AREA AND PERIMETER OF CIRCLE

CODE:-

import java.io.\*;

class circle

{ publicstaticvoid main(String arg[])

 { int radius;

try

 { InputStreamReader In=new InputStreamReader(System.in);

 BufferedReader br= new BufferedReader(In);

 System.out.print("Enter the radius of circle = ");

 String str= br.readLine();

radius= Integer.parseInt(str);

double area,perimeter;

area=3.14\*radius\*radius;

perimeter=2\*3.14\*radius;

System.out.println("Area of circle = " + area);

System.out.println("Perimeter of circle = "+perimeter);

}

catch(Exception e)

 { System.out.println("Error : " + e);

}

}

}

OUTPUT:-

Enter the radius of circle=3

Area of circle=28.26

Perimeter of circle=18.84

SUBSTRING REMOVAL

CODE:-

import java.io.\*;

class subrem

 { publicstaticvoid main(String arg[])

 { StringBuffer str;

String substr,temp;

int strlen,i=0,ctr=0;

int substrlen;

DataInputStream din=new DataInputStream(System.in);

try

 { System.out.println("Enter the string : ");

 str=new StringBuffer(din.readLine());

System.out.println("Enter the substring to be removed");

substr=din.readLine();

strlen=str.length();

substrlen=substr.length();

strlen=strlen-substrlen;

while(i<=strlen)

 { temp=str.substring(i,i+substrlen);

if(temp.equals(substr))

 { str.delete(i,i+substrlen);

strlen=strlen-substrlen;

 ctr++;

}

else

 { i=i+1;

}

}

System.out.println(str);

 System.out.println("number of substring removed"+ ctr);

 }

catch(IOException e)

 { System.out.println(e);

}

catch(StringIndexOutOfBoundsException e)

 {System.out.println(e);

 }

}

 }

OUTPUT:-

Enter the string:

Welcome to the game

Enter the substring to be removed:

Me

Welco to the ga

Number of substring removed:2

 METHOD OVERLOADING

CODE:-

class overload

 { int p;

Void volume(int a)

 { p=a\*a\*a;

System.out.println("Volume of cube:" + p);}

void volume(int a,int b,int c)

 { p=a\*b\*c;

System.out.println("Volume of cuboid" + p);

 }

Public static void main(String arg[])

 {

 overload b=new overload();

b.volume(5);

 b.volume(5,10,5);

 }

}

OUTPUT:-

Volume of cube:125

Volume of cuboid:250

 MULTI-LEVEL INHERITANCE

CODE:-

import java.io.\*;

class multilvl

 {

int sid;

 String sname,branch;

char year;

 multilvl(int id,String sn,String b,char y)

 {

 sid=id;

 sname=sn;

 branch=b;

 year=y;

 }

void display()

 {

 String yrdetail=" ";

if(year=='f')

 yrdetail="First Year";

elseif (year=='s')

 yrdetail="Second Year";

elseif (year=='t')

 yrdetail="Third Year";

 System.out.println("\n\t\t\t The student details :");

 System.out.println("Student id : " + sid);

 System.out.println("Name : " + sname);

 System.out.println("Branch : " + branch);

 System.out.println("Year : " + yrdetail);

 }

}

class mark extends multilvl

 {

 String category;

int mark1,mark2,mark3;

 mark(int id, String sn,String b,char y,String cat,int m1,int m2,int m3)

 {

super(id,sn,b,y);

 category=cat;

 mark1=m1;

 mark2=m2;

 mark3=m3;

 }

void display()

 {

super.display();

 System.out.println("Category : " + category);

 System.out.println("Mark1 : " + mark1);

 System.out.println("Mark2 : " + mark2);

 System.out.println("Mark3 : " + mark3);

 }

 }

class report extends mark

 {

double total,avg;

 String grade;

 report(int id,String sn,String b,char y,String cat,int m1,int m2,int m3)

 {

super(id,sn,b,y,cat,m1,m2,m3);

