**NITHIN RAJ**

ROLL NO: MCA 227 | SEMESTER-II

**OBJECT ORIENTED PROGRAMMING LAB**

LAB RECORD (CO5, CO6)

**COURSE OUTCOME 5**

**PROGRAM NO: 1**

**AIM:**

To write a program to draw Circle, Rectangle, Line in Applet.

**ALGORITHM:**

Step.1: Start the program.

Step.2: Define a class ‘CO5Q1’ that extends Applet class.

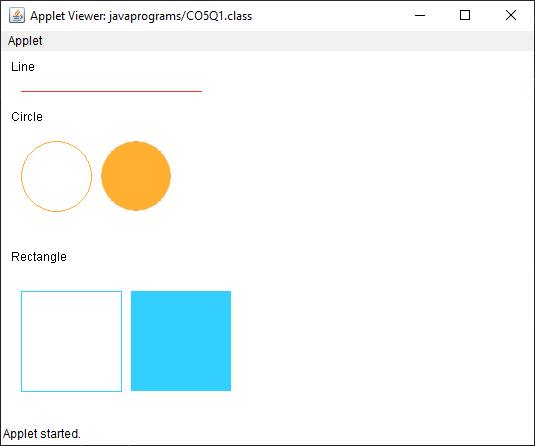
Step.3: Draw a line, rectangle and circle using drawLine, drawRect and drawOval methods of Graphics class respectively.

Step.4: Stop the program.

**PROGRAM CODE:**

|  |  |
| --- | --- |
| CO5Q1.java | //Program to draw Circle, Rectangle, Line in Applet.  package javaprograms;  import java.applet.\*;  import java.awt.\*;  public class CO5Q1 extends Applet  {  public void paint(Graphics g)  {  g.drawString("Line",10,20);  g.setColor(Color.decode("#FF3333"));  g.drawLine(20, 40, 200, 40);  g.setColor(Color.black);  g.drawString("Circle", 10, 70);  g.setColor(Color.decode("#FFB033"));  g.drawOval(20, 90, 70, 70);  g.fillOval(100, 90, 70, 70);  g.setColor(Color.black);  g.drawString("Rectangle", 10, 210);  g.setColor(Color.decode("#33D0FF"));  g.drawRect(20, 240, 100, 100);  g.fillRect(130, 240, 100, 100);  }  } |

**OUTPUT:**



**RESULT:**

The program is successfully executed and the output is verified.

**PROGRAM NO: 2**

**AIM:**

To write a program to find maximum of three numbers using AWT.

**ALGORITHM:**

Step.1: Start the program.

Step.2: Define a class ‘CO5Q2’ that extends Applet class and implements ActionListener interface.

Step.3: Using TextField class object, construct the required no. of textfields wide enough to hold the values entered by the user.

Step.4: Using Button class object, construct a labeled button that sends an instance of ActionEvent.

Step.5: Call addActionListener() method to send events from the button to the new listener.

Step.6: Get the string values from textfields and then parse them as integers.

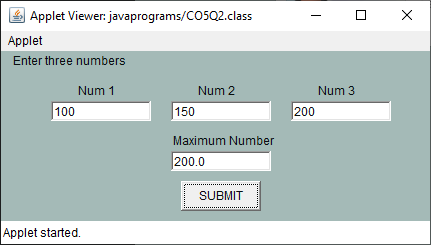
Step.7: Compare each value using if-else statements to find the maximum value and set the result accordingly.

Step.8: Stop the program.

**PROGRAM CODE:**

|  |  |
| --- | --- |
| CO5Q2.java | package javaprograms;  //Program to find maximum of three numbers using AWT.  import java.applet.Applet;  import java.awt.\*;  import java.awt.event.ActionEvent;  import java.awt.event.ActionListener;  public class CO5Q2 extends Applet implements ActionListener  {  TextField t1,t2,t3,result;  Button result\_button;  Label l1,l2,l3,result\_label,title;  public void init()  {  setLayout(null);  setBackground(Color.decode("#A4BAB7"));  setForeground(Color.decode("#080F0F"));  t1 = new TextField();  t2 = new TextField();  t3 = new TextField();  result = new TextField();  l1 = new Label("Num 1");  l2 = new Label("Num 2");  l3 = new Label("Num 3");  title = new Label("Enter three numbers");  result\_label = new Label("Maximum Number");  result\_button = new Button("SUBMIT");  title.setBounds(10,0,200,20);  l1.setBounds(75,30,100,20);  t1.setBounds(50,50,100,20);  l2.setBounds(195,30,100,20);  t2.setBounds(170,50,100,20);  l3.setBounds(315,30,100,20);  t3.setBounds(290,50,100,20);  result\_label.setBounds(170,80,110,20);  result.setBounds(170,100,100,20);  result\_button.setBounds(180,130,80,30);  add(t1);  add(t2);  add(t3);  add(result\_label);  add(result);  add(result\_button);  add(l1);  add(l2);  add(l3);  add(title);  result\_button.addActionListener(this);  }  @Override  public void actionPerformed(ActionEvent e) {  // throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.  float num1,num2,num3;  String str;  str = t1.getText();  num1 = Float.parseFloat(str);  str = t2.getText();  num2 = Float.parseFloat(str);  str = t3.getText();  num3 = Float.parseFloat(str);  if(num1>num2 && num1>num3)  result.setText(num1+"");  else if(num2>num1 && num2>num3)  result.setText(num2+"");  else  result.setText(num3+"");  }  } |

**OUTPUT:**



**RESULT:**

The program is successfully executed and the output is verified.

