## Syllabus:

1. Write a program to find the Area of Square, Rectangle and Circle using MethodOverloading.
2. Write a program to sort the list of numbers using Command Line Arguments.
3. Write a program to multiply the given two matrices.
4. Write a program to design a class to represent a bank account. Include thefollowing:

Data Members: Name of the depositor, Account number, Type of account,and Balance amount in the account.

Methods: To assign initial values, To deposit an amount, To withdraw anamount after checking balance, and To display the name and balance.

1. Write a program that import the user defined package and access the Membervariable of classes that contained by Package.
2. Write a program to handle the Exception using try and multiple catch blocks.
3. Write a program to illustrate the use of multithreads.
4. Write a program to create student registration form using applet with Name, Address, Sex, Class, Email-id.
5. Write a program to draw the line, rectangle, oval, text using the graphics method.
6. 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.

## METHOD OVERLOADING

1.Write a program to find the Area of Square, Rectangle and Circle usingMethod Overloading.

## AIM:

To write a java program to find the Area of Square, Rectangle and Circle using

Method Overloading

### ALGORITHM

STEP 1: Start the program.

STEP 2: Create a class with main() method.

STEP 3: Overload the method to calculate the area of square, rectangle and

circle withinthe class.

STEP 4: Create object for the class and call the methods.

STEP 5:Stop the program.

### SOURCE CODE:

import java.io.\*; class area

{

void calculateArea(float x)

{

System.out.println("Area of the Square:"+x\*x+" sq units");

}

void calculateArea(float x,float y)

{

System.out.println("Area of the rectangle:"+x\*y+" sq units");

}

void calculateArea(double r)

{

double area=3.14\*r\*r;

System.out.println("Area of the circle:"+area+" sq units");

}

public static void main(String args[])

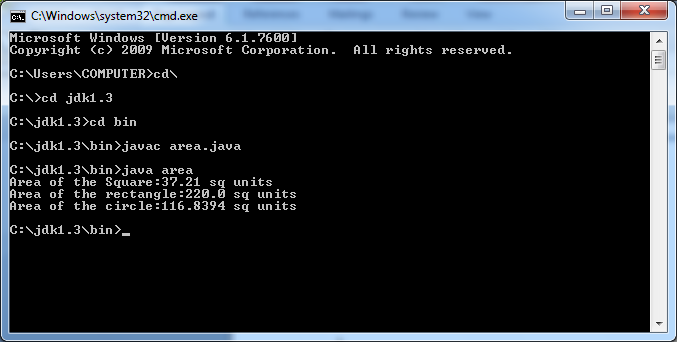
{

area obj=new area(); obj.calculateArea(6.1f); obj.calculateArea(10,22);

obj.calculateArea(6.1);

}

## OUTPUT:



## RESULT:

Thus the above program has been executed successfully.

**COMMAND LINE ARGUMENTS**

2.Write a program to sort the list of numbers using Command Line Arguments.

**AIM:**

To write a java program to sort the list of numbers using Command Line Arguments

**ALGORITHM:**

**STEP 1 :** Start the program.

**STEP 2 :** Create a class with main() method.

**STEP 3 :** Create an array and store the value passed from the command line

argument.

**STEP 4 :** Sort the elements of the array by comparing all the values with each

other.

**STEP 5 :** Display the sorted array element.

**STEP 6 :** Stop the program.

**SOURCE CODE**

public class Main

{

public static void main(String[] args)

{

int a[]=new int[20]; int j,temp,i;

for(i=0;i<args.length;i++)

{

a[i]=Integer.valueOf(args[i]);

}

System.out.println("Elements in the unsorted array are: ");

for(i=0;i<args.length;i++)

{

System.out.println(a[i]+"\t"); } for(i=0;i<args.length;i++)

{

for(j=0;j<args.length-i-1;j++)

{

if(a[j]>a[j+1])

{

temp=a[j]; a[j]=a[j+1]; a[j+1]=temp;

}

}

}

System.out.println("Sorted array elements are: ");

for(i=0;i<args.length;i++)

{

System.out.println(a[i]+"\t");

}

}

}

**OUTPUT:**

>Java Main 10 50 15 12

Elements in the unsorted array are: 10

50

15

12

Sorted array elements are: 10

12

15

50

**RESULT:**

Thus the above program has been executed successfully.

