**Department of Computer Science and Engineering**

This is to certify that it is a bonafide record of practical work done by Mr/Ms \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_bearing Roll No. \_\_\_\_\_\_\_\_\_\_\_\_\_\_ B.Tech \_\_\_\_\_\_\_\_\_ Year, \_\_\_\_\_ Semester, Computer Science and Engineering branch in the \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ laboratory during the academic year 2020-2021 has satisfactorily completed the course of programs prescribed by Malla Reddy Engineering College under my supervision.

Lab Incharge Head of the Dept

Internal Examiner External Examiner

**INDEX**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Name of the Experiment** | **Page No** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**MALLA REDDY ENGINEERING COLLEGE MREC(A)**

**JAVA ASSIGNMENT**

1. Write Java Programs that implement the following.

a) Constructor

**Program**

class Const

{

Const()

{

System.out.println("default constructor is executed");

}

public static void main(String args[])

{

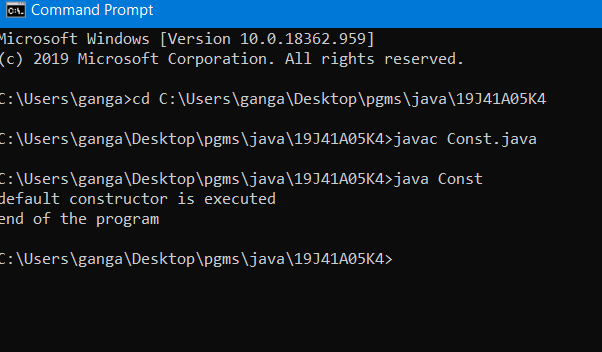
Const ob=new Const();

System.out.println("end of the program");

}

}

**Output**

****

(b) Parameterized constructor

**Program**

class ConDemo

{

int a;

double b,c;

String s;

ConDemo(int x)

{

System.out.println("Constructor with one int argument");

a=x;

}

ConDemo(double x,double y)

{

System.out.println("Constructor with two double arguments");

b=x;

c=y;

}

ConDemo(String x)

{

System.out.println("Constructor with one String argument");

s=x;

}

void display()

{

System.out.println("value of int a:"+a);

System.out.println("value of double b:"+b);

System.out.println("value of double c:"+c);

System.out.println("value of String s:"+s);

}

public static void main(String args[])

{

ConDemo ob=new ConDemo(10); // calls Constructor with one int argument

ConDemo ob1=new ConDemo(7.5,8.6); //Constructor with two double arguments

ConDemo ob2=new ConDemo("Hello");//Constructor with one string argument

ob.display();

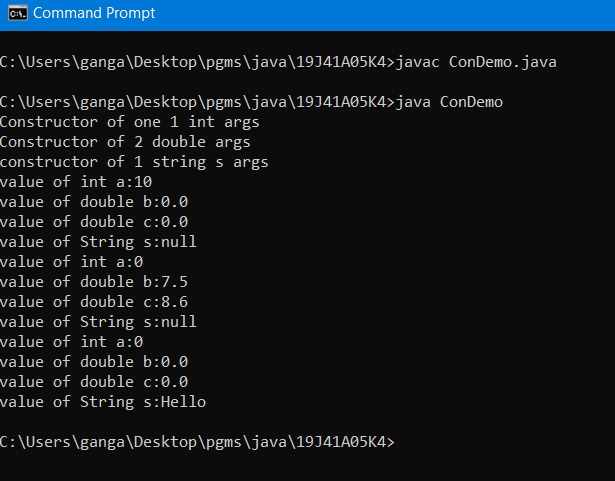
ob1.display();

ob2.display();

}

}

**Output:**

****

c) Method overloading

**Program:**

class Method

{

double vol;

int a,b,c;

void v()

{

a=2;

vol=a\*a\*a;

System.out.println("volume of cube is:"+vol);

}

void v(int len,int breadth,int height)

{

a=len;

b=breadth;

c=height;

vol=a\*b\*c;

System.out.println("volume of rectangle is:"+vol);

}

public static void main(String args[])

{

Method ob=new Method();

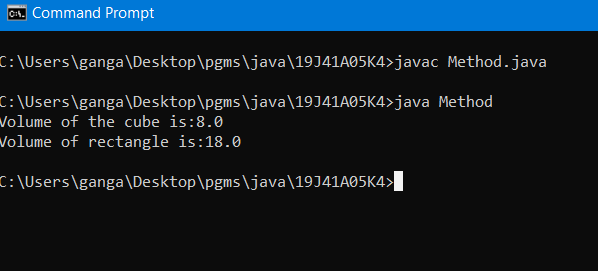
ob.v();

ob.v(2,3,3);

}

}

**Output:**

****

(d) Constructor overloading

**Program:**

class Const\_load

{

Const\_load()

{

System.out.println("Default constructor");

}

Const\_load(int a,int b)

{

System.out.println("parameterized constructor with 2 args");

System.out.println("a is:"+a+" and "+"b is:"+b);

}

Const\_load(double x,double y,String z)

{

System.out.println("parameterized constructor with 3 args");

System.out.println("x is:"+x);

System.out.println("y is:"+y);

System.out.println("z is:"+z);

}

public static void main(String args[])

{

Const\_load ob1=new Const\_load();

Const\_load ob2=new Const\_load(44,33);

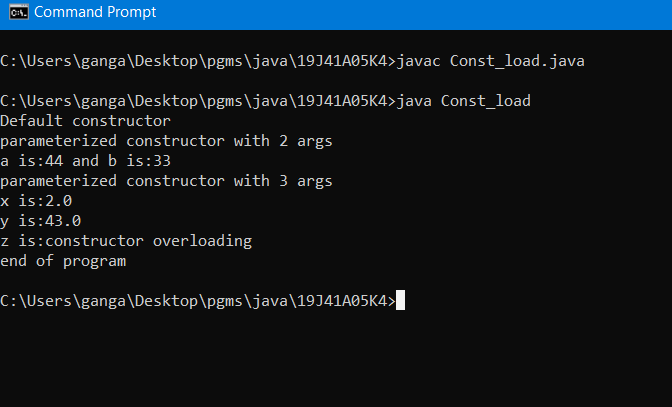
Const\_load ob3=new Const\_load(2,43,"constructor overloading");

System.out.println("end of program");