**PROGRAM NO: 3**

**AIM:**

To find the percentage of marks obtained by a student in 5 subjects. Display a happy face if he secures above 50% or a sad face if otherwise.

**ALGORITHM:**

Step.1: Start the program.

Step.2: Define a class ‘CO5Q3’ that extends Applet class and implements ActionListener interface.

Step.3: Using TextField class object, construct textfields to receive marks of 5 subjects from the user.

Step.4: Using Button class object, construct a labeled button that sends an instance of ActionEvent.

Step.5: Call addActionListener() method to send events from the button to the new listener.

Step.6: Get the string values from textfields and then parse them as float values.

Step.7: Calculate the percentage:

Percent = ((mark1+mark2+mark3+mark4+mark5)\*100)/500

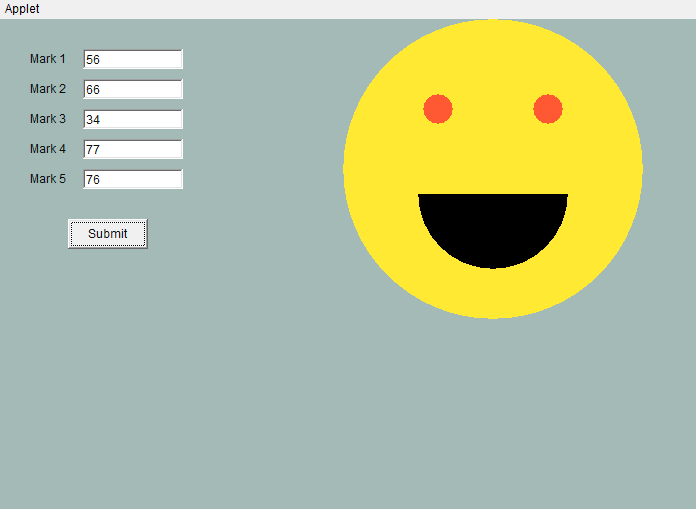
Step.8: Define a paint() method that contains functions from Graphics class to display a happy face if student secures above 50% or a sad face if otherwise

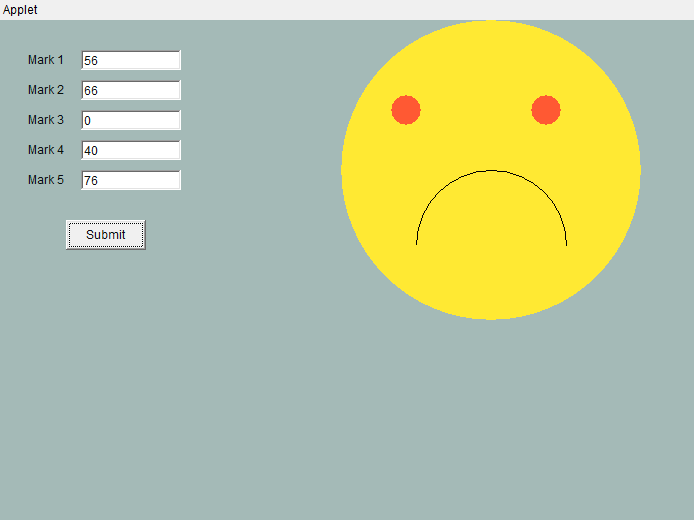
Step.9: Stop the program.

