Practical 1: Write Swing Program to divide screen in horizontally in 3 equal parts and fill it with different colours.

import javax.swing.\*;

import java.awt.\*;

public class LayoutColor {

public static void main(String[] args) {

JFrame f= new JFrame();

JPanel panel=new JPanel();

panel.setBackground(Color.orange);

JPanel b1=new JPanel();

b1.setBackground(Color.white);

JPanel b2=new JPanel();

b2.setBackground(Color.green);

f.add(panel);

f.add(b1);

f.add(b2);

f.setLayout(new GridLayout(3,1));

f.setSize(400,600);

f.setVisible(true);

}

}

OUTPUT:

****

Practical 2: Write Swing code to display digital clock .

import javax.swing.\*;

import java.awt.\*;

import java.text.\*;

import java.util.\*;

public class DigiteClock implements Runnable{

JFrame f;

Thread t=null;

int hours=0, minutes=0, seconds=0;

String timeString = "";

JButton b;

DigiteClock(){

f=new JFrame();

t=new Thread(this);

t.start();

b=new JButton();

b.setBounds(100,100,100,50);

f.add(b);

f.setSize(300,400);

f.setLayout(null);

f.setVisible(true);

}

public void run() {

try {

while (true) {

Calendar cal = Calendar.getInstance();

hours = cal.get( Calendar.HOUR\_OF\_DAY );

minutes = cal.get( Calendar.MINUTE );

seconds = cal.get( Calendar.SECOND );

SimpleDateFormat formatter = new SimpleDateFormat("hh:mm:ss");

Date date = cal.getTime();

timeString = formatter.format( date );

b.setText(timeString);

}

}

catch (Exception e) { }

}

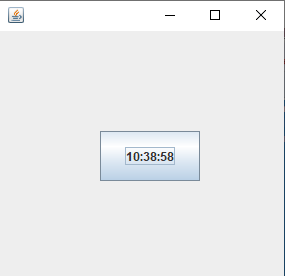
public static void main(String[] args) {

new DigiteClock();

}

}

OUTPUT:

****

Practical 3: Write a Program in Java to implement Calculator (+, -, \*, /, %, sqrt,pow, sin, cos, tan) using Swing..

* “=” Button Actions

JButton button\_10 = new JButton("=");

button\_10.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

double second=Double.parseDouble(textField.getText());

double first=Double.parseDouble(textField\_1.getText());

if(actionRecived.getText().equals("Sub")) {

double sub=first-second;

textField\_1.setText(first+" - "+second);

textField.setText(Double.toString(sub));

}

else if(actionRecived.getText().equals("Sum")) {

double sum=first+second;

textField\_1.setText(first+" + "+second);

textField.setText(Double.toString(sum));

}

else if(actionRecived.getText().equals("Div")) {

double div=first/second;

textField\_1.setText(first+" / "+second);

textField.setText(Double.toString(div));

}

else if(actionRecived.getText().equals("Mul")) {

double mul=first\*second;

textField\_1.setText(first+" \* "+second);

textField.setText(Double.toString(mul));

}

}

});

* “+,/,\*,-“ Button Actions

JButton button\_11 = new JButton("-");

button\_11.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

textField\_1.setText(textField.getText());

actionRecived.setText("Sub");

textField.setText("0");

}

});

* “C” Button Actions(Clear)

JButton btnC = new JButton("C");

btnC.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

textField.setText("0");

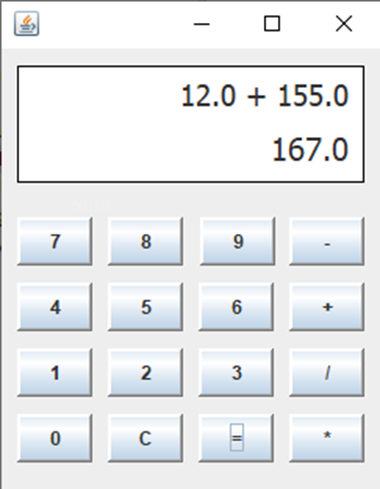
textField\_1.setText(null);

actionRecived.setText(null);

}

});

OUTPUT:



Practical 2.1: Create a Math Server (Factorial, Fibonacci, Prime, and Palindrome) with multi-client support using TCP.