}

}

**Output:**

****

2.Write a Java program

1. Checks whether a given number is a palindrome or not.

**Program:**

import java.util.\*;

class Palindrome

{

void ispal()

{

Scanner sc=new Scanner(System.in);

System.out.println("enter the String:");

String s1=sc.next();

int n=s1.length();

String s2="";

for(int i=n-1;i>=0;i--)

{

char c=s1.charAt(i);

s2=s2+c;

}

if(s1.equals(s2)==true)

System.out.println("Given String is a palindrome...");

else

System.out.println("Given String is not a palindrome...");

}

public static void main(String args[])

{

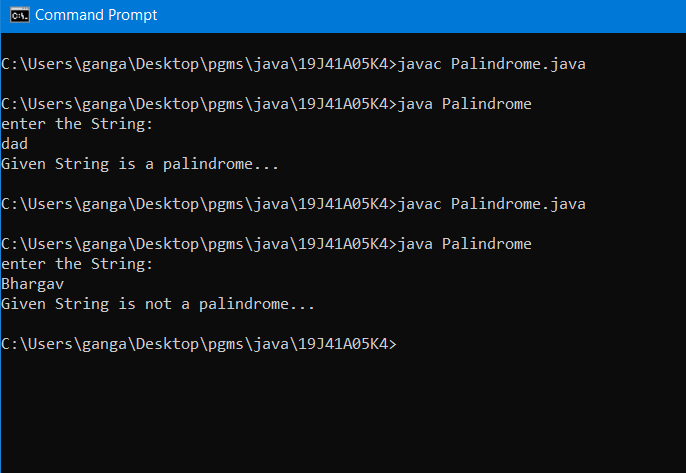
Palindrome ob=new Palindrome();

ob.ispal();

}

}

**Output:**



1. For sorting a given list of names in ascending order.

**Program:**

import java.util.\*;

class Ascending

{

public static void main(String args[])

{

Scanner sc=new Scanner(System.in);

System.out.println("enter length of list");

int n=sc.nextInt();

int a[]=new int[n];

System.out.println("enter list of elements");

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

{

a[i]=sc.nextInt();

}

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

{

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

{

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

{

int temp;

temp=a[i];

a[i]=a[j];

a[j]=temp;

}

}

}

System.out.println("elements in sorted order are:");

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

{

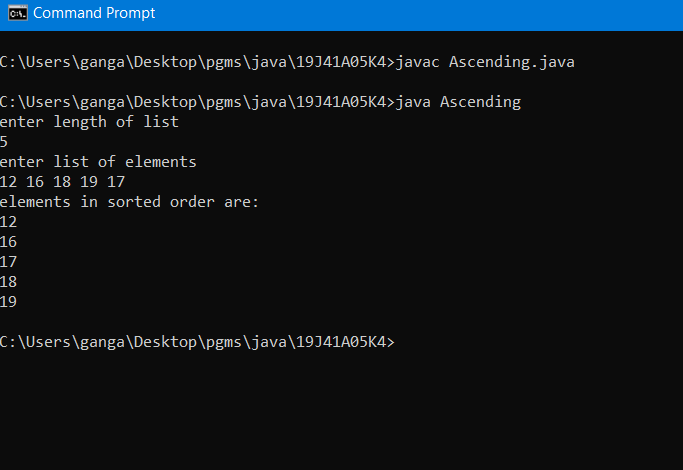
System.out.println(a[i]);

}

}

}

**Output:**



c) that reads a line of integers and then displays each integer and the sum of all integers(use string tokenizer class of java.util).

**Program:**

import java.util.\*;

class Token

{

public static void main(String args[])

{

Scanner sc=new Scanner(System.in);

System.out.println("enter line of text");

String text=sc.nextLine();

StringTokenizer st=new StringTokenizer(text);

int sum=0;

while(st.hasMoreTokens())

{

String s=st.nextToken();

int n=Integer.parseInt(s);

sum+=n;

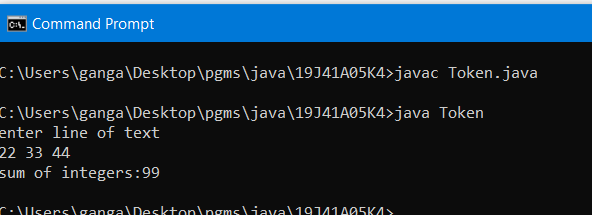
}

System.out.println("sum of integers:"+sum);

}

}

**Output:**



3.Write a Java program that uses the following keywords

(a) this

**Program:**

class This

{

This()

{

System.out.print("welcome ");

}

This(String s)

{

this();

System.out.print(s);

}

This(String t,String u)

{

this("to");

System.out.print(t+u);

}

public static void main(String args[])

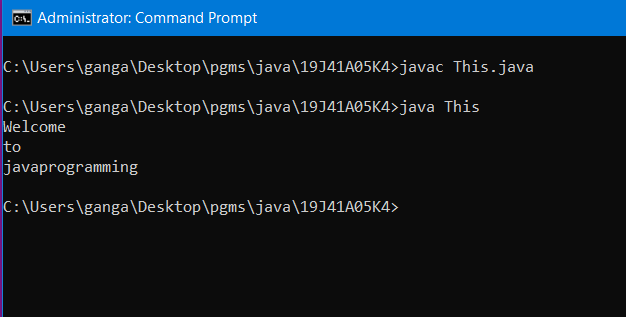
{

This ob=new This("java"," programming");

}

}

**Output:**



(b) Super

**Program:**

class Super

{

Super()

{

System.out.println("parrent class constructor");

}

void m(int a)

{

System.out.println("val of a is:"+a);

}

}

class Sub extends Super

{

Sub()

{

System.out.println("child class constructor");

super.m(10);

}

}

class Superclass

{

public static void main(String args[])

{

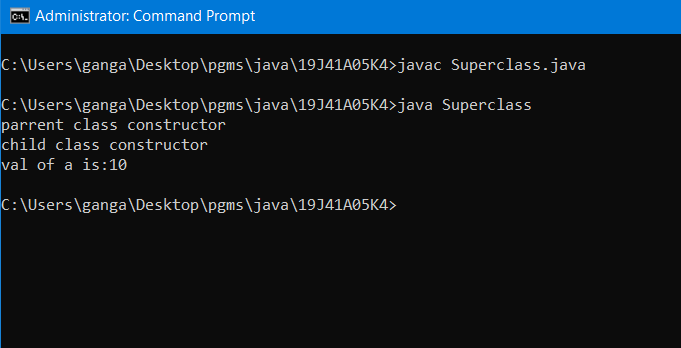
Superclass ob1=new Superclass();

