 4: Inside deposit method,
Compute balance = balance + amount;
Step 5: Inside withdraw method, compute
if ((bal > amount) && (bal >= 500)) then
bal = bal - amount;
else
Print "Insufficient funds!! Please maintain
-minimum balance 000";
Step 6: Inside Display method,
Print "Available balance:" + bal";
Step 7: stop.

OUTPUT:

D:\nj ava > set Path = C:\jdk1.5.0\bin
D:\nj ava > java acc Bank.java
D:\nj ava > java Bank
Enter the Account Number:
1113
Enter the Account holder Name:
Naresh kumar. M
Enter the type of Account
current
menu
1. Deposit
2. withdraw
3. Balance enquiry
4. Exit

PROGRAM:

```
import java.io.*;
import java.io.DataInputStream;
class Bank
{
  Public static void main (string args[])throws Io
                                        Exception
  {
  int accno,ch;
  float amount;
  String name, atype;
 DataInput Stream in = new DataInput Stream (System.in);
 System.out.println("Enter the Account holder name:");
 name=in.readline();
 System.out.println("Enter the type of account");
 atype =in.read.line();
 bank acc a=new bank.acc();
 do
 {
   System.out.println("\n menu\n");
   System.out.println("1. Deposit\n");
   System.out.println("2. withdraw\n");
   System.out.println("3. Balance enquiry\n");
   System.out.println("4. Exit\n");
   System.out.println("Enter your choice :");
   ch=Integer.ParseInt(in.read line());
Switch (ch)
{
  Case 1: System.out.println("Enter amount to be
                              Deposited :");
```

```
            amount = Float.parseFloat(in.readLine());
            a.deposit(amount);
            break;
    case 2: System.out.println("Enter Amount to be withdrawn:");
            amount = Float.parseFloat(in.readLine());
            a.withdraw(amount);
            break;
    case 3: a.display();
            break;
    case 4: exist();
    default: system.out.println("Enter the correct choice");
    }
}
   while (ch!=4);
   System.out.println("INDIAN OVERSEAS BANK - KRISHNAGIRI");
   System.out.println("Account Number:" +accno);
   System.out.println("Account Holder Name:"+name);
   System.out.println("Account Type:"+atype);
   a.display();
   System.out.println("Thank you for your service");
   }
}
class bank acc
{
   Public Static float bal = 500.0f;
   void deposit(float amount)
     {
        bal = bal + amount;
        System.out.println("Deposited! Account Balance is"+bal);
     }
```

```
Void withdraw (float amount)
{
  if (cbal > amount) && (bal >= 500))
  {
    bal = bal - amount;
    System.Out.println("withdrawn ! Account balance is "+bal);
  }
  else
  {
  System.out.println("Insufficient funds!! Please maintain
      minimum balance 500");
  }
}
void display()
{
  System.out.println("Available Balance:"+bal);
}
}
```

Enter your choice
1
Enter amount to be deposited :
20000
Deposited ! Account Balance is 20500.0
menu
1. Deposit
2. withdraw
3. Balance Enquiry
4. Exit
Enter your choice :
2
Enter the Amount to be withdrawn :
1000
withdrawn ! Account Balance is 19500.0
menu
1. Deposit
2. withdraw
3. Balance Enquiry
4. Exit
Enter your choice :
2
Enter the amount to be withdrawn :
19400
Insufficient funds !! Please maintain minimum balance
500
menu
1. Deposit
2. Withdraw.

3. Balance Enquiry.
4. Exit
Enter your choice:
3
Available Balance is: 19500
menu
1. Deposit
2. withdraw
3. Balance enquiry.
4. Exit
Enter your choice:
4
INDIAN OVERSEAS BANK - KRISHNAGIRI
ACCOUNT NUMBER :1001
Account Holder Name : Naresh kumar.M.
Account Type : current
Available Balance is: 19500
Thank you for your service.

RESULT:

Thus, the above java programming has been created successfully and the output are verified.

Expt.No.: 5
Date :

Page No.: 25

WRITE A PROGRAM THAT IMPORT THE USER DEFINED PACKAGE AND ACCESS THE MEMBER VARIABLE OF CLASSES THAT CONTAINED BY PACKAGE.

AIM:

To write a program that import the user defined Package and Access the member variable of classes that contained by Package.