**PROGRAM CODE:**

|  |  |
| --- | --- |
| CO5Q3.java | //Find the percentage of marks obtained by a student in 5 subjects. Display a happy face if  //he secures above 50% or a sad face if otherwise.  package javaprograms;  import java.applet.Applet;  import java.awt.\*;  import java.awt.event.ActionEvent;  import java.awt.event.ActionListener;  public class CO5Q3 extends Applet implements ActionListener  {  TextField t1,t2,t3,t4,t5;  Button submit;  Label l1,l2,l3,l4,l5;  float num1,num2,num3,num4,num5,sum,percentage;  String str;  @Override  public void init()  {  setLayout(null);  setSize(700, 500);  setBackground(Color.decode("#A4BAB7"));  setForeground(Color.decode("#080F0F"));  t1 = new TextField();  t2 = new TextField();  t3 = new TextField();  t4 = new TextField();  t5 = new TextField();  l1 = new Label("Mark 1");  l2 = new Label("Mark 2");  l3 = new Label("Mark 3");  l4 = new Label("Mark 4");  l5 = new Label("Mark 5");  submit = new Button("Submit");  l1.setBounds(30,30,100,20);  t1.setBounds(85,30,100,20);  l2.setBounds(30,60,100,20);  t2.setBounds(85,60,100,20);  l3.setBounds(30,90,100,20);  t3.setBounds(85,90,100,20);  l4.setBounds(30,120,100,20);  t4.setBounds(85,120,100,20);  l5.setBounds(30,150,100,20);  t5.setBounds(85,150,100,20);  submit.setBounds(70, 200, 80, 30);  add(t1);  add(t2);  add(t3);  add(t4);  add(t5);  add(l1);  add(l2);  add(l3);  add(l4);  add(l5);  add(submit);  submit.addActionListener(this);  }  @Override  public void actionPerformed(ActionEvent e) {    str = t1.getText();  num1 = Float.parseFloat(str);  str = t2.getText();  num2 = Float.parseFloat(str);  str = t3.getText();  num3 = Float.parseFloat(str);  str = t4.getText();  num4 = Float.parseFloat(str);  str = t5.getText();  num5 = Float.parseFloat(str);  sum = num1+num2+num3+num4+num5;  percentage = (sum\*100)/500;  System.out.println(sum+" "+percentage);  repaint();  }  @Override  public void paint(Graphics g)  {  if(percentage > 50.0){  g.setColor(Color.decode("#FFE933"));  g.fillOval(345,0,300,300);  g.setColor(Color.decode("#FF5933"));  g.fillOval(425,75,30,30);  g.fillOval(535,75,30,30);  g.setColor(Color.black);  g.fillArc (420,100,150,150,0,-180);  }  else {  g.setColor(Color.decode("#FFE933"));  g.fillOval(345,0,300,300);  g.setColor(Color.decode("#FF5933") );  g.fillOval(395,75,30,30);  g.fillOval(535,75,30,30);  g.setColor(Color.black);  g.drawArc(420,150,150,150,0,180);  }  }  } |

**OUTPUT:**





**RESULT:**

The program is successfully executed and the output is verified.

**PROGRAM NO: 4**

**AIM:**

Using 2D graphics commands in an Applet, construct a house. On mouse click event, change the color of the door from blue to red.

**ALGORITHM:**

Step.1: Start the program.

Step.2: Define a class ‘CO5Q4 ‘ that extends Applet and implements MouseListener.

Step.3: Define methods to add MouseListener to the panel.

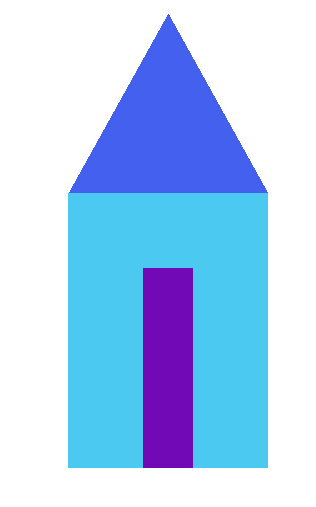
Step.4: Using getX() and getY() methods, get the coordinates of the door to repaint when the MousePressed event occurs.

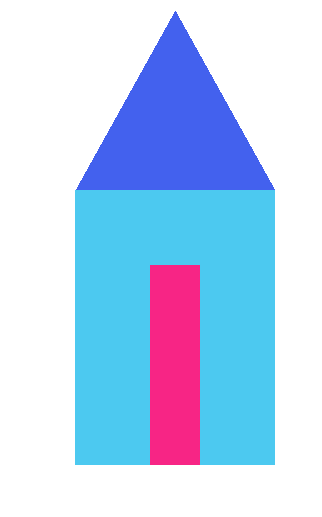
Step.5: Stop the program.

**PROGRAM CODE:**

|  |  |
| --- | --- |
| CO5Q4.java | //Using 2D graphics commands in an Applet, construct a house. On mouse click event,  //change the color of the door from blue to red.  package javaprograms;  import java.applet.Applet;  import java.awt.\*;  import java.awt.event.MouseEvent;  import java.awt.event.MouseListener;  public class CO5Q4 extends Applet implements MouseListener  {  int a, b;  public void init() {  addMouseListener(this);  }  public void paint (Graphics g) {    int x[]= {200,300,400};  int y[]= {200,20,200};    g.setColor(Color.decode("#4361EE"));  g.fillPolygon(x,y,3);  g.setColor(Color.decode("#4CC9F0"));  g.fillRect(200, 200, 200, 275);  g.setColor(Color.decode("#7209B7"));  g.fillRect(275, 275, 50, 200);    if(a>269 && a<321 && b>275 && b<470) {  g.setColor(Color.decode("#F72585"));  g.fillRect(275, 275, 50, 200);  }    }  @Override  public void mouseClicked(MouseEvent e) {    }  @Override  public void mousePressed(MouseEvent e) {  a=e.getX();  b=e.getY();  repaint();  }  @Override  public void mouseReleased(MouseEvent e) {    }  @Override  public void mouseEntered(MouseEvent e) {    }  @Override  public void mouseExited(MouseEvent e) {    }    } |

**OUTPUT:**





**RESULT:**

The program is successfully executed and the output is verified.

**PROGRAM NO: 5**

**AIM:**

To implement a simple calculator using AWT components.

**ALGORITHM:**

Step.1: Start the program.

Step.2: Define a class ‘CO5Q5 ‘ that extends Frame and implements ActionListener interface.