Sub ob=new Sub();

}

}

**Output:**



c) static

**Program:**

class Static

{

static String s="hello world";

static

{

System.out.println("Static block 1");

}

static

{

System.out.println("Static block 1");

}

static int cube(int x)

{

return(x\*x\*x);

}

public static void main(String args[])

{

Static ob=new Static();

int result=Static.cube(10);

System.out.println(result);

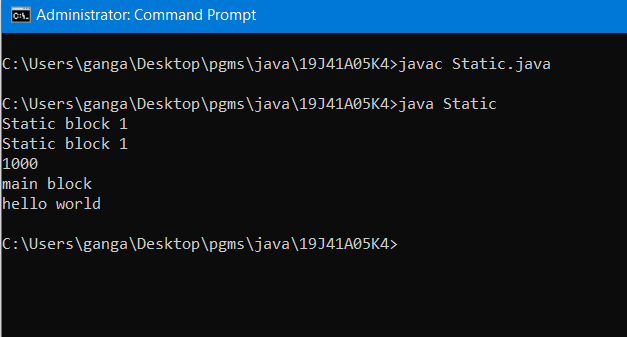
System.out.println("main block");

System.out.println(s);

}

}

**Output:**



(d) final

**Program:**

class A

{

final void m()

{

System.out.println("final method");

}

}

class B extends A

{

B()

{

System.out.println("class A constructor");

}

}

class Final

{

public static void main(String args[])

{

final int a=100;

Final ob=new Final();

B ob1=new B();

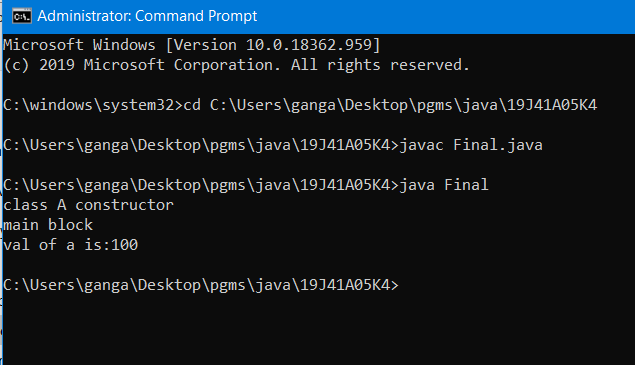
System.out.println("main block");

System.out.println("val of a is:"+a);

}

}

**Output:**



4. Write Java program to implement

(a) Method overriding.

**Program:**

class Super

{

void m(int a)

{

System.out.println("parent class method");

System.out.println("val of a is:"+a);

}

}

class Sub extends Super

{

void m()

{

super.m(10);

System.out.println("child class method");

}

}

class Method\_Override

{

public static void main(String args[])

{

Sub ob=new Sub();

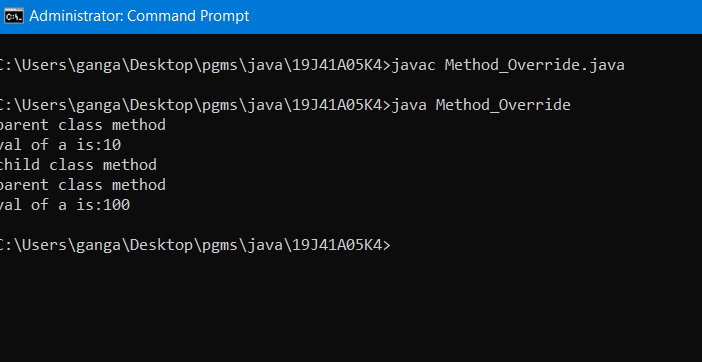
ob.m();

ob.m(100);

}

}

**Output:**

****

(b) Dynamic method dispatch.

**Program:**

class A

{

int a;

A()

{

a=10;

}

void m()

{

System.out.println("val of a is:"+a);

}

}

class B extends A

{

int b;

B()

{

b=20;

}

void m()

{

System.out.println("val of b is:"+b);

}

}

class C extends B

{

int c;

C()

{

c=30;

}

void m()

{

System.out.println("val of c is:"+c);

}

}

class Dynamic\_dispatch

{

public static void main(String args[])

{

System.out.println("main method");

A ob=new A();

ob.m();

ob=new B();

ob.m();

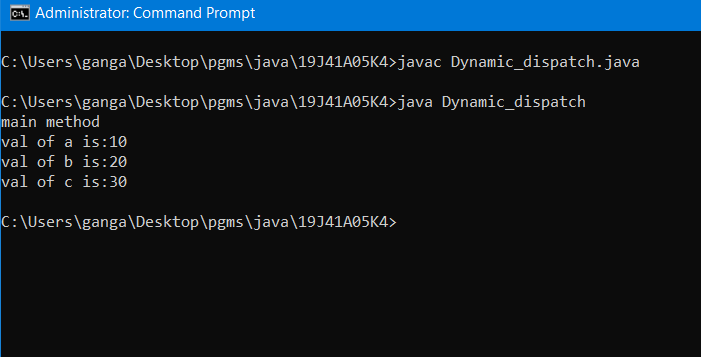
ob=new C();

ob.m();

}

}

**Output:**



c) Multiple inheritance.