 }

void tot()

 {

 total=mark1+mark2+mark3;

 avg=total/3.0;

if((mark1>=40) && (mark2>=40) && (mark3>=40))

 {

if(avg>=80)

 grade="Distinction";

elseif((avg>=60)&&(avg<80))

 grade="First class";

else

 grade="Second class";

 }

else

 grade="Fail";

 }

void display()

 {

 tot();

super.display();

 System.out.println("\t\t Total : " + total);

 System.out.println("\t\t Average : " + avg);

 System.out.println("\t\t Grade : " + grade);

 }

 }

class inheritance

 {

publicstaticvoid main(String arg[]) throws IOException

 {

int id,m1,m2,m3;

 String name,br,cat;

char yr,ch;

 DataInputStream din=new DataInputStream(System.in);

do

 {

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

 id=Integer.parseInt(din.readLine());

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

 name=din.readLine();

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

 br=din.readLine();

 System.out.println("Enter year (f/s/t) : ");

 yr=(din.readLine()).charAt(0);

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

 cat=din.readLine();

 System.out.println("Enter mark1 : ");

 m1=Integer.parseInt(din.readLine());

 System.out.println("Enter mark2 : ");

 m2=Integer.parseInt(din.readLine());

 System.out.println("Enter mark3 : ");

 m3=Integer.parseInt(din.readLine());

 report r1=new report(id,name,br,yr,cat,m1,m2,m3);

 r1.display();

 System.out.println("\n\n Do you want to continue? (Y/N) : ");

 ch=(din.readLine()).charAt(0);

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

 }

}

OUTPUT:-

Enter the id:100

Enter name:AAA

Enter branch:CS

Enter year(f/s/t):s

Enter category:III sem

Enter mark1:90

Enter mark2:90

Enter mark3:90

Student detail

Student id:100

Student name:AAA

Branch:CS

Year:second year

Category: III sem

Mark1:90

Mark2:90

Mark3:90

Total:270

Average:90

Grade=Distinction

Do you want to continue? (y/n):n

 INTERFACE

CODE:-

interface shapes

{publicdouble calculatearea();

}

class rectangle implements shapes

{int l,b;

rectangle(int l,int b)

{this.l=l;

this.b=b;

 }

publicdouble calculatearea()

 {

return(l\*b);

 }

}

class triangle implements shapes

{

int b,h;

 triangle(int b,int h)

 {

this.b=b;

this.h=h;

 }

publicdouble calculatearea()

 {

return(0.5\*b\*h);

 }

}

class shapesEx

{

publicstaticvoid main(String arg[])

 {

 rectangle r1=new rectangle(2,7);

 triangle t1=new triangle(4,16);

double area;

 area=r1.calculatearea();

 System.out.println("Area of rectangle = " + area);

 area=t1.calculatearea();

 System.out.println("Area of triangle = " + area);

 }

}

OUTPUT:-

Area of rectangle=14

Area of triangle=32

 ABSTRACT CLASS

CODE:-

import java.io.\*;

abstract class shapes

{abstract void area(int l,int b);

}

class rectangle extends shapes

{void area(int l,int b)

 {System.out.println("Area of rectangle = " + (l\*b));

 }

}

class triangle extends shapes

{void area(int l,int b)

 {System.out.println("Area of triangle = " + (0.5\*l\*b));

 }

}

class abstractEx

{public static void main(String arg[])

 {

int l,b;

 shapes s1;

 System.out.println("Enter the length and breadth:");

try

 {

DataInputStream din=new DataInputStream(System.in);

 l=Integer.parseInt(din.readLine());

 b=Integer.parseInt(din.readLine());

 s1=new rectangle();

 s1.area(l,b);

 s1=new triangle();

 s1.area(l,b);

 }

catch(Exception e)

 {

 System.out.println(e);