PROCEDURE:

Factorial.java

Step 1: Start
Step 2: Declare a package fact
Step 3: Inside void fact (int n) method
int i, fact=1;
using for with i=1 until i<=n repeat
fact=fact*i;
Print the value of fact
end for
Step 4: Stop

Step 1: Start
Step 2: Import Package fact
Step 3: Read the value of n.
Step 4: Call the method fact(n)
Step 5: Stop.

PROGRAM:

```
Package fact;
Public class factorial
{
    Public void fact (int n)
    {
        int i, fact = 1;
        for (i=1; i<=n; i++)
        {
         fact = fact * i;
        } System.out.println("The factorial of" +n+ "is:
                                + fact);
        }
    }
import fact.*;
import java.io.*;
import java.io.DataInputStream;
class fcal
{
   Public Static void main (String args []) throws IO
                                        Exception
   {
     int n;
     DataInputStream in = new DataInputStream (System.in);
     System.out.println("Enter the N value:");
     n = Integer.parseInt(in.readLine());
     factorial f = new factorial();
     f.fact(n);
   }
}
```

OUTPUT:

```
D:\nJava>cd fact
D:\nJava\fact>set path=c:\jdk1.5.0\bin
D:\nJava\fact>javac factorial.java
D:\nJava\fact>java factorial
D:\nJava\javac fact>cd..
D:\nJava>set path=c:\jdk1.5.0\bin
D:\nJava>javac fcal.java
D:\nJava>java fcal
Enter the N value:
5
The factorial of 5 is :120
```

RESULT:

Thus, the above Java Programming has been created successfully and the output are verified.

Expt.No.: 6
Date :

Page No.: 29

WRITE A PROGRAM TO HANDLE THE EXCEPTION USING TRY AND HANDLE MULTIPLE CATCH BLOCKS.

AIM:

TO write a program to Handle the exception using try and handle multiple catch Blocks.

PROCEDURE:

Step 1: Start
Step 2: Declare a[] = {0,25} & b=5
Step 3: Inside try block
int x= a[5]/b-a[1];
Step 4: Define catch (Arithmetic exception e)
Print Division by zero.
Step 5: Define catch (Array Index out of Bounds Exception e)
Print Array Index Error.
Step 6: Define catch (Array Store Exception e)
Print wrong Data type.
Step 7: Inside try block.
int y=a[1]/a[0];
Step 8: Define catch (Arithmetic Exception e)
Print user Errors.
Step 9: Stop.

PROGRAM:

```
import java.io.*;
class exp
{
  public static void main (String args[])
   {
      int a[]={0,25};
      int b=5;
      try
      {
         int x=a[2]/b-a[1];
      }
    catch (Arithmetic Exception e)
    {
       System.out.println("Division by zero");
    }
    catch (Array Index Out of Bounds Exception e)
    {
        System.out.println("Array Index error");
    }
    catch (Array Store exception e)
    {
      System.out.println("wrong Data type");
    }
try
{
   int y = a[1]/a[0];
}
```

```
        Catch (Arithmetic Exception e)
        {
          System.out.println("In user error" +e);
        }
    }
}
```

Expt.No.:
Date :

Page No.: 21

OUTPUT:

```
D:\nJava >set Path=c:\jdk 1.5.0\bin
D:\nJava >Java c exp.java
D:\nJava >Java exp.
EXCEPTION HANDLING
* * * * * * * * * * * * *
Array Index Error.

User error java.lang.Arithmetic Exception : / by zero.
```

RESULT:

Thus, the Java programming has been created successfully and the output are verified.

WRITE A PROGRAM TO ILLUSTRATE THE USE OF MULTI THREADS.

AIM:

To write a program to illustrate the use of multi Threads.

PROCEDURE:

Step 1: Start
Step 2: Define class A which extends thread.
Step 3: Using for loop with i=0 until i<=5 repeat
Step 4: Print "From Thread A:+i" end for.
define class B which extends Thread
Step 5: using for loop with j=0 until j<=5 repeat
Print "From thread A:+j" end for.
Step 6: Define class C which extends Thread.
Step 7: Define using for loop with k=0 until
k<=5 repeat
Print "From thread A:+k" end for.
Step 8: Inside main method. Define thread A.
Thread B & thread C.
Step 9: Set Thread C to maximum priority, Thread B to Original priority +1 and thread A to the minimum priority.
Step 10: Start all the treads.
Step 11: Stop.