**Program:**

interface Arith1

{

void add(int x,int y);

void sub(int x,int y);

}

interface Arith2

{

void mul(int x,int y);

void div(int x,int y);

}

class Arith implements Arith1,Arith2

{

public void add(int x,int y)

{

System.out.println("add of x and y is:"+(x+y));

}

public void sub(int x,int y)

{

System.out.println("sub of x and y is:"+(x-y));

}

public void mul(int x,int y)

{

System.out.println("mul of x and y is:"+(x\*y));

}

public void div(int x,int y)

{

System.out.println("div of x and y is:"+(x/y));

}

public static void main(String args[])

{

Arith ob=new Arith();

ob.add(1,2);

ob.sub(2,4);

ob.mul(5,4);

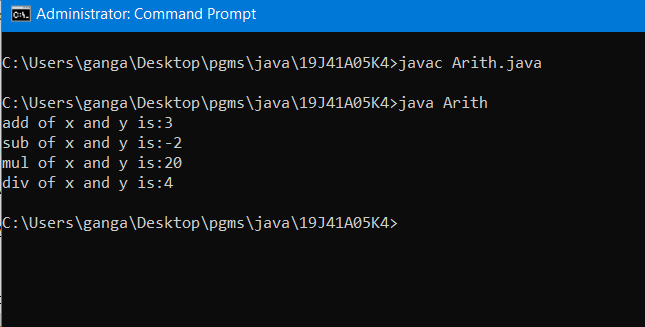
ob.div(8,2);

}

}

**Output:**

**\**

****

(d)Access specifiers

**Program:**

class A

{

public String s="access specifier";

String c="default value";

A()

{

System.out.println("public var:"+s);

System.out.println("length of default string is:"+c.length());

}

}

class B

{

private int a=20;

protected double b=11.11;

void m()

{

System.out.println("val of a is:"+a+"\n"+"val of b is:"+b);

}

}

class C extends B

{

void n()

{

System.out.println("protected var val of b is:"+b);

}

}

class Acess\_specifier

{

public static void main(String args[])

{

A ob=new A();

C ob1=new C();

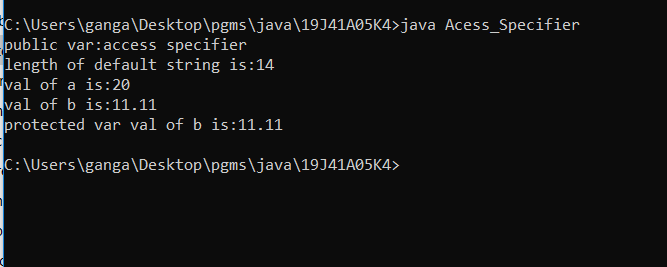
ob1.m();

ob1.n();

}

}

**Output:**



5. Write a Java program to implement

(a) reads a file name from the user ,and then displays information about whether the file exits, whether the file is readable , whether the file is writable, the type of file and the length of the file in bytes.

**Program:**

import java.util.\*;

import java.io.\*;

class Lab\_file1

{

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

{

Scanner sc=new Scanner(System.in);

System.out.println("enter the file with extension");

String fname=sc.next();

File ob=new File(fname);

int pos=fname.indexOf('.');

String type=fname.substring(pos+1);

System.out.println("file exist or not:"+ob.exists());

System.out.println("is it readable:"+ob.canRead());

System.out.println("is it Writable:"+ob.canWrite());

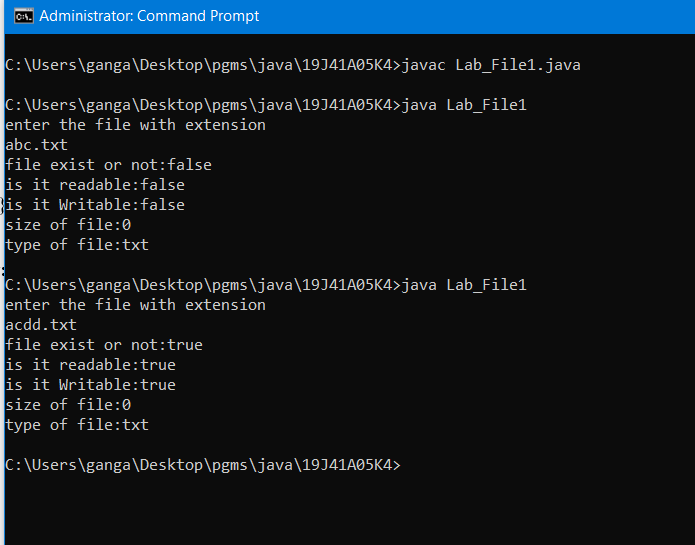
System.out.println("size of file:"+ob.length());

System.out.println("type of file:"+type);

}

}

**Output:**

****

(b) reads a file and displays the file on the screen, with a line number before each line.

**Program:**

import java.io.\*;

class Lab\_file

{

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

{

FileReader ob=new FileReader("abc.txt");

BufferedReader ob1=new BufferedReader(ob);

String s="";

int linenum=1;

while((s=ob1.readLine())!=null)

{

System.out.println(linenum+":"+s);

linenum++;

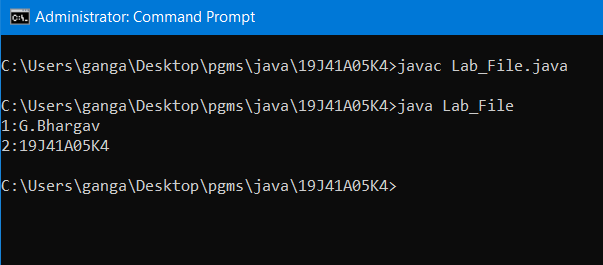
}

ob.close();

}

}

**Output:**

****

c) displays the number of characters, lines and words in a text file.

**Program:**

import java.io.\*;

public class Lab5c

{

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

{

File file = new File("abc.txt");

FileInputStream fileStream = new FileInputStream(file);

InputStreamReader input = new InputStreamReader(fileStream);

BufferedReader reader = new BufferedReader(input);

String line;

int countWord = 0;

int sentenceCount = 0;

int characterCount = 0;

while((line = reader.readLine()) != null)

{

characterCount += line.length();

String[] wordList = line.spl it("\\s+");

countWord += wordList.length;

String[] sentenceList = line.split("[!?.:]+");

sentenceCount += sentenceList.length;

}

System.out.println("Total word count = " + countWord);

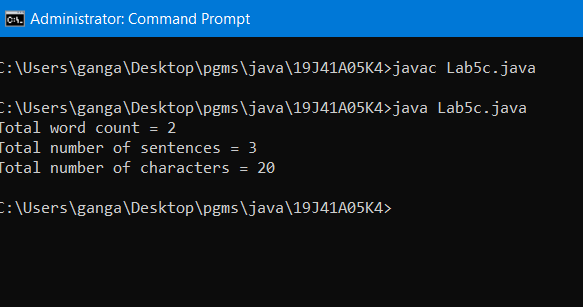
System.out.println("Total number of sentences = " + sentenceCount);

System.out.println("Total number of characters = " + characterCount);

}

}

**Output:**

****

6. Write a java program for handling

(a) Checked Exception

**Program:**

import java.io.\*;

class Checked\_Except

{

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

{

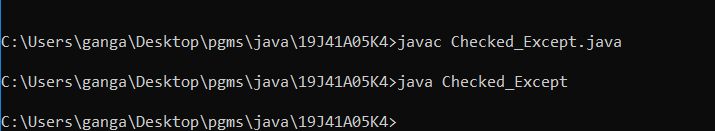
File f=new File("abc.txt");

f.createNewFile();

}

}

**Output:**



(b) Unchecked Exception.