## MULTIPLICATION OF TWO MATRIX

### 3. Write a program to multiply the given two matrices.

### AIM:

To write a java program to multiply the given two matrices.

**ALGORITHM:**

STEP 1: Start the program.

STEP 2: Create a class with main() method. STEP 3: Read theinput for order of two matrices. STEP 4: Read the input for two matrices.

STEP 5: Multiply the two matrices and store in another matrix.

STEP 6:Write the elements of the third matrix. STEP 7: Stop the program.

### SOURCE CODE:

import java.io.\*;

import java.util.Scanner; public class Matrix

{

public static void main(String args[])

{

int m,n,p,q,sum=0,c,d,k;

Scanner in=new Scanner(System.in);

System.out.print("Enter Number of Rows and Colums of first Matrix:"); m=in.nextInt();

n=in.nextInt();

int first[][]=new int[m][n]; System.out.print("Enter first Matrix Elements:"); for(c=0;c<m;c++)

{

for(d=0;d<n;d++)

{

first[c][d]=in.nextInt();

}

}

System.out.print("Enter Number of Rows and Colums of second Matrix:"); p=in.nextInt();

q=in.nextInt(); if(n!=p)

{

System.out.print("Matrix of the entered order can't be Multiplied..!!");

}

else

{

int second[][]=new int[p][q];

int multiply[][]=new int[m][q]; System.out.print("Enter second Matric Elements:\n");

for(c=0;c<p;c++)

{

for(d=0;d<q;d++)

{

second[c][d]=in.nextInt();

}

}

System.out.print("Multiplying both Matrix..\n"); for(c=0;c<m;c++)

{

for(d=0;d<q;d++)

{

for(k=0;k<p;k++)

{

sum=sum+first[c][k]\*second[k][d];

}

multiply[c][d]=sum; sum=0;

}

}

System.out.print("Multiplication Successfull performed..!!\n"); System.out.print("Now the Matrix Multiplication Result is:\n"); for(c=0;c<m;c++)

{

for(d=0;d<q;d++)

{

System.out.print(multiply[c][d]+"\t");

}

System.out.print("\n");

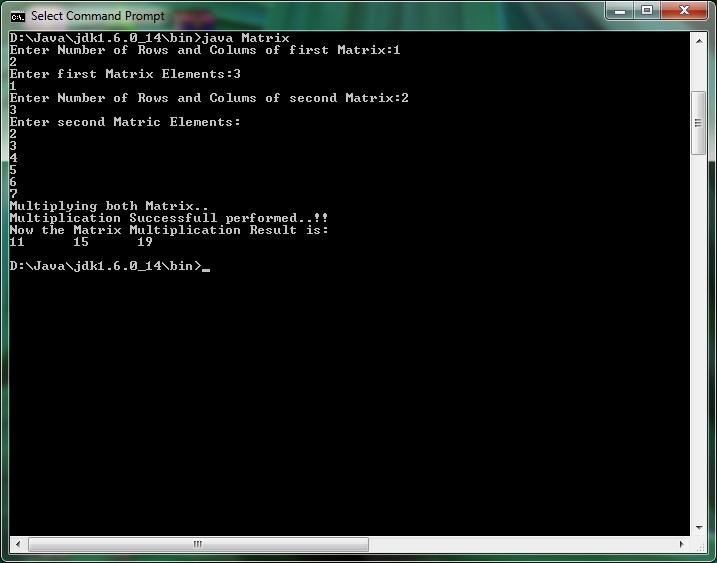
}

}

}

}

**OUTPUT:**



## RESULT:

Thus the above program has been executed successfully.

## BANK DETAILS

4. Write a program to design a class to represent a bank account. Include the following: Data Members: Name of the depositor, Account number, Type of account, and Balance amount in theaccount. Methods: To assign initial values, To deposit an amount, To withdraw an amount after checking balance, and To display the name and balance.

### AIM:

To write a java program for bank operation

**ALGORITHM:**

STEP 1: Start the program.

STEP 2: Create a class with data member such as depositor name, account number, type of accountand balance amount.