PROGRAM:

```
import java.io.*;
import java.util.*;
class A extends Thread
{
  Public void run()
  {
    for(int i=0; i<=5; i++)
    {
      System.out.println("\t From thread A:"+i);
    }
    System.out.println("Exit from A");
  }
}
class B extends Thread
{
  public void run()
  {
    for (int j=0; j<=5; j++)
    {
      System.out.println("\t From thread B:"+j);
    }
    System.out.println("Exit from B");
  }
}
class C extends Thread
{
  public void run()
  {
```

```
        for (int k=0; k<=5; k++)
        {
          System.out.println("It From thread C:"+k);
        }
        System.out.println("Exit from C");
      }
}
class Threadset
{
   Public static void main(string args[])
   {
      A Thread A=new A();
      B Thread B=new B();
      C Thread C=new C();
     Thread C.set Priority(Thread.MAX-PRIORITY);
     Thread B.Set Priority(Thread A.get Priority()+1);
     Thread A.Set priority (Thread.MIN-PRIORITY);
     System.out.println("start Thread A");
     Thread A.start();
     System.Out println("start Thread B");
     Thread B.start();
     System.out.println("start Thread C");
     Thread C.start();
   }
}
```

Output:

```
D:\n Java>set path=c:\jdk1.5.0\bin
D:\n Java>javac Thread Set.java
D:\n Java>java Thread Set
Start Thread A
Start Thread B
    From thread A:0
Start Thread C
    From thread B:0
    From thread A:1
    From thread B:0
    From thread C:0
    From thread A:2
    From thread B:2
    From thread C:1
    From thread A:3
    From thread B:3
    From thread C:2
    From thread A:4
    From thread B:4
    From thread C:3
    From thread A:5
    From thread B:5
    From thread C:4
Exit from A
Exit from B
    From thread C:5
Exit from C
```

Expt.No.:
Date:

Page No.: 3.1

RESULT:

Thus, the Java Programming has been created successfully and the output is verified.

Expt No: 8
Date:

Page No: 29

WRITE A PROGRAM TO CREATE STUDENT REGISTRATION FORM USING APPLET WITH NAME, ADDRESS, SEX, CLASS, EMAIL-ID.

AIM:

To write a Program to create Student Registration form using Applet with name, Address, Sex, class, Email-ID.

PROCEDURE:

Step 1: Start
Step 2: Define class window extends frame implements Action Listener.
Step 3: Define labels for "student Registration form", "Name", "Address", "Sex", "class", and "Email-ID" in respective text boxes.
Step 4: Define and enter values of "student registration form", "Name", "Address", "sex", "class", and "Email-ID" in respective text boxes.
Step 5: Add all elements to screen
Step 6: Call repaint() method set to paint all the elements.
Step 7: Inside main method set object visible to "True".
Step 8: Stop.

PROGRAM:

```
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
class window extends Frame implements ActionListener.
{
    Public window (String title)
    {
    Super (title);
    Set Layout (new Flowlayout (Flow layout.LEFT));
    Label l1 = new label ("Student Registration Form");
    Label l2 = new label ("Name");
    Label l3 = new label ("Address");
    Label l4 = new label ("Sex");
    Label l5 = new label ("class");
    Label l6 = new label ("email-ID");
    Text Field t1 = new Text Field (25);
    TextField t2 = new Text Field (25);
    Text Field t3 = new Text Field (25);
    Text Field t4 = new Text Field (25);
    Text Field t5 = new Text Field (25);
    t1.set Text ("Nareshkumar.M");
    t2.set Text ("krishnagiri");
    t3.set Text ("male");
    t4.Set Text ("II-B.sc[c.s]");
    t5.set Text ("nareshkBsc.m@gmail.com");
    add (l1);
    add (l2);
    add (t1);
```

```
        add(l3);
        add(t2);
        add(l4);
        add(t3);
        add(l5);
        add(t4);
        add(l6);
        add(t5);
        add window listner (new my window Adapter());
        Show();
}
    Public void action Performed (Action Event e)
    {
        repaint();
    }
    Class my window Adapter extends window Adapter
{
    Public void window closing (window Event e)
    {
        System.exit(0);
    }
}
}
Class student
{
    Public static void main (string args[])
    {
        window Obj = new window ("student Registration Form");
        Obj.set visible (true);
        Obj.set size (600, 600);
}
}
```