Step.3: Using TextField class object, construct the required no. of textfields wide enough to hold the values entered by the user.

Step.4: Using Label class object, construct and provide the appropriate labels.

Step.5: Using Button class object, construct labeled buttons that send the instances of ActionEvent.

Step.6: Call addActionListener() method to send events from the button to the new listener.

Step.7: Get the string values from textfields and then parse them as integers.

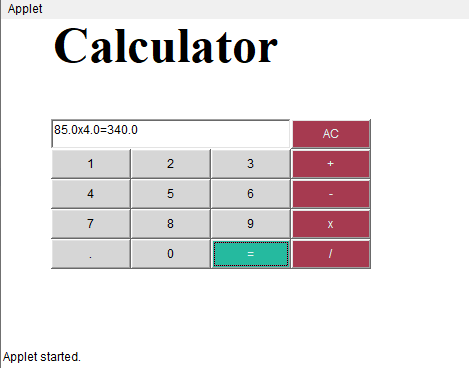
Step.8: Perform various methods to add, subtract, multiply and divide those integers.

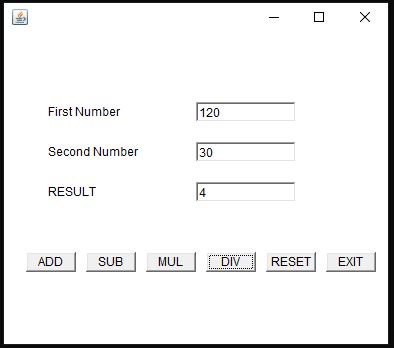
Step.8: Stop the program.