 }

 }

}

OUTPUT:-

Enter the length and breadth

4

5

Area of rectangle=20

Area of triangle=10

PREDEFINED EXCEPTION

CODE:-

import java.io.\*;

class predefinedException

{publicstaticvoid main(String arg[]) throws IOException

 {double a[]=newdouble[5];

double b[]=newdouble[5];

double c[]=newdouble[5];

int limit;

 DataInputStream din=new DataInputStream(System.in);

try

 {System.out.print("Enter the limit : ");

limit=Integer.parseInt(din.readLine());

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

 {System.out.print("Enter a["+i+"] : ");

 a[i]=Double.parseDouble(din.readLine());

 System.out.print("Enter b["+i+"] : ");

 b[i]=Double.parseDouble(din.readLine());

 c[i]=a[i]/b[i];

 System.out.println("c["+i+"] : " + c[i]);

 }

 }

catch(ArrayIndexOutOfBoundsException e)

 {System.out.print("Exception thrown" + e);

 }

catch(IOException e)

 {System.out.print("Exception thrown" + e);

 }

catch(NumberFormatException n)

 {

 System.out.print("Exception thrown" + n);

 }

catch(ArithmeticException n)

 {

 System.out.print("Exception thrown" + n);

 }

 }

}

OUTPUT:-

Enter the limit:1

Enter a[0]:2

Enter b[0]:4

C[0]=.5

 USERDEFINED EXCEPTION

OUTPUT:-

import java.io.\*;

class MarksOutOfBoundsException extends Exception

 {

int mark;

 MarksOutOfBoundsException(int mark)

 {

this.mark=mark;

 }

public String toString()

 {

return("Mark should be greater than 0 and less than 100 and your input is " + mark);

 }

}

class demo

 {

publicstaticvoid main(String arg[])

 {

 BufferedReader br= new BufferedReader(new InputStreamReader(System.in));

int mark;

try

 {

 String str= br.readLine();

 mark=Integer.parseInt(str);

if(mark<16)

thrownew MarksOutOfBoundsException(mark);

 }

catch(IOException e)

 {

 System.out.println(e);

 }

catch(MarksOutOfBoundsException e)

 {

 System.out.println(e);

 }

 }

 }

OUTPUT:-

Enter the marks:45

 RUNNABLE

CODE:-

class EvenThread implements Runnable

{ int a[]=newint[10];

 String name;

 Thread t;

 EvenThread(String name,int a[])

 { t=new Thread(this,name);

this.a=a;

 t.start();

 }

publicvoid run()

 { try

 { for(int i=0;i<a.length;i++)

 { if(a[i]%2==0)

 System.out.println("Even no" + i + "is" + a[i]);

 }

 }

catch(Exception e)

 { System.out.println(e);

 }

 }

}

class OddThread implements Runnable

{ int a[]=newint[10];

 String name;

 Thread t;

 OddThread(String name,int a[])

 { t=new Thread(this,name);

this.a=a;

 t.start();

 }

publicvoid run()

 { try

 { for(int i=0;i<a.length;i++)

 { if(a[i]%2!=0)

 System.out.println("Odd no" + i + "is" + a[i]);

 }

 }

catch(Exception e)

 { System.out.println(e);

 }

 }

}

class ThreadRunnable

{ publicstaticvoid main(String arg[])

 { int a[]={10,57,78,32,43};

 EvenThread e=new EvenThread("Even",a);

 OddThread o=new OddThread("Odd",a);

 }

}

OUTPUT:-

Even1=10

Odd1=57

Even2=78

Odd2=43

Even3=32

STRING MANUPULATION

CODE:-

import java.io.\*;

class strmanp

{

publicstaticvoid main(String arg[]) throws IOException

 {

 String str;

int vowels=0,digits=0,blanks=0,cons=0,spchar=0;

char ch;

 BufferedReader br= new BufferedReader(new InputStreamReader(System.in));

 System.out.print("Enter a string : ");

 str=br.readLine();

for(int i=0;i<str.length();i++)