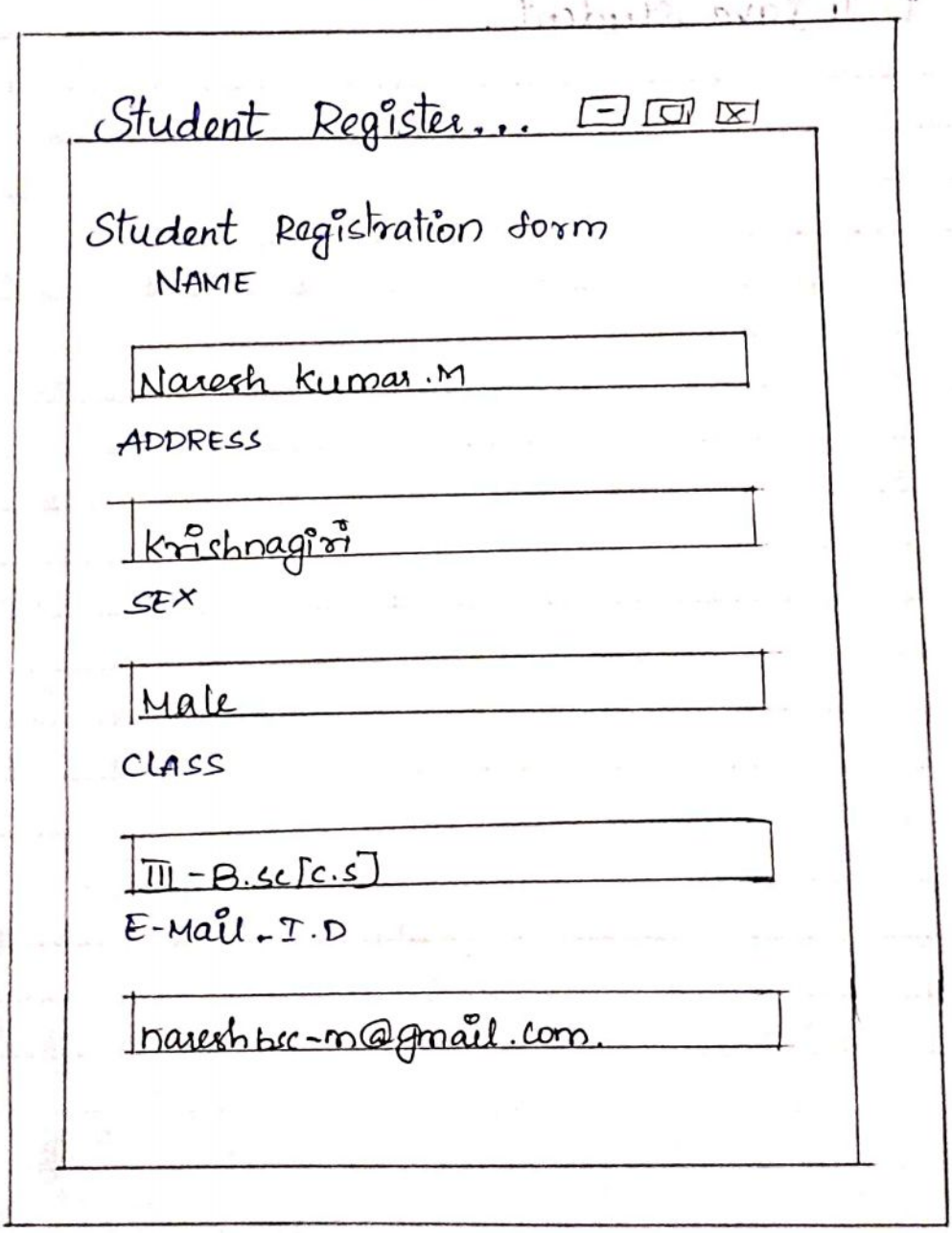
Expt.No.:
Date :

Page No.:11

OUTPUT:

```
D:\n java -set path=c:\jdk 1.5.0\bin
D:\n java c student.java
D:\n java student.
```

OUTPUT:


Student Register...
Student Registration form
NAME
Naresh kumar.M
ADDRESS
krishnagiri
SEX
Male
CLASS
III - B.sc[c.s]
E-Mail.I.D
nareshbsc-m@gmail.com.

Expt.No.:
Date :

RESULT:

Thus, the Java Programming has been Created Successfully and the output are verified.