**PROGRAM CODE:**

|  |  |
| --- | --- |
| CO5Q5.java | package javaprograms;  //Implement a simple calculator using AWT components.  import java.applet.Applet;  import java.awt.\*;  import java.awt.event.ActionEvent;  import java.awt.event.ActionListener;  public class CO5Q5 extends Applet implements ActionListener  {  TextField t1;  Button b0,b1,b2,b3,b4,b5,b6,b7,b8,b9,button\_add,button\_sub,button\_mul,button\_div,button\_equal,button\_clear,button\_dot;  Label l1;  @Override  public void init()  {  setLayout(null);  l1 = new Label("Calculator");  t1 = new TextField();  b0 = new Button("0");  b1 = new Button("1");  b2 = new Button("2");  b3 = new Button("3");  b4 = new Button("4");  b5 = new Button("5");  b6 = new Button("6");  b7 = new Button("7");  b8 = new Button("8");  b9 = new Button("9");  button\_add = new Button("+");  button\_sub = new Button("-");  button\_mul = new Button("x");  button\_div = new Button("/");  button\_equal = new Button("=");  button\_clear = new Button("AC");  button\_dot = new Button(".");  l1.setBounds(50,0,2000,50);  Font myFont = new Font("Serif",Font.BOLD,50);  l1.setFont(myFont);  t1.setBounds(50, 100, 240, 30);  button\_clear.setBounds(290, 100, 80, 30);  b1.setBounds(50,130,80,30);  b2.setBounds(130,130,80,30);  b3.setBounds(210,130,80,30);  button\_add.setBounds(290,130,80,30);  b4.setBounds(50,160,80,30);  b5.setBounds(130,160,80,30);  b6.setBounds(210,160,80,30);  button\_sub.setBounds(290,160,80,30);  b7.setBounds(50,190,80,30);  b8.setBounds(130,190,80,30);  b9.setBounds(210,190,80,30);  button\_mul.setBounds(290,190,80,30);  button\_dot.setBounds(50,220,80,30);  b0.setBounds(130,220,80,30);  button\_equal.setBounds(210,220,80,30);  button\_div.setBounds(290,220,80,30);  button\_equal.setForeground(Color.decode("#E1F4F1"));  button\_equal.setBackground(Color.decode("#25BA9F"));  b0.setBackground(Color.decode("#D6D6D6"));  b1.setBackground(Color.decode("#D6D6D6"));  b2.setBackground(Color.decode("#D6D6D6"));  b3.setBackground(Color.decode("#D6D6D6"));  b4.setBackground(Color.decode("#D6D6D6"));  b5.setBackground(Color.decode("#D6D6D6"));  b6.setBackground(Color.decode("#D6D6D6"));  b7.setBackground(Color.decode("#D6D6D6"));  b8.setBackground(Color.decode("#D6D6D6"));  b9.setBackground(Color.decode("#D6D6D6"));  button\_dot.setBackground(Color.decode("#D6D6D6"));  button\_clear.setBackground(Color.decode("#A63A50"));  button\_add.setBackground(Color.decode("#A63A50"));  button\_sub.setBackground(Color.decode("#A63A50"));  button\_mul.setBackground(Color.decode("#A63A50"));  button\_div.setBackground(Color.decode("#A63A50"));  button\_add.setForeground(Color.decode("#E1F4F1"));  button\_sub.setForeground(Color.decode("#E1F4F1"));  button\_mul.setForeground(Color.decode("#E1F4F1"));  button\_div.setForeground(Color.decode("#E1F4F1"));  button\_clear.setForeground(Color.decode("#E1F4F1"));  add(l1);  add(t1);  add(b1);  add(b2);  add(b3);  add(b4);  add(b5);  add(b6);  add(b7);  add(b8);  add(b9);  add(b0);  add(button\_dot);  add(button\_equal);  add(button\_add);  add(button\_clear);  add(button\_sub);  add(button\_mul);  add(button\_div);  b0.addActionListener(this);  b1.addActionListener(this);  b2.addActionListener(this);  b3.addActionListener(this);  b4.addActionListener(this);  b5.addActionListener(this);  b6.addActionListener(this);  b7.addActionListener(this);  b8.addActionListener(this);  b9.addActionListener(this);  button\_add.addActionListener(this);  button\_sub.addActionListener(this);  button\_mul.addActionListener(this);  button\_div.addActionListener(this);  button\_dot.addActionListener(this);  button\_clear.addActionListener(this);  button\_equal.addActionListener(this);  }  int op = 5;  boolean flag=false;  String str1,str2,str\_result;  float num1=0,num2=0,result;  @Override  public void actionPerformed(ActionEvent e) {  // throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.  if(e.getSource()==b0)  if(flag!=true)  {  str1 = t1.getText();  t1.setText(str1+"0");  }  else  {  str2 = t1.getText();  t1.setText(str2+"0");  }  if(e.getSource()==b1)  if(flag!=true)  {  str1 = t1.getText();  t1.setText(str1+"1");  }  else  {  str2 = t1.getText();  t1.setText(str2+"1");  }  if(e.getSource()==b2)  if(flag!=true)  {  str1 = t1.getText();  t1.setText(str1+"2");  }  else  {  str2 = t1.getText();  t1.setText(str2+"2");  }  if(e.getSource()==b3)  if(flag!=true)  {  str1 = t1.getText();  t1.setText(str1+"3");  }  else  {  str2 = t1.getText();  t1.setText(str2+"3");  }  if(e.getSource()==b4)  if(flag!=true)  {  str1 = t1.getText();  t1.setText(str1+"4");  }  else  {  str2 = t1.getText();  t1.setText(str2+"4");  }  if(e.getSource()==b5)  if(flag!=true)  {  str1 = t1.getText();  t1.setText(str1+"5");  }  else  {  str2 = t1.getText();  t1.setText(str2+"5");  }  if(e.getSource()==b6)  if(flag!=true)  {  str1 = t1.getText();  t1.setText(str1+"6");  }  else  {  str2 = t1.getText();  t1.setText(str2+"6");  }  if(e.getSource()==b7)  if(flag!=true)  {  str1 = t1.getText();  t1.setText(str1+"7");  }  else  {  str2 = t1.getText();  t1.setText(str2+"7");  }  if(e.getSource()==b8)  if(flag!=true)  {  str1 = t1.getText();  t1.setText(str1+"8");  }  else  {  str2 = t1.getText();  t1.setText(str2+"8");  }  if(e.getSource()==b9)  if(flag!=true)  {  str1 = t1.getText();  t1.setText(str1+"9");  }  else  {  str2 = t1.getText();  t1.setText(str2+"9");  }  if(e.getSource()==button\_dot)  if(flag!=true)  {  str1 = t1.getText();  t1.setText(str1+".");  }  else  {  str2 = t1.getText();  t1.setText(str2+".");  }  if(e.getSource()==button\_clear)  {  t1.setText("");  }  if(e.getSource()==button\_add)  {  flag=true;  op=1;  num1=Float.parseFloat(t1.getText());  t1.setText("");  }  if(e.getSource()==button\_sub)  {  flag=true;  op=2;  num1=Float.parseFloat(t1.getText());  t1.setText("");  }  if(e.getSource()==button\_mul)  {  flag=true;  op=3;  num1=Float.parseFloat(t1.getText());  t1.setText("");  }  if(e.getSource()==button\_div)  {  flag=true;  op=4;  num1=Float.parseFloat(t1.getText());  t1.setText("");  }  if(e.getSource()==button\_equal)  {  num2 = Float.parseFloat(t1.getText());  switch(op)  {  case 1: result = num1 + num2;  str\_result = String.valueOf(num1);  str\_result += "+";  str\_result += String.valueOf(num2);  str\_result += "=";  str\_result += String.valueOf(result);  t1.setText(str\_result);  break;  case 2: result = num1 - num2;  t1.setText(String.valueOf(result));  str\_result = String.valueOf(num1);  str\_result += "-";  str\_result += String.valueOf(num2);  str\_result += "=";  str\_result += String.valueOf(result);  t1.setText(str\_result);  break;  case 3: result = num1 \* num2;  str\_result = String.valueOf(num1);  str\_result += "x";  str\_result += String.valueOf(num2);  str\_result += "=";  str\_result += String.valueOf(result);  t1.setText(str\_result);  break;  case 4: result = num1 / num2;  str\_result = String.valueOf(num1);  str\_result += "/";  str\_result += String.valueOf(num2);  str\_result += "=";  str\_result += String.valueOf(result);  t1.setText(str\_result);  break;  default:t1.setText("Error");  }  }  }  } |

**OUTPUT:**





**RESULT:**

The program is successfully executed and the output is verified.