SERVER :

import java.net.\*;

import java.io.\*;

import java.math.BigInteger;

public class Mathrealserver

{

public Mathrealserver(int port)

{

Socket soc = null;

ServerSocket ss = null;

try {

ss = new ServerSocket(port);

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

System.out.println("Server started");

System.out.println("Waiting for a client ...");

while(true)

{

try

{

soc = ss.accept();

System.out.println("Client accepted");

DataInputStream dis = new DataInputStream(soc.getInputStream());

DataOutputStream dout = new DataOutputStream(soc.getOutputStream());

String line = "";

String ans=new String();

line = (String)dis.readUTF();

String[] test=line.split("#");

String str=test[0];

if(str.isEmpty())

ans="Enter Text First";

else

{

int data=Integer.parseInt(str);

if(test[1].equals("1"))

{

int flag=0;

for(int i=2;i<data/2;i++)

{

if(data%i==0)

{

flag=1;

ans=data+" is not Prime.";

break;

}

}

if(flag==0)

{

ans=data+" is Prime.";

}

}

else if(test[1].equals("2"))

{

int len=str.length();

int f=0;

for(int i=0;i<len/2;i++)

{

if(str.charAt(i)!=str.charAt(--len))

{

f=1;

ans=str+" is not Palidrome.";

break;

}

}

if( f==0)

ans=str+" is Palidrome.";

}

else if(test[1].equals("3"))

{

BigInteger f = new BigInteger("1");

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

f = f.multiply(BigInteger.valueOf(i));

String str1 = f.toString();

ans=str1;

}

else if(test[1].equals("4"))

{

String ans2;

ans2="0 1 ";

int a=0,b=1,temp;

for(int i=0;i<data-2;i++)

{

temp=a+b;

a=b;

b=temp;

ans2=ans2+Integer.toString(b)+" “;

}

ans=ans2;

}

}

dout.writeUTF(ans);

dout.flush();

}

catch(IOException i)

{

System.out.println(i);

}

}

}

public static void main(String args[])

{

Mathrealserver server = new Mathrealserver(8004);

}

}

CLIENT :

import java.awt.EventQueue;

import javax.swing.JFrame;

import javax.swing.JPanel;

import java.awt.BorderLayout;

import javax.swing.JTextField;

import javax.swing.JLabel;

import java.awt.Font;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.DataInputStream;

import java.io.DataOutputStream;

import java.io.IOException;

import java.io.PrintWriter;

import java.math.BigInteger;

import java.net.Socket;

import java.net.UnknownHostException;

import javax.swing.JRadioButton;

import javax.swing.ButtonGroup;

import javax.swing.JButton;

import javax.swing.border.LineBorder;

import java.awt.Color;

import javax.swing.border.BevelBorder;

import java.awt.Component;

import javax.swing.SwingConstants;

import javax.swing.UIManager;

public class mathserver {

private JFrame frame;

private JTextField input;

private JTextField output;

private JLabel lblNewLabel\_1;

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

mathserver window = new mathserver();

window.frame.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

public mathserver() {

initialize();

}

private void initialize() {

frame = new JFrame();

frame.setBounds(100, 100, 518, 341);

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

JPanel panel = new JPanel();

frame.getContentPane().add(panel, BorderLayout.CENTER);

panel.setLayout(null);

input = new JTextField();

input.setBounds(177, 79, 153, 22);

panel.add(input);

input.setColumns(10);

output = new JTextField();

output.setBorder(new BevelBorder(BevelBorder.LOWERED, null, null, null, null));

output.setBounds(96, 205, 316, 41);

panel.add(output);

output.setColumns(10);

JLabel lblNewLabel = new JLabel("Math Server Created by Harshil");

lblNewLabel.setHorizontalAlignment(SwingConstants.CENTER);

lblNewLabel.setBorder(new BevelBorder(BevelBorder.LOWERED, null, null, null, null));

lblNewLabel.setFont(new Font("Segoe UI", Font.PLAIN, 20));

lblNewLabel.setBounds(107, 13, 305, 41);

panel.add(lblNewLabel);

ButtonGroup G1 = new ButtonGroup();

lblNewLabel\_1 = new JLabel("Input");

lblNewLabel\_1.setBounds(112, 75, 77, 31);

panel.add(lblNewLabel\_1);

JRadioButton factbtn = new JRadioButton("Factorial");

factbtn.setBounds(108, 122, 127, 25);

panel.add(factbtn);

G1.add(factbtn);

JRadioButton palibtn = new JRadioButton("Palindrome");

palibtn.setBounds(108, 154, 127, 25);

panel.add(palibtn);

G1.add(palibtn);

JRadioButton fibobtn = new JRadioButton("Fibonacci");

fibobtn.setBounds(239, 122, 98, 25);

panel.add(fibobtn);