Expt.No.: 9
Date : 9

Page No.: 45

WRITE A PROGRAM TO DRAW THE LINE, RECTANGLE, OVAL, TEXT USING THE GRAPHICS METHOD.

AIM:

To write a program to draw the line, rectangle, Oval, Text using the Graphics method.

PROCEDURE:

Step 1: Start.

Step 2: Define class draw extends Frame implements Action Listener.

Step 3: Use the methods drawLine(), drawRect(), Set color(), fillRoundRect(), Draw oval() & drawstring() as required to insert lines, Rectangles, ovals, strings and set color to them.

Step 4: Add all the elements to screen.

Step 5: Call Repaint() method to paint all the elements

Step 6: Inside 'main method' set object visible to "True".

Step 7: Stop.

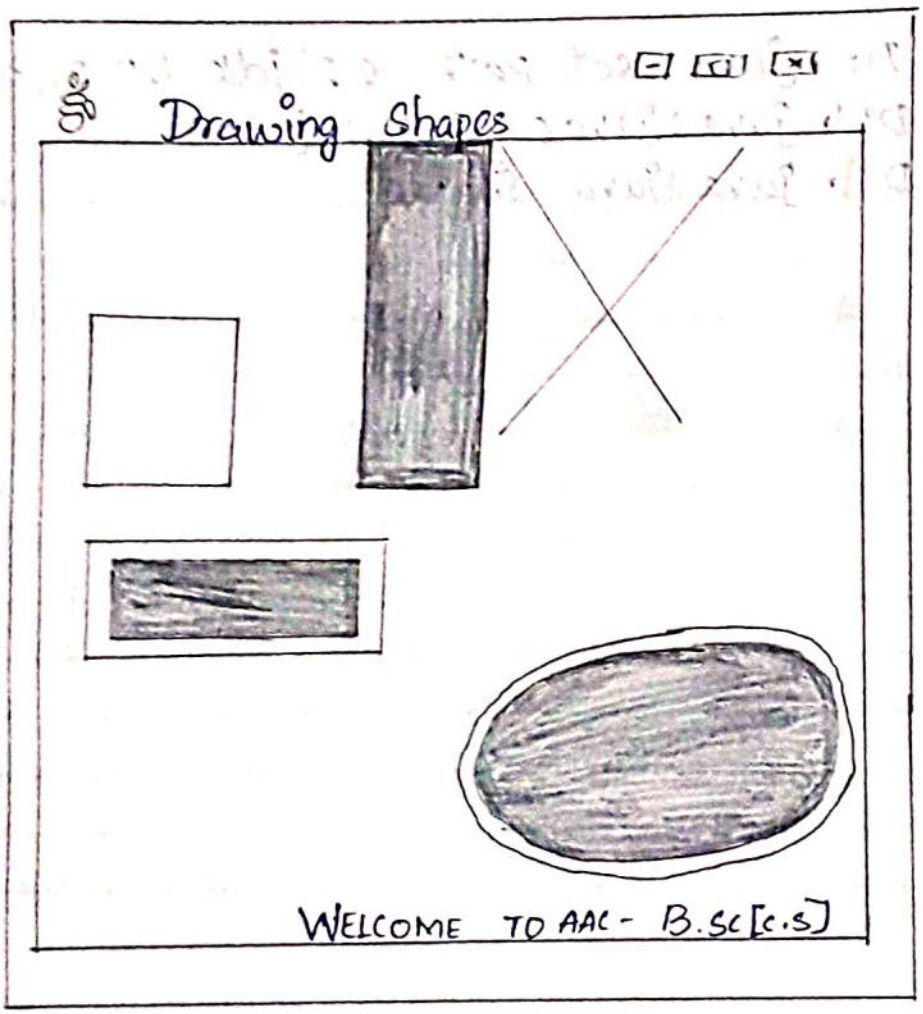
PROGRAM:

```
import java.io.*;
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
class draw extends Frame implements ActionalListener.
{
  public draw (String title)
  {
  super (title);
  set layout (new FlowLayout (FlowLayout LEFT));
  }
  public void Paint (Graphics g)
  {
  g.drawline (100,10,  230,140);
  g.drawline (100,140,230,10);
  g.drawRect (10,60,140,30);
  g.set color (color.yellow);
  g.fill Rect (60,10,30,80);
  g.set color (color.Orange);
  g.draw Round Rect (10,100,80,50,10,10);
  g.set color (color.Blue);
  g.fill Round Rect (90,110,60,30,5,5);
  g.set color (color.red);
  g.fill Round Rect (20,110,60,30,5,5);
  g.set Color (Color.green);
  g.fill oval (160,160,280,200);
  g.set color (color.red);
  g.draw string ("WELCOME TO AAC-B.Sc [C.S]", 200,400);
```

```
    add window listener (new mywindow Adapter ());
    Show ();
  }
  Public void action Performed (Action Event e)
  {
   repaint ();
  }
  Class my window Adapter extends window Adapter
  {
   Public void window closing (window Event e)
   {
     System.exit (0);
   }
  }
  }
  class shapes
  {
  Public static void main (String args [])
  {
  draw obj = new draw ("Drawing shapes");
  obj.set visible (true);
  obj.set Size (600, 600);
  }
}
```