STEP 3: Create Methods such as assign initial values, deposit, withdraw, balance checkingand display the details.

STEP 4: Create an object for the class in main() method.

STEP 5: Call the methods. STEP 6: Stop the program.

### SOURCE CODE:

import java.io.\*; class Account {

private String name; double balance = 0.0;

Account(String name, double balance) { this.name = name;

this.balance = balance;

}

public void deposit(double amount)

{

balance = balance + amount;

}

public void withdraw(double amount) { balance = balance - amount;

}

public void display()

{

System.out.println("Name of the depositor:"+name); System.out.println("Balance Amount:"+balance);

}

public double getBalance(double balance) { return balance;

}

}

public class Bank {

public static void main(String[] args) {

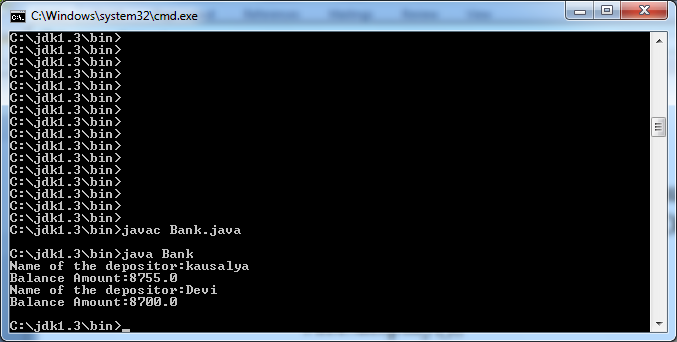
Account Acc = new Account("kausalya",8555); Account Acc1 = new Account("Devi",9000); Acc.deposit(200);

Acc1.withdraw(300); Acc.display(); Acc1.display();

}

}

**OUTPUT:**



## RESULT

Thus the above program has been executed successfully

## PACKAGE

### 5. Write a program that import the user defined package and access the Membervariable of classes that contained by Package.

**AIM:**

To write a java program that import the user defined package and access the Member variable of classes that contained by Package.

### ALGORITHM:

STEP 1: Start the program.

STEP 2: Create the user-defined package.

STEP 3: Create the stack class with member variables. STEP 4: Create the methods for stack operations.

STEP 5: Create a program outside the package. STEP 5: Import the user-defined package.

STEP 6: Create a class with main() method.

STEP 7: Create an object to access the methods inside the user-defined package for stack operation.

STEP 8: Stop the program.

### SOURCE CODE:

/\*Source code of package p1 under the directory C:\jdk1.6.0\_26\bin>p1\edit Student.java \*/ package p1;

public class Student

{

int regno; String name;

public void getdata(int r,String s)

{

regno=r; name=s;

}

public void putdata()

{

System.out.println("regno = " +regno); System.out.println("name = " + name);

}

}

/\* Source code of the main function under C:\jdk1.6.0\_26\bin>edit StudentTest.java \*/ import p1.\*;

class StudentTest

{

public static void main(String arg[])

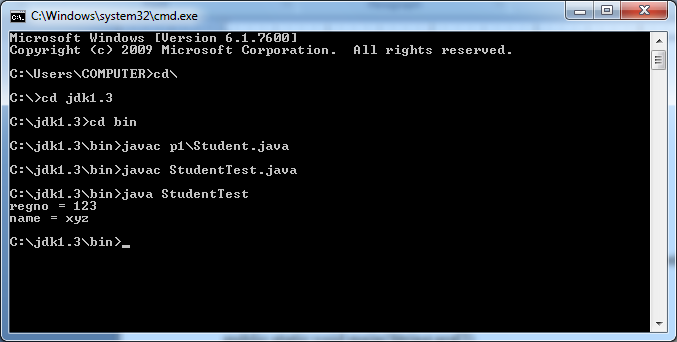
{

student s=new student(); s.getdata(123,"xyz"); s.putdata();

}

}

**OUTPUT:**



**RESULT:**

Thus the above program has been executed successfully.

# EXCEPTION HANDLING

6.Write a program to handle the Exception using try and multiple catch blocks.

### AIM:

To write a java program to handle the exception using try and multiple catch blocks.

**ALGORITHM:**

STEP 1: Start the program.