**Program:**

import java.util.\*;

class ArithHandle

{

public static void main(String args[])

{

Scanner sc=new Scanner(System.in);

System.out.println("enter numerator and denominator");

int a=sc.nextInt();

int b=sc.nextInt();

try

{

int c=a/b;

System.out.println(c);

}

catch(Exception e)

{

System.out.println("the Exception generated is:"+e);

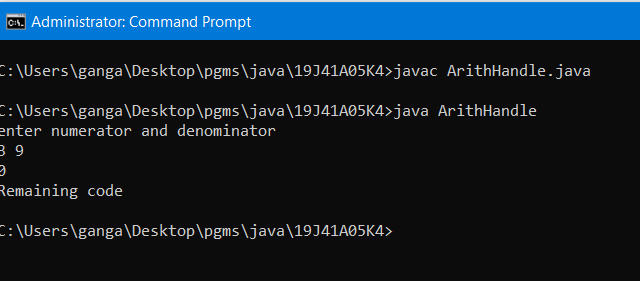
}

System.out.println("Remaining code");

}

}

**Output:**



7. Write a java program

(a) Create three threads. First thread displays “Good Morning” for every one second, Second thread displays “Hello” for every two seconds ,the third thread displays “welcome” for every three seconds.

**Program:**

class Thread1 extends Thread

{

public void run()

{

try

{

Thread.sleep(1000);

}

catch(Exception e)

{

System.out.println(e);

}

System.out.println("Good Morning");

}

}

class Thread2 extends Thread

{

public void run()

{

try

{

Thread.sleep(2000);

}

catch(Exception e)

{

System.out.println(e);

}

System.out.println("Hello");

}

}

class Thread3 extends Thread

{

public void run()

{

try

{

Thread.sleep(3000);

}

catch(Exception e)

{

System.out.println(e);

}

System.out.println("Welcome");

}

}

class Main\_Thread

{

public static void main(String args[])

{

Thread t1=new Thread1();

Thread t2=new Thread2();

Thread t3=new Thread3();

t1.start();

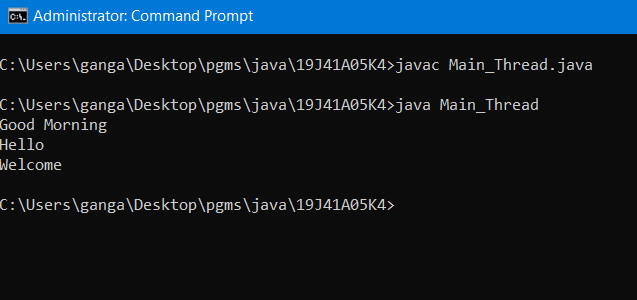
t2.start();

t3.start();

}

}

**Output:**



(b) that correctly implements producer consumer problem using concept of inter thread communication.

**Program:**

class PC

{

boolean lock=false;

int item;

synchronized void producer(int x)

{

if(lock)

{

try

{

wait();

}

catch(Exception e)

{

System.out.println(e);

}

}

item=x;

System.out.println("itemproduced:"+item);

lock=true;

notify();

}

synchronized void consumer()

{

if(!lock)

{

try

{

wait();

}

catch(Exception e)

{

System.out.println(e);

}

}

System.out.println("item consumed:"+item);

lock=false;

notify();

}

}

class PCdemo

{

public static void main(String args[])

{

PC ob=new PC();

new Thread()

{

public void run()

{

for(int i=1;i<=10;i++)

ob.producer(i);

}

}.start();

new Thread()

{

public void run()

{

for(int i=1;i<=10;i++)

ob.consumer();

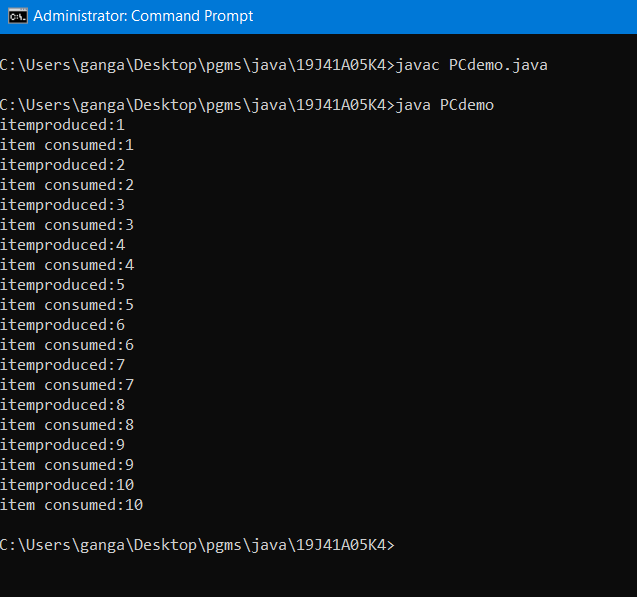
}

}.start();

}

}

**Output:**



8. Develop an Applet that

(a) Displays a simple message.

**Program:**

import java.applet.\*;

import java.awt.\*;

public class Applet\_sample extends Applet

{

public void paint(Graphics g)

{

g.drawString("welcome to applet programming",50,50);

}

}

/\*

<applet code="Applet\_sample.class" height="300" width="300">

</applet>

\*/

**Output:**

(b) receives an integer in one text field and computes its factorial value and returns it in another text field , when the button named “Compute” is clicked.