 {

 ch=str.charAt(i);

if(ch=='a' || ch=='A' || ch=='e' || ch=='E' || ch=='i' || ch=='I' || ch=='o' || ch=='O' || ch=='u' || ch=='U')

 vowels++;

elseif (Character.isLetter(ch))

 cons++;

elseif (Character.isDigit(ch))

 digits++;

elseif (Character.isWhitespace(ch))

 blanks++;

else

 spchar++;

 }

 System.out.println("Vowels : " + vowels);

 System.out.println("Constants : " + cons);

 System.out.println("Digits : " + digits);

 System.out.println("Blanks : " + blanks);

 System.out.println("Special Character : " + spchar);

 }

}

OUTPUT:-

Enter the string:ake 23@

Vowels:2

Consonants:1

Digits:2

Whitespaces:2

Special characters:1

 FILE COPY

CODE:-

import java.io.\*;

class filecopy

 {

publicstaticvoid main(String arg[]) throws IOException

 {

int c;

 String a,b;

 DataInputStream din=new DataInputStream(System.in);

 FileInputStream f1;

 FileOutputStream f2;

 System.out.println("Enter the name of first file : ");

try

 {

 a=din.readLine();

 System.out.println("Enter the name of second file : ");

 b=din.readLine();

 f1=new FileInputStream(a);

 f2=new FileOutputStream(b);

while(true)

 {

 c=f1.read();

if(c==-1)

 {

break;

 }

 f2.write(c);

 }

 f1.close();

 f2.close();

 }

catch(FileNotFoundException e)

 {

 System.out.println(e);

 }

}

}

Output:-

Enter the name of first file:original.txt

Enter the name of second file:copy

LEAP YEAR

CODE:-

import java.io.\*;

publicclass leapyear

{ publicstaticvoid main(String arg[])

 { int year;

 DataInputStream din=new DataInputStream(System.in);

 try

 { System.out.println(“Enter the year”);

 year=Integer.parseInt(din.readLine());

if((year%400==0)||((year%4==0)&&(year%100!=0)))

 { System.out.println(year + " is a leap year");

 }

else

 {

 System.out.println(year + " is not a leap year");

 }

 }

 catch(Exception e)

 { System.out.println(e);

 }

}

OUTPUT:-

Enter the year:2004

2004 is a leap year

 FIBONACCI SERIES

CODE:-

import java.io.\*;

class fibonacci

 { publicstaticvoid main(String arg[])

 { int x1=-1, x2=1, x3=0, n;

 DataInputStream din=new DataInputStream(System.in);

 try

 { System.out.println(“Enter the number of terms”);

 n=Integer.parseInt(din.readLine());

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

 { x3=x1+x2;

 System.out.print(x3 + " ");

 x1=x2;

 x2=x3;

 }

 }

 Catch(Exception e)

 { System.out.println(e);

 }

 }

}

OUTPUT:-

Enter the number of terms:10

0 1 1 2 3 5 8 13 21 34

 VOLUME OF BOX

CODE:-

import java.io.\*;

class box

 {

int a,l,b,h;

 box(int a)

 {

this.a=a;

 }

 box(int l,int b,int h)

 {

this.l=l;

this.b=b;

this.h=h;

 }

void calc1()

 {

 System.out.println("Volume of cube : " + a\*a\*a);

 }

void calc2()

 {

 System.out.println("Volume of cuboid : " + l\*b\*h);

 }

 }

class boxvol

 {

publicstaticvoid main(String arg[])

 {

int a,l,b,h;

try

 { DataInputStream br=new DataInputStream(System.in);

 System.out.print("Enter the value of side of cube : ");

 String str=br.readLine();

 a=Integer.parseInt(str);

 box b1=new box(a);

 b1.calc1();

 System.out.print("Enter the value of length : ");

 String str1=br.readLine();

 l=Integer.parseInt(str1);

 System.out.print("Enter the value of breadth : ");