**PROGRAM NO: 6**

**AIM:**

To develop a program that has a Choice component which contains the names of shapes such as rectangle, triangle, square and circle. Draw the corresponding shapes for given parameters as per user’s choice.

**ALGORITHM:**

Step.1: Start the program.

Step.2: Define an interface ‘CO5Q7’ that extends Applet class and implements ItemListener interface.

Step.3: Declare a new constructor of the Choice class to create an empty Choice menu.

Step.4: Use add() method to include items in the menu.

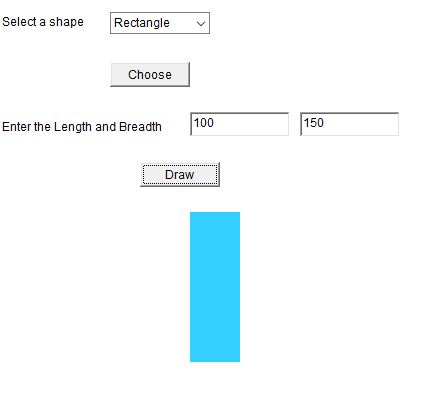
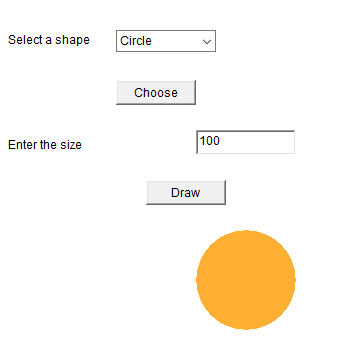
Step.5: Using getSelectedItem() method, get the item chosen by the user from the menu and repaint accordingly.

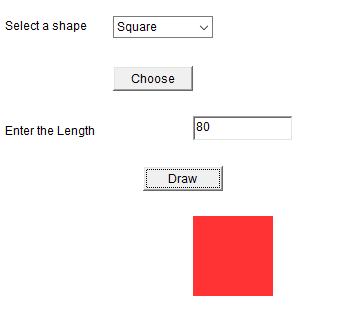
Step.6: Stop the program.

**PROGRAM CODE:**

|  |  |
| --- | --- |
| CO5Q7.java | //Develop a program to handle all mouse events and window events  package javaprograms;  import java.applet.Applet;  import java.awt.Graphics;  import java.awt.event.MouseEvent;  import java.awt.event.MouseListener;  import java.awt.event.MouseMotionListener;  public class CO5Q7 extends Applet implements MouseListener,MouseMotionListener  {  String s = "";  int x=0,y=0;  @Override  public void init()  {  addMouseListener(this);  addMouseMotionListener(this);  }  @Override  public void mouseClicked(MouseEvent e) {  s = "Mouse Clicked";  repaint();  }  @Override  public void mousePressed(MouseEvent e) {  s = "Mouse Pressed";  repaint();  }  @Override  public void mouseReleased(MouseEvent e) {  s = "Mouse Released";  repaint();  }  @Override  public void mouseEntered(MouseEvent e) {  s = "Mouse Entered";  repaint();  }  @Override  public void mouseExited(MouseEvent e) {  s = "Mouse Exited";  repaint();  }  @Override  public void mouseDragged(MouseEvent e) {  s = "Mouse Dragged";  x = e.getX();  y = e.getY();  showStatus("Mouse Position: X="+x+" Y="+y);  repaint();  }  @Override  public void mouseMoved(MouseEvent e) {  s = "Mouse Moved";  x = e.getX();  y = e.getY();  showStatus("Mouse Position: X="+x+" Y="+y);  repaint();  }  @Override  public void paint(Graphics g)  {  g.drawString("Handling Mouse Events",30,20);  g.drawString(s,60,40);  }    } |

**OUTPUT:**





**RESULT:**

The program is successfully executed and the output is verified.

**PROGRAM NO: 8**

**AIM:**

To develop a program to handle Key events.

**ALGORITHM:**

Step.1: Start the program.

Step.2: Define a class CO5Q8 that extends Applet and implements KeyListener.

Step.3: Define methods to add KeyListener to the panel which will have the following methods:

void keyTyped(KeyEvent e) – Invoked when a key has been typed.

void keyPressed(KeyEvent e) - Invoked when a key has been pressed.

void keyReleased(KeyEvent e) - Invoked when a key has been released.

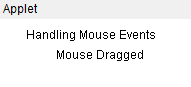
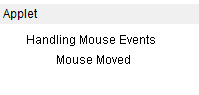
Step.4: Using **getKeyChar(),** get the unicode and character representation of the key pressed. Use them to display the necessary message in the output.

Step.5: Stop the program.