G1.add(fibobtn);

JRadioButton primebtn = new JRadioButton("Prime");

primebtn.setBounds(239, 154, 98, 25);

panel.add(primebtn);

G1.add(primebtn);

JButton runbtn = new JButton("RUN!");

runbtn.setBounds(368, 78, 109, 25);

panel.add(runbtn);

JLabel lblNewLabel\_2 = new JLabel("Result");

lblNewLabel\_2.setBounds(46, 216, 67, 28);

panel.add(lblNewLabel\_2);

runbtn.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent arg0) {

String str;

String option=new String();

try

{

Socket s=new Socket("localhost",8004);

DataInputStream dis = new DataInputStream(s.getInputStream());

DataOutputStream dout = new DataOutputStream(s.getOutputStream());

str=input.getText();

if(primebtn.isSelected())

option="1";

else if(palibtn.isSelected())

option="2";

else if(factbtn.isSelected())

option="3";

else if(fibobtn.isSelected())

option="4";

str=str+"#"+option;

dout.writeUTF(str);

String s2=new String();

s2 = (String)dis.readUTF();

output.setText(s2);

s.close();

}

catch(Exception e)

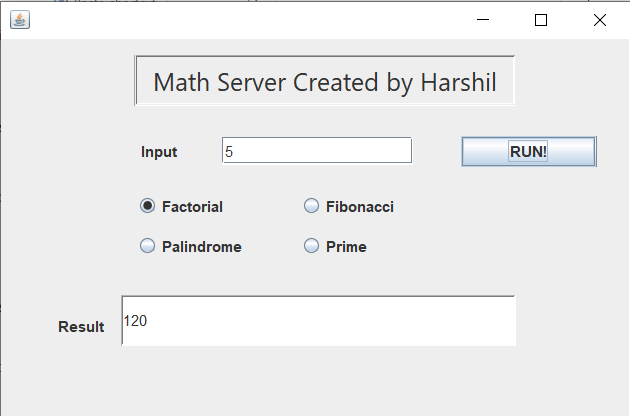
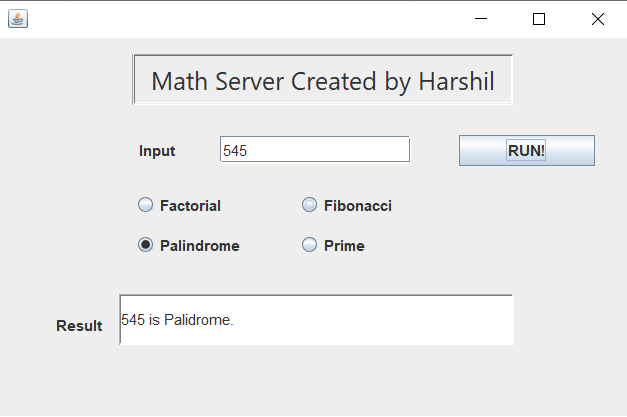
{

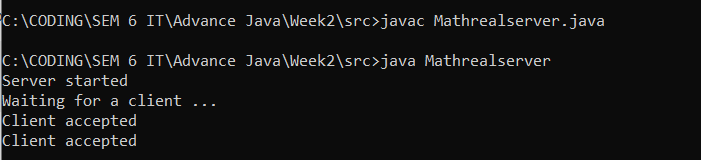
e.printStackTrace();

}

}

});}}



Practical 2.2:Implement any one sorting algorithm using TCP on Server application and Give input on client side and client should sorted output from server and display sorted on input side.

SERVER :

import java.net.\*;

import java.io.\*;

public class sortingserver

{

public sortingserver(int port)

{

Socket soc = null;

ServerSocket ss = null;

try

{

ss = new ServerSocket(port);

}

catch (IOException e)

{

e.printStackTrace();

}

System.out.println("Server started");

System.out.println("Waiting for a client ...");

while(true)

{

try

{

soc = ss.accept();

System.out.println("Client accepted");

DataInputStream dis = new DataInputStream(soc.getInputStream());

DataOutputStream dout = new DataOutputStream(soc.getOutputStream());

String line = "";

String ans=new String();

line = (String)dis.readUTF();

String[] test=line.split(" ");

if(line.isEmpty())

{

ans="Enter Text First";

}

else

{

int len=test.length;

int[] arr=new int[len];

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

{

arr[i]=Integer.parseInt(test[i]);

}

for(int i=0;i<len-1;i++)

{

for(int j=0;j<len-i-1;j++)