OUTPUT:

```
D:\nJava>set path=c:\jdk1.5.0\bin
D:\nJava>javac Shapes.java
D:\nJava>java Shapes.
```


Drawing Shapes
WELCOME TO AAC - B.Sc[C.S]

Expt.No.:
Date :

Page No.: 49

RESULT:

Thus, the Java programming has been created successfully and the output are verified.

Expt.No.: 10
Date :

Page No.: 51

WRITE A PROGRAM TO CREATE A SEQUENTIAL FILE THAT COULD STORE DETAILS ABOUT FIVE PRODUCTS. DETAILS INCLUDE PRODUCT CODE, COST AND NUMBER OF ITEMS AVAILABLE AND ARE PROVIDED THROUGH THE KEYBOARD. COMPUTE AND PRINT THE TOTAL VALUE OF ALL THE FIVE PRODUCTS.

AIM:

To write a program to create a sequential file that could store details about five Products. Details Include Product code, cost and number of items Available and are provided through the keyboard. compute and print the total value of all the five products.

PROCEDURE:

Step 1: Start
Step 2: Declare required variables
Step 3: Define Data output Stream file and object.
Step 4: Read value from user.
Step 5: Write the values of variables to file and Close the file.
Step 6: Define DataInput stream file and object.
Step 7: React the values of stored in file Sequentially.
Step 8: Print the values read.
Step 9: Close the file.
Step 10: Stop.

PROGRAM:

```
import java.util.*;
import java.io.*;
import java.io.DataInputStream;
import java.io.DataOutputStream;
Public Class inven 1
{
  Static DataInputStream clin=new DataInputStream
                              (system.in);
  Static String TOkenizer st;
  Public static void main (string args[]) throws IO
                                  Exception.
  {
       int code, items, i;
       double cost;
       DataOutputStream dos=new DataOutputStream
                  (new FileOutputStream("invent.dat"));
      for(i=1;ic=5;i++)
      {
            System.out.println("Enter the code number of"+
                                "Product");
            code=Integer.ParseInt(din.readLine());
            System.out.println("Enter the Number of items");
            items=Integer.ParseInt(din.readLine());
            dos.writeInt(code);
            dos.writeInt(items);
            dos.writeDouble(cost);
}
```

```
dos.close();
DataInputStream dis = new DataInputStream(new FileInputStream("invent.dat"));
int codeNumber, totalItems;
double itemCost, totalCost=0, totalValue=0;
System.out.println("****** PRODECT DETAILS ******")
System.out.println("code Number:\t Item Cost:\t Total items:\t Total Cost");
for(i=1;i<=5;i++)
{
  codeNumber = dis.readInt();
  totalItems = dis.readInt();
  ItemCost = dis.readDouble();
  totalCost = totalItems * itemCost;
  totalValue += totalCost;
  System.out.println("\t"+codeNo+"\t\t"+itemCost+"\t\t"+totalItems+"\t\t"+totalCost+"\n");
  System.out.println(" ");
}
System.out.println("Total value of products is = "+totalValue);
}
}
```

OUTPUT:

```
D:\nJava>set path=C:\jdk1.5.0\bin
D:\nJava>javac inven.java
D:\nJava>java inven.
Enter Code Number of 1 Product
101
Enter Number of items
10
Enter Cost
5
Enter Code Number of 2 Product
102
Enter Number of items
20
Enter Cost
10

Enter code Number of 2 product
103
Enter Number of items
30
Enter Cost
15.
```

RESULT:

Thus, the above Java program has been created successfully and the Output are verified.