STEP 2: Create a class with main() method.

STEP 3: Read an integer value as input.

STEP 4: Create a try block and write the arithmetic operation.

STEP 5: Create multiple catch block to display the exception.

STEP 6: Stop the program.

### SOURCE CODE:

**import** java.io.\*;

**class** Myexception

{

p**ublic static void** main(String args[])

{

### try

{

**int** a[]=**new int**[5]; a[5]=30/0;

}

**catch**(ArithmeticException e)

{

System.***out***.println("Arithmetic Exception occurs");

}

**catch**(ArrayIndexOutOfBoundsException e)

{

System.***out***.println("ArrayIndexOutOfBounds Exception occurs");

}

**catch**(Exception e)

{

System.***out***.println("Parent Exception occurs");

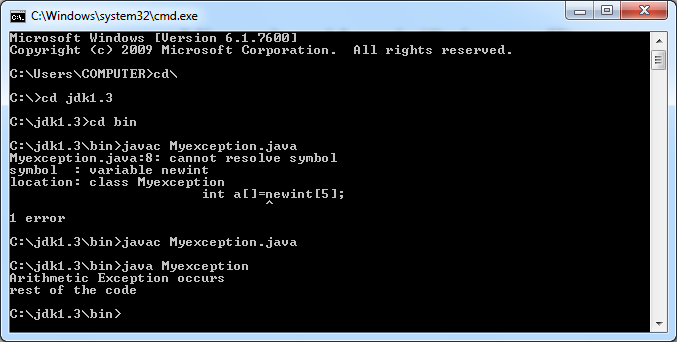
}

System.***out***.println("rest of the code");

}

}

**OUTPUT:**



**RESULT:**

Thus the above program has been executed successfully.

# MULTITHREADING

7. Write a program to illustrate the use of multithreads.

### AIM:

To write a java program to illustrate the use of multithreads.

### ALGORITHM

STEP 1: Start the program.

STEP 2: Create a class that inherits the Thread class.

STEP 3: Override the run() method and execute the loop to display some value. STEP 4:Create a class with main() method.

STEP 5: Create two objects for the thread class. STEP 6: Start thethread of the two objects.

STEP 7: Stop the program.

### SOURCE CODE:

import java.io.\*;

class Thread3 extends Thread{

public void run(){ for(int i=1;i<5;i++){ try{

Thread.sleep(500);

}catch(InterruptedException e){ System.out.println(e);

}

System.out.println(i);

}

}

public static void main(String args[]){

Thread3 t1=new Thread3(); Thread3 t2=new Thread3();

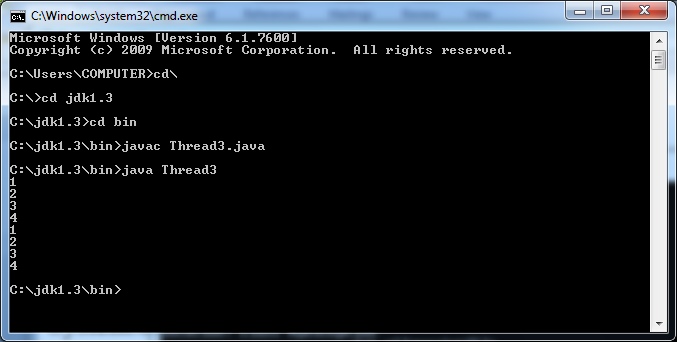
t1.run();

t2.run();

}

}

**OUTPUT:**



**RESULT:**

Thus the above program has been executed successfully.

**STUDENT REGISTRATION FORM USING APPLET**

8. Write a program to create student registration form using applet with Name,Address, Sex, Class, Email id.

### AIM:

To write an applet program for student registration form.

## ALGORITHM

STEP 1: Start the program.

STEP 2: Create a class that extends the Applet class.

STEP 3: Create an object for TextField, Button, Checkbox and Label classes in the init().

STEP 4: Override the actionPerformed() method to check and display the events.