{

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

{

int temp = arr[j];

arr[j] = arr[j+1];

arr[j+1] = temp;

}

}

}

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

ans=ans+Integer.toString(arr[i])+" ";

}

dout.writeUTF(ans);

dout.flush();

}

catch(IOException i)

{

System.out.println(i);

}

}

}

public static void main(String args[])

{

sortingserver server = new sortingserver(8004);

}

}

CLIENT :

import java.awt.EventQueue;

import javax.swing.JFrame;

import javax.swing.JLabel;

import java.awt.Font;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.DataInputStream;

import java.io.DataOutputStream;

import java.net.Socket;

import javax.swing.JTextField;

import javax.swing.JButton;

public class sortingclient1 {

private JFrame frame;

private JTextField input;

private JTextField output;

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

sortingclient1 window = new sortingclient1();

window.frame.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

public sortingclient1() {

initialize();

}

private void initialize() {

frame = new JFrame();

frame.setBounds(100, 100, 450, 300);

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.getContentPane().setLayout(null);

JLabel lblNewLabel = new JLabel("Bubble Sort by Harshil ");

lblNewLabel.setFont(new Font("Segoe UI", Font.BOLD, 16));

lblNewLabel.setBounds(120, 13, 178, 33);

frame.getContentPane().add(lblNewLabel);

JLabel lblNewLabel\_1 = new JLabel("Input : ");

lblNewLabel\_1.setBounds(25, 66, 56, 16);

frame.getContentPane().add(lblNewLabel\_1);

JLabel lblNewLabel\_2 = new JLabel("Output : ");

lblNewLabel\_2.setBounds(25, 112, 56, 16);

frame.getContentPane().add(lblNewLabel\_2);

input = new JTextField();

input.setBounds(93, 63, 210, 22);

frame.getContentPane().add(input);

input.setColumns(10);

output = new JTextField();

output.setBounds(93, 109, 210, 22);

frame.getContentPane().add(output);

output.setColumns(10);

JButton sortbtn = new JButton("SORT!");

sortbtn.setBounds(323, 62, 97, 25);

frame.getContentPane().add(sortbtn);

sortbtn.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent arg0) {

try

{

String str;

Socket s=new Socket("localhost",8004);

DataInputStream dis = new DataInputStream(s.getInputStream());

DataOutputStream dout = new DataOutputStream(s.getOutputStream());

str=input.getText();

dout.writeUTF(str);

String s2=new String();

s2 = (String)dis.readUTF();

output.setText(s2);

s.close();

}

catch(Exception e)

{

e.printStackTrace();

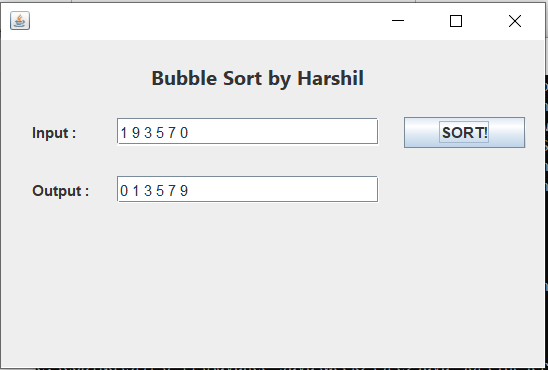
}

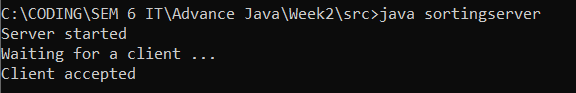
}

});

}

}





Practical 2.3:Implement a basic web server which will serve html files. Use browser to send request to server and server returns requested html page. (Note: Don’t create client program. Use browser as a client program).

SERVER :

import java.io.BufferedReader;

import java.io.File;

import java.io.FileReader;

import java.io.IOException;

import java.net.ServerSocket;

import java.net.Socket;

import java.util.Date;

public class Browserconnection

{

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

{

ServerSocket server2 = new ServerSocket(4488);

System.out.println("Listening for connection on port "+server2.getLocalPort());

while (true)

{

try (Socket socket = server2.accept())

{

File file = new File("C:\\CODING\\SEM 6 IT\\WTP\\WT\\Practical 1\\Practical\_1.html");

BufferedReader br = new BufferedReader(new FileReader(file));

String st;

String forward=new String();

while ((st = br.readLine()) != null)

forward=forward+ st;

String httpResponse = "HTTP/1.1 200 OK\r\n\r\n" + forward;

socket.getOutputStream().write(httpResponse.getBytes("UTF-8"));

}

}

}

}