**Program:**

import java.awt.\*;

import java.awt.event.\*;

import java.applet.\*;

/\*<applet code="Factorial.class" width=500 height=200></applet>\*/

public class Factorial extends Applet implements ActionListener

{

Button compute;

int fact=0;

TextField input,output;

public void init()

{

compute=new Button("Compute");

Label inp=new Label("Enter any number :");

Label opt=new Label("Factorial of number is : ");

input=new TextField(5);

output=new TextField(10);

add(inp);

add(input);

add(opt);

add(output);

add(compute);

output.setText("0");

output.setEditable(false);

compute.addActionListener(this);

}

public void actionPerformed(ActionEvent ae)

{

String str=ae.getActionCommand();

if(str.equals("Compute"))

{

fact=1;

int n=Integer.parseInt(input.getText());

if(n<=12)

{

for(int i=n;i>=2;i--)

fact=fact\*i;

output.setText(""+fact);

}

else

fact=-1;

}

repaint();

}

public void paint(Graphics g)

{

if(fact==-1)

{

output.setText("0");

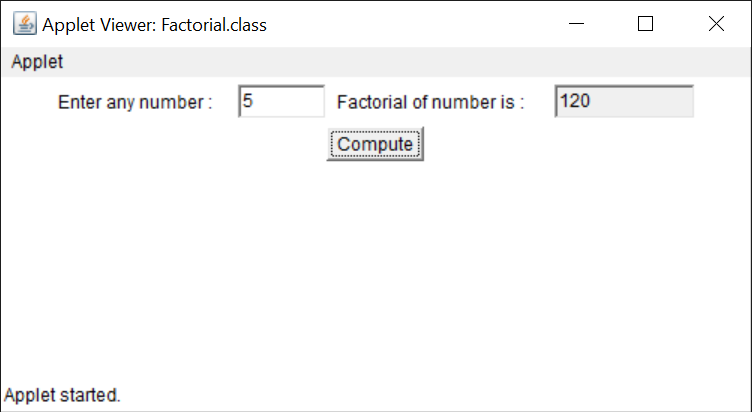
g.drawString("Sorry number exceeds greater than 12",100,100);

}

}

}

**Output:**



9 . Write a Java program that works as a simple calculator. Use a grid layout to arrange buttons for the digits and for the +,-,\*,/ operations. Add a text field to display the result.

**Program :**

//Program for implementing a Simple Calculator

import java.awt.\*;

import java.awt.event.\*;

import java.applet.\*;

/\*<applet code="Calculator1" width=300 height=300></applet>\*/

public class Calculator1 extends Applet implements ActionListener

{

TextField t;

Button b[]=new Button[15];

Button b1[]=new Button[6];

String op2[]={"+","-","\*","/","=","C"};

String str1="";

String str2="";

String msg="";

int p=0,q=0;

String oper;

public void init()

{

setLayout(new GridLayout(5,4));

t=new TextField(20);

setBackground(Color.pink);

setFont(new Font("Arial",Font.BOLD,20));

int k=0;

t.setEditable(false);

t.setBackground(Color.white);

t.setText("0");

for(int i=0;i<10;i++)

{

b[i]=new Button(""+k);

add(b[i]);

k++;

b[i].setBackground(Color.pink);

b[i].addActionListener(this);

}

for(int i=0;i<6;i++)

{

b1[i]=new Button(""+op2[i]);

add(b1[i]);

b1[i].setBackground(Color.pink);

b1[i].addActionListener(this);

}

add(t);

}

public void actionPerformed(ActionEvent ae)

{

String str=ae.getActionCommand();

if(str.equals("+")){

p=Integer.parseInt(t.getText());

oper=str;

msg=p+oper;

t.setText(msg);

str2="";

}

else if(str.equals("-")){

p=Integer.parseInt(t.getText());

oper=str;

msg=p+oper;

t.setText(msg);

str2="";

}

else if(str.equals("\*")){

p=Integer.parseInt(t.getText());

oper=str;

msg=p+oper;

t.setText(msg);

str2="";

}

else if(str.equals("/")){

p=Integer.parseInt(t.getText());

oper=str;

msg=p+oper;

t.setText(msg);

str2="";

}

else if(str.equals("=")) {

if(oper.equals("+")) {

q=Integer.parseInt(str2);

t.setText(msg=String.valueOf((p+q)));

}

else if(oper.equals("-")) {

q=Integer.parseInt(str2);

t.setText(msg=String.valueOf((p-q))); }

else if(oper.equals("\*")){

q=Integer.parseInt(str2);

t.setText(msg=String.valueOf((p\*q))); }

else if(oper.equals("/")){

q=Integer.parseInt(str2);

t.setText(msg=String.valueOf((p/q))); }

}

else if(str.equals("C")){ p=0;q=0;

t.setText("");

str2=""; msg="";

t.setText("0");

}

else{

str2+=str;

msg=msg+str;

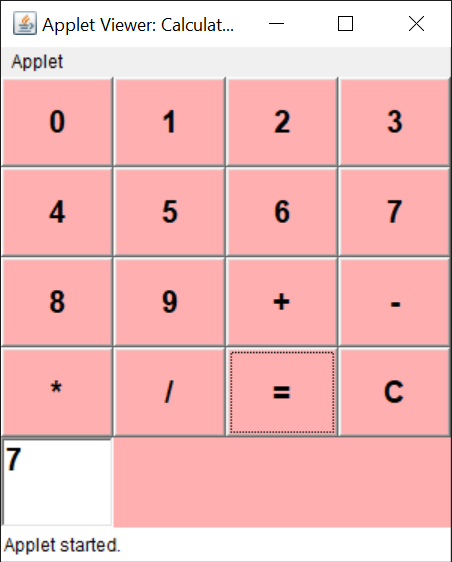
t.setText(msg);

}

}

}

**Output :**



10. Write a Java program for handling a) mouse events

**Program :**

import java.awt.\*;

import java.awt.event.\*;

public class MouseListenerExample extends Frame implements MouseListener{

Label l;

MouseListenerExample(){

addMouseListener(this);

l=new Label();

l.setBounds(20,50,100,20);

add(l);

setSize(300,300);

setLayout(null);

setVisible(true);

}

public void mouseClicked(MouseEvent e) {

l.setText("Mouse Clicked");

}

public void mouseEntered(MouseEvent e) {

l.setText("Mouse Entered");

}

public void mouseExited(MouseEvent e) {

l.setText("Mouse Exited");

}

public void mousePressed(MouseEvent e) {

l.setText("Mouse Pressed");

}

public void mouseReleased(MouseEvent e) {

l.setText("Mouse Released");