 String str2=br.readLine();

 b=Integer.parseInt(str2);

 System.out.print("Enter the value of height : ");

 String str3=br.readLine();

 h=Integer.parseInt(str3);

 box b2=new box(l,b,h);

 b2.calc2();

 }

catch(Exception e)

 {

 System.out.println(e);

 }

 }

 }

OUTPUT:-

Enter the value of side of cube:2

Volume of cube:8

Enter the value of length:2

Enter the value of breadth:3

Enter the value of height:2

Volume of cuboid:12

 PRIME NUMBER

CODE:-

publicclass prime {

publicstaticvoid main(String[] args)

 { int n,flag=1;

 DataInputStream din=new DataInputStream(System.in);

 try

 { System.out.println(“Enter the number”);

 n=Integer.parseInt(din.readLine());

for (int i=2;i<n;i++)

 { if(n%i==0)

{ flag=0;

break;

}

 }

if(flag==0)

 {

 System.out.println(n + " is not a prime number");

}

else

{

 System.out.println(n + " is a prime number");

}

 }

 catch(Exception e)

 { System.out.println(e);

}

}

 }

OUTPUT:-

Enter the number:5

5 is a prime number

 SORTING USING RANDOM

CODE:-

import java.util.Random;

import java.io.\*;

class sorting

 {

publicstaticvoid main(String arg[]) throws Exception

 {

int temp,i,j,n;

 DataInputStream din=new DataInputStream(System.in);

 System.out.println(“Enter the number of terms:”);

 n=Integer.parseInt(din.readLine());

int a[]=newint[n];

 Random rg=new Random();

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

 {

 a[i]=rg.nextInt(1000);

 System.out.println(a[i]);

 }

 System.out.println("After sorting : ");

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

 {

for(j=i+1;j<n;j++)

 {

if(a[i]>a[j])

 {

 temp=a[i];

 a[i]=a[j];

 a[j]=temp;

 }

 }

 }

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

 {

 System.out.println(a[i]);

 }

 }

}

 OUTPUT:-

Enter the number of terms:5

11 54 987 2 542

After sorting:

2 11 54 542 987

 MATRIX ADDITION

CODE:-

import java.io.\*;

class rec45

 {

publicstaticvoid main(String arg[])

 {

double a[][]=newdouble[5][5];

double b[][]=newdouble[5][5];

double c[][]=newdouble[5][5];

int l1,i,j,l2;

try

 {

 System.out.println("Enter number of rows : ");

 DataInputStream din=new DataInputStream(System.in);

 String str=din.readLine();

 l1=Integer.parseInt(str);

 System.out.println("Enter number of columns : ");

 l2=Integer.parseInt(str);

for(i=0;i<l1;i++)

 {

for(j=0;j<l2;j++)

 {

 System.out.println("Enter a["+i+"]["+j+"] : ");

 a[i][j]=Double.parseDouble(din.readLine());

 }

 }

for(i=0;i<l1;i++)

 {

for(j=0;j<l2;j++)

 {

 System.out.println("Enter b["+i+"]["+j+"] : ");

 b[i][j]=Double.parseDouble(din.readLine());

 }

 }

for(i=0;i<l1;i++)

 {

for(j=0;j<l2;j++)

 {

 c[i][j]=a[i][j]+b[i][j];

 }

 }

 System.out.println(“After addition:”);

for(i=0;i<l1;i++)

 {

for(j=0;j<l2;j++)

 {

 System.out.println(c[i][j] + “ “);

 }

 System.out.println();

 }

 }

catch(IOException e)

 {

 System.out.println(e);

 }

catch(ArrayIndexOutOfBoundsException e)

 {

 System.out.println(e);

 }

catch(ArithmeticException e)

 {

 System.out.println(e);

 }

catch(NumberFormatException e)

 {

 System.out.println(e);