**PROGRAM CODE:**

|  |  |
| --- | --- |
| CO5Q8.java | //Develop a program to handle Key events  package javaprograms;  import java.applet.Applet;  import java.awt.Graphics;  import java.awt.TextField;  import java.awt.event.KeyEvent;  import java.awt.event.KeyListener;  public class CO5Q8 extends Applet implements KeyListener  {  String msg="";  TextField t1;  @Override  public void init()  {  setLayout(null);  addKeyListener(this);  t1 = new TextField();  t1.setBounds(50,50,100,20);  t1.addKeyListener(this);  add(t1);  }  @Override  public void keyTyped(KeyEvent e) {  msg+= e.getKeyChar();  repaint();  }  @Override  public void keyPressed(KeyEvent e) {  showStatus("Key pressed");  }  @Override  public void keyReleased(KeyEvent e) {  showStatus("Key released");  }  @Override  public void paint(Graphics g)  {    }  } |

**OUTPUT:**



**RESULT:**

The program is successfully executed and the output is verified.

**COURSE OUTCOME 6**

**PROGRAM NO: 1**

**AIM:**

To write program to list the sub directories and files in a given directory and also search for a file name.

**ALGORITHM:**

Step.1: Start the program.

Step.2: Create a class named ‘CO6Q1’ that implements FilenameFilter interface.

#### Step.3: Create an object of the class File to to initialize its constructor with the file source.

Step.4: Using list(), get the names of all the files present in the directory.

Step.5: Create an object for the FileNameFilter [interface](https://www.journaldev.com/1601/interface-in-java) that contains the method Boolean accept ( File dir, String name) to test if a specified file should be included in the file list or not. .

Step.6: Filter accordingly and store the file names to the list.

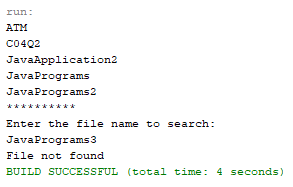
Step.7: Display the list.

Step.8: Stop the program.

**PROGRAM CODE:**

|  |  |
| --- | --- |
| CO6Q1.java | //Program to list the sub directories and files in a given directory and also search for a file  //name.  package javaprograms;  import java.io.File;  import java.util.Scanner;  public class CO6Q1 {  public static void main(String[] args) {  int flag=0;  File f = new File("../");  Scanner sc = new Scanner(System.in);  String name,filename;  String[] filelist = f.list();  try{  for(String str: filelist)  System.out.println(str);  System.out.println("\*\*\*\*\*\*\*\*\*\*");  System.out.println("Enter the file name to search:");  name = sc.nextLine();  for(int i =0; i<f.length();i++)  {  filename = filelist[i];  if(filename.equals(name))  {  flag=1;  break;  }  else  flag=0;  }  }  catch(Exception e)  {  if(flag==1)  System.out.println("File found");  else  System.out.println("File not found");  }  }  } |

**OUTPUT:**



**RESULT:**

The program is successfully executed and the output is verified.

**PROGRAM NO: 2**

**AIM:**

To write a program to write to a file, then read from the file and display the contents on the console.

**ALGORITHM:**

Step.1: Start the program.

Step.2: Create a class named ‘CO6Q2’.

#### Step.3: Create an object of the class File to initialize its constructor with the file source.

Step.4: Create and use an object of the FileWriter class to write the file.

Step.5: Create and use an object of the BufferedReader class to read the stream of characters the specified file.

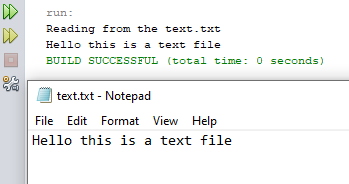
Step.6: Display the contents read from the file on the console.

Step.7: Stop the program.

**PROGRAM CODE:**

|  |  |
| --- | --- |
| CO6Q2.java | //Write a program to write to a file, then read from the file and display the contents on the  //console.  package javaprograms;  import java.io.\*;  public class CO6Q2 {  public static void main(String[] args) {  try  {  String str;  FileWriter fw = new FileWriter("text.txt",true);  fw.write("Hello this is a text file");  fw.close();  FileReader fr = new FileReader("text.txt");  BufferedReader br = new BufferedReader(fr);  System.out.println("Reading from the text.txt");  while((str=br.readLine())!=null)  System.out.println(str);    }  catch(Exception e)  {    }  }  } |

**OUTPUT:**



**RESULT:**

The program is successfully executed and the output is verified.

**PROGRAM NO: 3**

**AIM:**

To write a program to copy one file to another.

**ALGORITHM:**

Step.1: Start the program.

Step.2: Create a class named ‘CO6Q3’.

Step.3: Create and use an object of the BufferedReader class to read the stream of characters from the specified file.

Step.4: Create and use an object of the FileWriter class to write the stream of characters read by the BufferedReader to the file.

while ((s = br.readLine()) != null) {

fw.write(s);

}

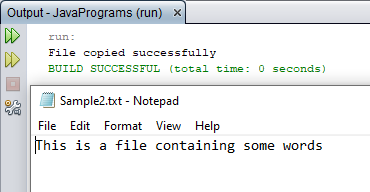
Step.6: Display the appropriate message on the console.

Step.7: Stop the program.