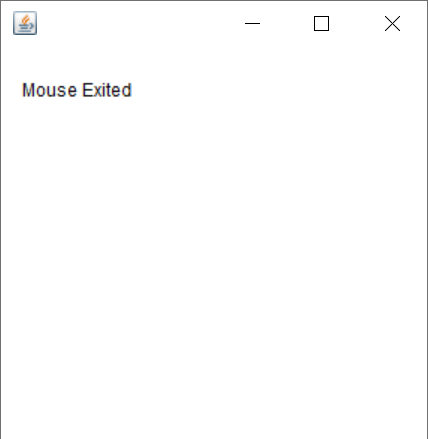
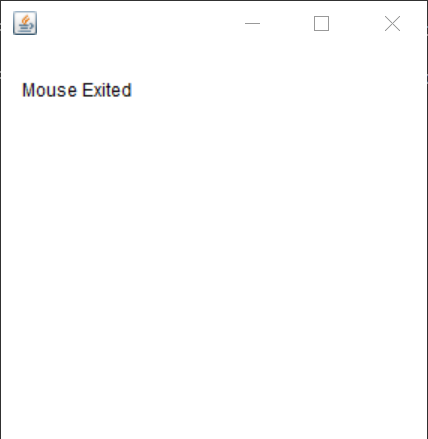
}

public static void main(String[] args) {

new MouseListenerExample();

} }

**Output :**



. b) key events.

**Program :**

import java.awt.\*;

import java.awt.event.\*;

public class KeyListenerExample extends Frame implements KeyListener{

Label l;

TextArea area;

KeyListenerExample(){

l=new Label();

l.setBounds(20,50,100,20);

area=new TextArea();

area.setBounds(20,80,300, 300);

area.addKeyListener(this);

add(l);add(area);

setSize(400,400);

setLayout(null);

setVisible(true);

}

public void keyPressed(KeyEvent e) {

l.setText("Key Pressed");

}

public void keyReleased(KeyEvent e) {

l.setText("Key Released");

}

public void keyTyped(KeyEvent e) {

l.setText("Key Typed");

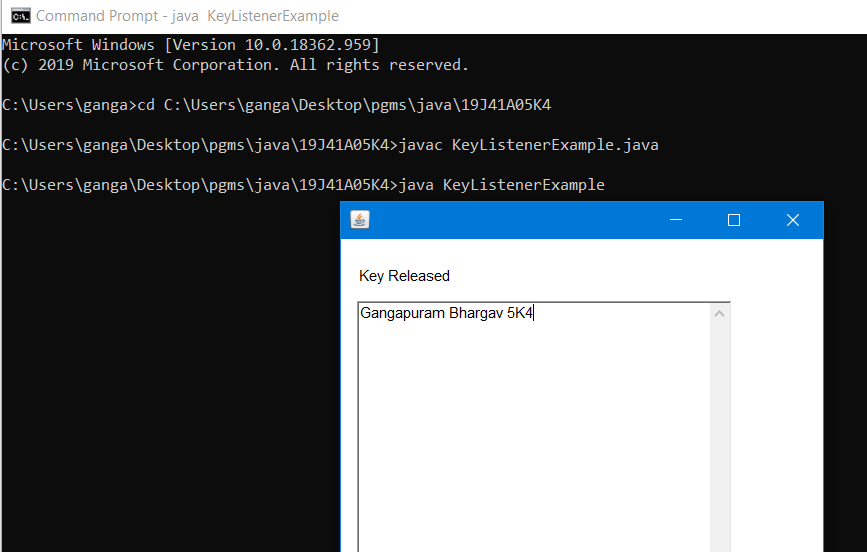
}

public static void main(String[] args) {

new KeyListenerExample(); }

}

Output :



11. Write a Java program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields num1 and num2. The division of num1 and num2 is displayed in the result field when the divide button is clicked. If num1 or num2 were not an integer, the program would throw number format exception. If num2 were zero, the program would throw an arithmetic exception and display the exception in the message dialogue box.

**Program:**

import java.awt.\*;

import javax.swing.\*;

import java.applet.Applet;

import java.awt.event.\*;

public class Division extends Applet implements ActionListener{

TextField t1,t2,t3;

Button b;

Label L1,L2,L3,L4;

String s;

public void init()

{

t1=new TextField(10);

t2=new TextField(10);

t3=new TextField(10);

L1=new Label("enter num1");

L2=new Label("enter num2");

L3=new Label("Result is");

L4=new Label("Division of 2 numbers");

b=new Button("Divide");

add(L4);

add(L1);

add(t1);

add(L2);

add(t2);

add(L3);

add(t3);

add(b);

b.addActionListener(this);

}

public void actionPerformed(ActionEvent ae)

{

try

{

int num1=Integer.parseInt(t1.getText());

int num2=Integer.parseInt(t2.getText());

s=""+(num1/num2);

t3.setText(s);

}

catch(ArithmeticException a)

{

JOptionPane.showMessageDialog(null,"Divide by zero");

}

catch(NumberFormatException b)

{

JOptionPane.showMessageDialog(null,"NumberFormat Exception");

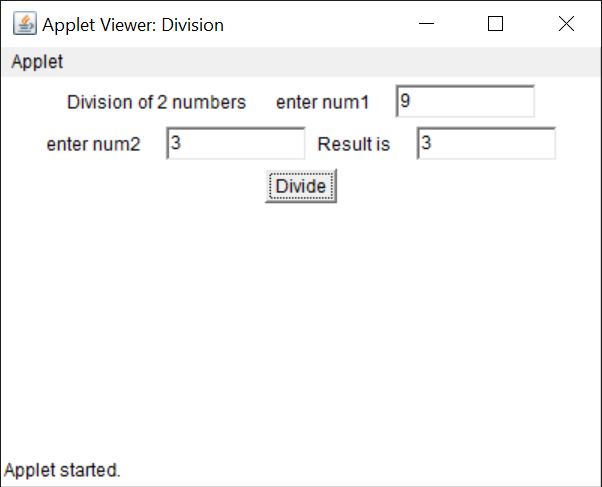
}

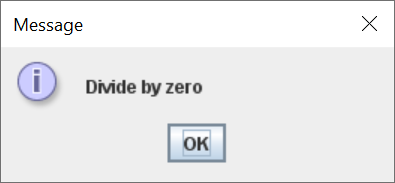
}

}

/\*<applet code="Division" width=400 height=250></applet>\*/

**Output :**





12. Write a Java program that a) Simulates traffic light. The program lets the user select one of three lights: red, yellow or green. When a radio button is selected, the light is turned on and only one light can be on at a time. No light is on when the program starts.