 }

 }

}

OUTPUT:-

Enter number of rows:2

Enter number of columns:2

Enter a[0][0]:1

Enter a[0][1]:2

Enter a[1][0]:3

Enter a[1][1]:4

Enter b[0][0]:1

Enter b[0][1]:2

Enter b[1][0]:3

Enter b[1][1]:4

After addition:

2 4

6 8

 FACTORIAL

CODE:-

import java.io.\*;

publicclass FACT

 {

publicstaticvoid main(String arg[])

 {

int n;

try

 { DataInputStream br=new DataInputStream(System.in);

 System.out.println("Enter the term : ");

 String str=br.readLine();

 n=Integer.parseInt(str);

int i,f=1;

for(i=2;i<=n;i++)

 {

 f=f\*i;

 }

 System.out.println("Factorial = " + f);

 }

catch(Exception e)

 {

 System.out.println(e);

 }

}

}

OUTPUT:-

Enter the term:4

Factorial=24

OVERRIDING METHODS

CODE:-

class A

{

 int i,j;

 A(int a,int b)

 {

 i=a;

 j=b;

 }

 void show()

 {

 System.out.println("i and j: "+i+" "+j);

 }

}

class B extends A

{

 int k;

 B(int a,int b,int c)

 {

 super(a,b);

 k=c;

 }

 void show()

 {

 System.out.println("k: "+k);

 }

}

public class override

{

 public static void main(String args[])

 {

 B ob=new B(1,2,3);

 ob.show();

 }

}

OUTPUT:

k: 3

INTERPROCESS COMMUNICATION

CODE:-

class Q {

 int n;

 boolean flag = false;

 synchronized int remove()

{

 if(!flag)

 try {

 wait();

 } catch(InterruptedException e) {

 System.out.println("InterruptedException caught");

 }

 System.out.println("Item Removed: " + n);

 flag = false;

 notify();

 return n;

 }

 synchronized void add(int n)

 {

 if(flag)

 try {

 wait();

 } catch(InterruptedException e) {

 System.out.println("InterruptedException caught");

 }

 this.n = n;

 flag = true;

 System.out.println("Item Inserted: " + n);

 notify();

 }

}

class Producer implements Runnable {

 Q q;

 Producer(Q q) {

 this.q = q;

 new Thread(this, "Producer").start();

 }

 public void run() {

 int i = 1;

 while(true ) {

 q.add(i++);

 }

 }

}

class Consumer implements Runnable {

 Q q;

 Consumer(Q q) {

 this.q = q;

 new Thread(this, "Consumer").start();

 }

 public void run() {

 while(true) {

 q.remove();

 }

 }

}

class IPC {

 public static void main(String args[]) {

 System.out.println("Press Control-C to stop.");

 Q q = new Q();

 new Producer(q);

 new Consumer(q);

 }

}

OUTPUT:-

Press Control-C to stop.

Item Inserted: 1

Item Removed: 1

Item Inserted: 2

Item Removed: 2

Item Inserted: 3

Item Removed: 3

Item Inserted: 4

Item Removed: 4

Item Inserted: 5

Item Removed: 5

Item Inserted: 6

Item Removed: 6

THREAD SYNCHRONIZATION

CODE:-

class Callme {

 void call(String msg) {

 System.out.print("[" + msg);

 try {

 Thread.sleep(1000);

 } catch (InterruptedException e) {

 System.out.println("Interrupted");

 }

 System.out.println("]");}

}

class Caller implements Runnable {

 String msg;

 Callme target;

 Thread t;

 public Caller(Callme targ, String s) {

 target = targ;

 msg = s;

 t = new Thread(this);

 t.start();

 }

public void run() {

 synchronized(target)

{ target.call(msg);}

 }

}

class Synch {

 public static void main(String args[]) {

 Callme target = new Callme();

 Caller ob1 = new Caller(target, "I");

 Caller ob2 = new Caller(target, "Like");

 Caller ob3 = new Caller(target, "Java");

try {

 ob1.t.join();

 ob2.t.join();

 ob3.t.join();