## SOURCE CODE:

### Tes.java

import java.awt.\*;

public class Tes extends java.applet.Applet

{

public void init()

{

setLayout(new FlowLayout(FlowLayout.LEFT)); add(new Label("Name :"));

add(new TextField(10)); add(new Label("Address :")); add(new TextField(10)); add(new Label("Birthday :")); add(new TextField(10)); add(new Label("Gender :")); Choice gender = new Choice(); gender.addItem("Man"); gender.addItem("Woman"); Component add = add(gender); add(new Label("Job :"));

CheckboxGroup job = new CheckboxGroup(); add(new Checkbox("Student", job, false)); add(new Checkbox("Teacher", job, false)); add(new Button("Register"));

add(new Button("Exit"));

}

}

### Tes.html

<html>

<head><title>Register</title></head>

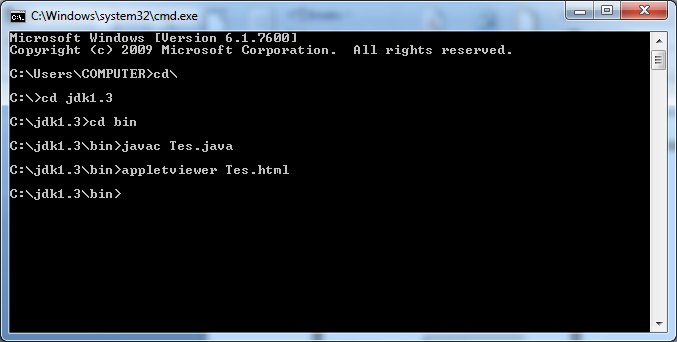
<body>

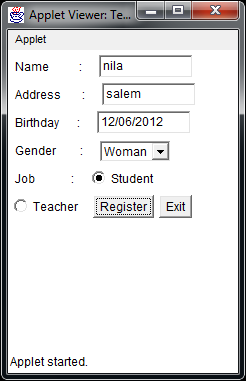
<applet code="Tes.class" width=230 height=300></applet>

</body>

</html>

**OUTPUT:**





## RESULT:

Thus the above program has been executed successfully.

## GRAPHICS METHOD

9.Write a program to draw the line, rectangle, oval, text using the graphics method.

### AIM:

To write an applet program to display line, rectangle, oval and text using graphics method.

**ALGORITHM:**

STEP 1: Start the program.

STEP 2: Create a class that extends Applet class.

STEP 3: Override the pain() method.

STEP 4: Call the method to display the line, rectangle, oval and text.STEP 4: Stop the program.

### SOURCE CODE:

import java.awt.\*; import java.applet.\*;

/\*

<applet code="Drawing.class"height="500"width="500">

</applet>

\*/

public class Drawing extends Applet

{

public void paint(Graphics g)

{

g.drawLine(10,10,50,50); g.drawRect(10,60,40,30); g.setColor(Color.green); g.fillRect(160,10,30,80); g.drawOval(220,20,200,120);

g.setColor(Color.red);

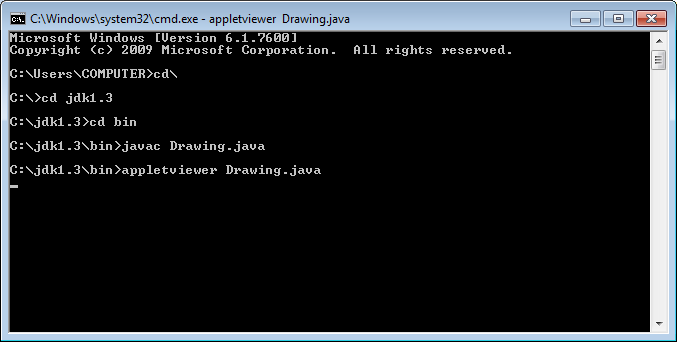
g.fillOval(220,170,150,100);

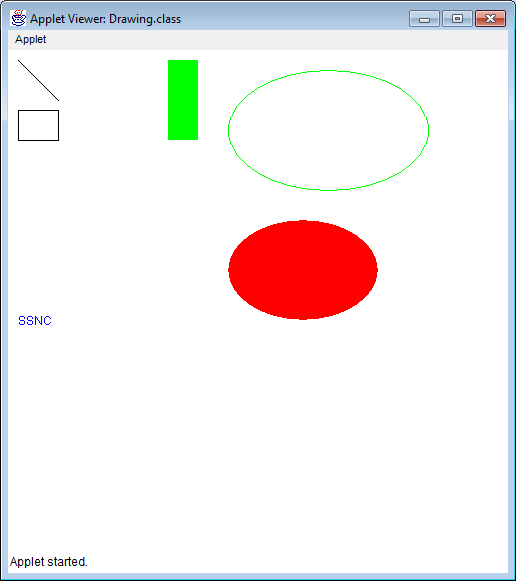
g.setColor(Color.blue); g.drawString("SSNC",10,275);

}

}

**OUTPUT:**



****

### RESULT:

Thus the above program has been executed successfully.