**Program:**

//Program for implementing Traffic Signals

import java.applet.\*;

import java.awt.\*;

import java.awt.event.\*;

/\*<applet code="Signals" width=400 height=250></applet>\*/

public class Signals extends Applet implements ItemListener

{

String msg="";

Checkbox stop,ready,go;

CheckboxGroup cbg;

public void init()

{

cbg = new CheckboxGroup();

stop = new Checkbox("Stop", cbg, false);

ready = new Checkbox("Ready", cbg, false);

go= new Checkbox("Go", cbg, false);

add(stop);

add(ready);

add(go);

stop.addItemListener(this);

ready.addItemListener(this);

go.addItemListener(this);

}

public void itemStateChanged(ItemEvent ie)

{

msg=cbg.getSelectedCheckbox().getLabel();

if(msg!=null)

repaint();

}

public void paint(Graphics g)

{

g.drawOval(165,40,50,50);

g.drawOval(165,100,50,50);

g.drawOval(165,160,50,50);

if(msg.equals("Stop"))

{

g.setColor(Color.red);

g.fillOval(165,40,50,50);

}

else if(msg.equals("Ready"))

{

g.setColor(Color.yellow);

g.fillOval(165,100,50,50);

}

else if(msg.equals("Go"))

{

g.setColor(Color.green);

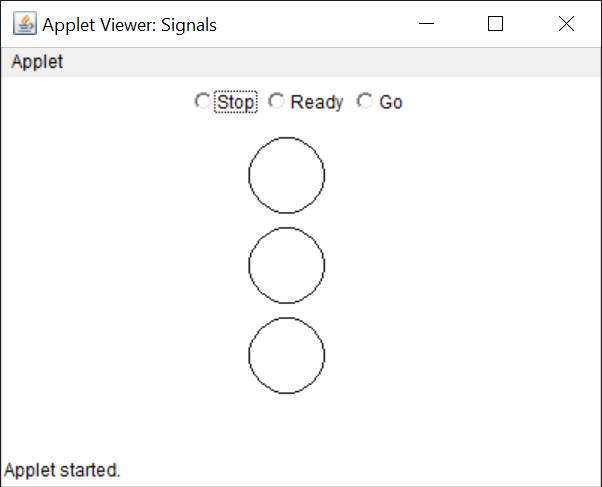
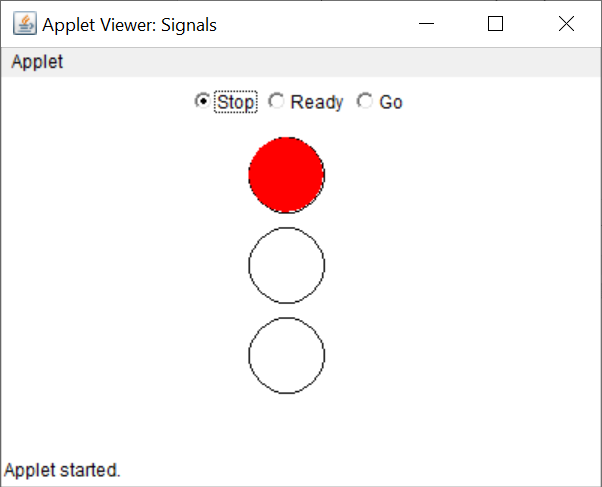
g.fillOval(165,160,50,50);

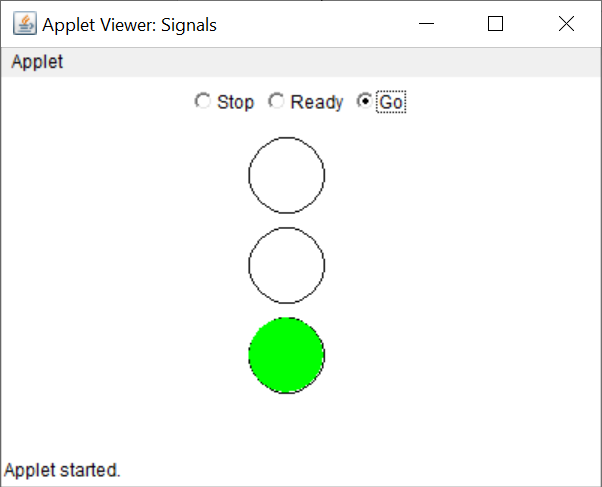
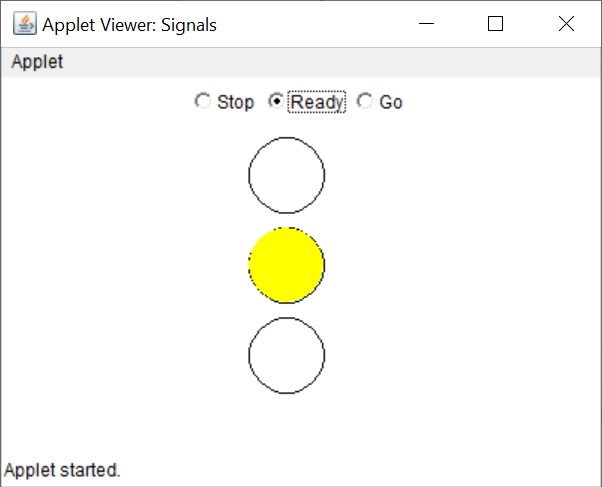
}

}

}

**Output :**

****

****

b) Allows the user to draw lines rectangles and ovals.

import java.applet.Applet;

import java.awt.\*;

public class GraphicsDemo extends Applet

{

public void paint(Graphics g)

{

g.setColor(Color.red);

g.drawString("welcome",50,50);

g.drawRect(70,100,30,30);

g.fillRect(170,100,30,30);

g.drawOval(70,200,30,30);

g.setColor(Color.pink);

g.fillOval(170,200,30,30);

showStatus("this is shown in status window.");

}

}

/\*<applet code="GraphicsDemo.class" width=400 height=300>

</applet>

\*/