 } catch(InterruptedException e) {

 System.out.println("Interrupted");}

 }

}

OUTPUT

[I]

[Like]

[Java]

UTILITY PROGRAM

CODE:-

import java.io.\*;

import java.util.\*;

class calenderdemo

{

 public static void main(String args[]) throws IOException

 {

 String month[]={"Jan","Feb","March","April","May","June","July","Aug","Sep","Oct","Nov","Dec"};

 int year;

 GregorianCalendar gc = new GregorianCalendar();

 System.out.print("Date: ");

 System.out.print(month[gc.get(Calendar.MONTH)]);

 System.out.print(" "+gc.get(Calendar.DATE)+" ");

 System.out.print(year=gc.get(Calendar.YEAR));

 System.out.print("TIME: ");

 System.out.print(gc.get(Calendar.HOUR)+":");

 System.out.print(gc.get(Calendar.SECOND));

 if(gc.isLeapYear(year))

 {

 System.out.println("The current year is leap year");

 }

 else

 {

 System.out.println("The current year is not leap year");

 }

 }

}

OUTPUT

Date: Oct 8 2013TIME: 11:24The current year is not leap year

 APPLET PROGRAMS

 BUTTONS

CODE:-

import java.awt.\*;

import java.applet.\*;

publicclass applet4 extends Applet

{

Button okButton1,okButton2,okButton3,okButton4,okButton5;

publicvoid init()

 {

 setLayout(new BorderLayout());

 okButton1=new Button("Centered Button");

 okButton2=new Button("Cold North");

 okButton3=new Button("Go West");

 okButton4=new Button("At East");

 okButton5=new Button("Hot South");

 add(okButton1,"Center");

 add(okButton2,"North");

 add(okButton3,"West");

 add(okButton4,"East");

 add(okButton5,"South");

 }

}



MOUSE LISTENER

CODE:-

import java.awt.\*;

import java.applet.\*;

import java.awt.event.\*;

publicclass applet10 extends Applet implements MouseListener

{

int xpos,ypos;

int rect1xco,rect1yco,rect1width,rect1height;

boolean mouseEntered;

boolean rect1Clicked;

publicvoid init()

 {

 rect1xco=20;

 rect1yco=20;

 rect1width=100;

 rect1height=50;

 addMouseListener(this);

}

publicvoid paint(Graphics g)

 {

 g.setColor(Color.green);

 g.fillRect(rect1xco,rect1yco,rect1width,rect1height);

 g.setColor(Color.red);

 g.drawString("mouse pointer", xpos, ypos);

if(rect1Clicked)

 {

 g.drawString("You clicked in the rectangle",20,120);

 }

else

 {

 g.drawString("You clicked outside the rectangle",20,120);

 }

if(mouseEntered)

 {

 g.drawString("Mouse is in the applet area",20,160);

 }

else

 {

 g.drawString("Mouse is outside the applet area",20,160);

 }

 }

publicvoid mouseClicked(MouseEvent me)

 {

 xpos=me.getX();

 ypos=me.getY();

if((xpos>rect1xco)&&(xpos<rect1xco+rect1width)&&(ypos>rect1yco)&&(ypos<rect1yco+rect1height))

 {

 rect1Clicked=true;

 }

else

 {

 rect1Clicked=false;

 repaint();

 }

 }

publicvoid mousePressed(MouseEvent me){}

publicvoid mouseReleased(MouseEvent me){}

publicvoid mouseEntered(MouseEvent me)

 {

 mouseEntered=true;

 repaint();

 }

publicvoid mouseExited(MouseEvent me)

 {

 mouseEntered=false;

 repaint();

 }

}



SHAPES AND COLORS

CODE:-

import java.awt.\*;

import java.applet.\*;

publicclass applet1 extends Applet

{Font bigFont;

Color redColor;

Color weirdColor;

Color bgColor;

publicvoid init()