# SEQUENTIAL FILE

**10. 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 java program for file operation.

## ALGORITHM:

STEP 1: Start the program.

STEP 2: Create a class for getting product information.

STEP 3: Declare data member such as product code, cost and count. STEP 4: Write the method to get input for the product information.

STEP 5: Write methods to return the product information. STEP 6: Create a class with main() function.

STEP 7: Create a file to store product information. STEP 8: Create an object for product information.

STEP 9: Call the methods of an object to get product information and store it in the file. STEP 10: Calculate the product value and display.

## SOURCE CODE:

import java.io.\*; import java.util.\*; public class Products1

{

public static void main(String arg[])

{

File file=new File("prod1"); int j=0;

try

{

FileOutputStream fos=new FileOutputStream(file);

DataOutputStream dos=new DataOutputStream(fos); DataInputStream dis=new DataInputStream(System.in); StringTokenizer st;

dos.writeBytes("Product Code"+"\t"+"Product Cost"+"\t"+"No. of items"+"\t"+"Total"+"\n");

for(int i=0;i<5;i++)

{

System.out.print("Please enter the product code:"); st=new StringTokenizer(dis.readLine());

String pnum=new String(st.nextToken()); System.out.print("Please enter the product cost:"); st=new StringTokenizer(dis.readLine());

double cost=new Double(st.nextToken()).doubleValue(); System.out.print("Please enter the product quantity:"); st=new StringTokenizer(dis.readLine());

int qty=new Integer(st.nextToken()).intValue(); double tot=cost\*qty;

String strcost=Double.toString(cost); String strqty=Integer.toString(qty); String strtot=Double.toString(tot);

dos.writeBytes(pnum+"\t\t"); dos.writeBytes(strcost+"\t\t"); dos.writeBytes(strqty+"\t\t"); dos.writeBytes(strtot+"\n");

}

dos.close();

}

catch(IOException e)

{

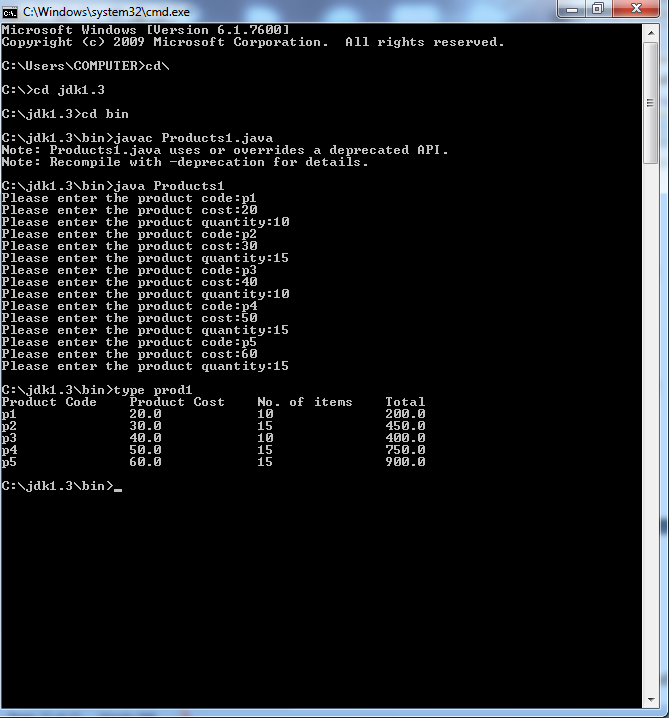
System.out.println("IO error:"+e);

}

}

}

### OUTPUT:



**RESULT:**

Thus the above program has been executed successfully.