 {bigFont=new Font("Arial",Font.BOLD,16);

 redColor=Color.red;

 weirdColor=new Color(60,60,122);

 bgColor=Color.blue;

 setBackground(bgColor);

 }

publicvoid paint(Graphics g)

 {g.setFont(bigFont);

 g.drawString("SHAPES AND COLOURS",80,20);

 g.setColor(redColor);

 g.drawRect(100,100,100,100);

 g.fillRect(110,110,80,80);

 g.setColor(weirdColor);

 g.fillArc(120,120,60,60,180,180);

 g.setColor(Color.yellow);

 g.drawLine(140,140,160,160);

 g.setColor(Color.black);} }



FONTS

CODE:-

import java.awt.\*;

import java.applet.\*;

publicclass applet7 extends java.applet.Applet

{ Font f1,f2,f3;

publicvoid init()

 { f1=new Font("Times New Roman",Font.BOLD|Font.ITALIC,12);

 f2=new Font("Lucida Sans",Font.BOLD,16);

 f3=new Font("Georgia",Font.PLAIN,20);

 }

publicvoid paint(Graphics g)

 { g.setFont(f1);

 g.drawString("Welcome",100,100);

 g.setFont(f2);

 g.drawString("Welcome",100,200);

 g.setFont(f3);

 g.drawString("Welcome",100,300);

 }

}



 ADDITION OF INTEGERS

CODE:-

import java.awt.\*;

import java.applet.\*;

publicclass applet6 extends Applet

{ Label p;

 TextField i;

int n,s;

publicvoid init()

 { p=new Label("Type an integer and press <Enter>");

 i=new TextField(10);

 add(p);

 add(i);

 s=0;

 }

publicboolean action(Event e,Object o)

 { n=Integer.parseInt(o.toString());

 i.setText("");

 s=s+n;

 showStatus(Integer.toString(s));

returntrue;

 }

}



 ACTION LISTENER

CODE:-

import java.awt.\*;

import java.awt.event.\*;

import java.applet.\*;

publicclass applet5 extends Applet implements ActionListener

{Button bluebutton;

Button redbutton;

Button greenbutton;

publicvoid init()

 {bluebutton=new Button("BLUE");

 redbutton=new Button("RED");

 greenbutton=new Button("GREEN");

 add(bluebutton);

 add(redbutton);

 add(greenbutton);

 bluebutton.addActionListener(this);

 redbutton.addActionListener(this);

 greenbutton.addActionListener(this);

}

publicvoid actionPerformed(ActionEvent evt)

 {if(evt.getSource()==bluebutton)

 {setBackground(Color.blue);

 }

elseif (evt.getSource()==redbutton)

 {setBackground(Color.red);

 }

elseif(evt.getSource()==greenbutton)

 {setBackground(Color.green);

 }

}

}



PANEL

CODE:-

import java.awt.\*;

import java.applet.\*;

publicclass applet9 extends java.applet.Applet

{

Label l1,l2,l3;

Panel p1,p2,p3;

publicvoid init()

 {

 setLayout(new BorderLayout());

 p1=new Panel();

 p1.setBackground(Color.blue);

 p2=new Panel();

 p2.setBackground(Color.green);

 p3=new Panel();

 p3.setBackground(Color.red);

 l1=new Label("THIS IS MY");

 p1.add(l1);

 l2=new Label("FIRST");

 p2.add(l2);

 l3=new Label("PANEL PROGRAM");

 p3.add(l3);

 add(p1,"North");

 add(p2,"Center");

 add(p3,"South");

}

publicvoid paint(Graphics g)

 {

}

}



 POINT CLASS

CODE:-

import java.awt.\*;

publicclass applet8 extends java.applet.Applet

{

Point p1,p2;

publicvoid init()

 {

 p1=new Point(100,250);

 p2=new Point(500,300);

}

publicvoid paint(Graphics g)

 {

 g.drawLine(p1.x,p1.y,p2.x,p2.y);

}

}