**PROGRAM CODE:**

|  |  |
| --- | --- |
| CO6Q3.java | //Write a program to copy one file to another.  package javaprograms;  import java.io.\*;  import java.util.\*;  public class CO6Q3 {  public static void main(String[] args) {  try {  String str;  FileReader fr = new FileReader("Sample1.txt");  BufferedReader br = new BufferedReader(fr);  FileWriter fw = new FileWriter("Sample2.txt",true);  BufferedWriter bw = new BufferedWriter(fw);  while((str=br.readLine())!=null)  {  fw.write(str);  fw.flush();  }  System.out.println("File copied successfully");  }  catch (Exception ex) {  System.out.println(ex);  }  }  } |

**OUTPUT:**



**RESULT:**

The program is successfully executed and the output is verified.

**PROGRAM NO: 4**

**AIM:**

To write a program that reads from a file having integers. Copy even numbers and odd numbers to separate files.

**ALGORITHM:**

Step.1: Start the program.

Step.2: Create a class named ‘CO6Q4’.

Step.3: Create an object of the class File to initialize its constructor with the given file.

Step.4: Get user inputs via the console, for the integers to be inserted into the file.

Step.6: Using an object of the FileWriter class, write those integers into the file.

Step.7: Using objects of the FileOutputStream class, create two separate files to store even and odd integers respectively and copy the integers accordingly to separate files just created.

while((i=r.read()) != -1)

{

if(i%2==0)

fo1.write(i);

else

fo2.write(i);

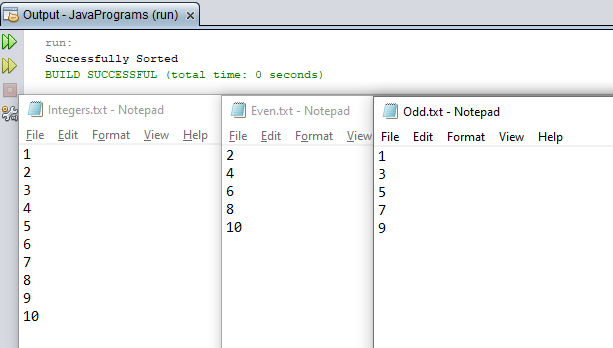
}

Step.8: Stop the program.

**PROGRAM CODE:**

|  |  |
| --- | --- |
| CO6Q4.java | //Write a program that reads from a file having integers. Copy even numbers and odd  //numbers to separate files.  package javaprograms;  import java.io.\*;  public class CO6Q4 {  public static void main(String[] args) {  try {  String str;  int n;  FileReader fr = new FileReader("Integers.txt");  BufferedReader br = new BufferedReader(fr);  FileWriter fileEven = new FileWriter("Even.txt",true);  FileWriter fileOdd = new FileWriter("Odd.txt",true);  BufferedWriter bwEven = new BufferedWriter(fileEven);  BufferedWriter bwOdd = new BufferedWriter(fileOdd);  while((str=br.readLine())!=null)  {  n = Integer.parseInt(str);  if(n%2==0)  {  bwEven.write(String.valueOf(n));  bwEven.write("\n");  bwEven.flush();  }  else  {  bwOdd.write(String.valueOf(n));  bwOdd.write("\n");  bwOdd.flush();  }  }  System.out.println("Successfully Sorted");  } catch (Exception e) {  }  }  } |

**OUTPUT:**



**RESULT:**

The program is successfully executed and the output is verified.

**PROGRAM NO: 5**

**AIM:**

To implement client server communication using Socket – TCP/IP.

**ALGORITHM:**

Step.1: Start the program.

Step.2: To create the Client application, create an instance of ClientSocket class.

2.1: Initiate connection to the server using hostname and a port number.

2.2: Send data to the server using an OutputStream object.

2.3: Read data from the server using an InputStream object.

2.4: Close the connection.

Step.3: To create the Server application, create an instance of ServerSocket class.

3.1: Wait till a connection is established.

Socket s = ss.accept();

3.2: Receive data from the client using an InputStream object.

3.3: Send data to the client using an OutputStream object.

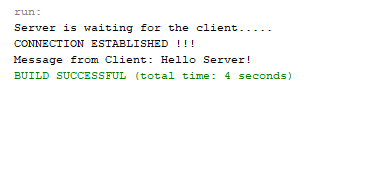
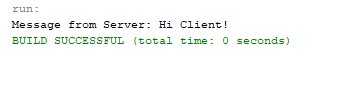
3.4: Close the connection.

Step.4: Stop the program.

**PROGRAM CODE:**

|  |  |
| --- | --- |
| CO6Q5Client.java | //Client server communication using Socket – TCP/IP  package javaprograms;  import java.net.\*;  import java.io.\*;  public class CO6Q5Client {  public static void main(String args[]) throws Exception{  try {  Socket s = new Socket ("localhost", 5555);  PrintWriter pw = new PrintWriter(s.getOutputStream(), true);  pw.println("Hello Server!");  InputStreamReader isr = new InputStreamReader(s.getInputStream());  BufferedReader br = new BufferedReader(isr);  String str= br.readLine();  System.out.println("Message from Server: "+str);  pw.close();  s.close();  }  catch(Exception e)  {  System.out.println("An error occured..." +e);  }  }  } |
| CO6Q5Server.java | //Client server communication using Socket – TCP/IP  package javaprograms;  import java.net.\*;  import java.io.\*;  public class CO6Q5Server {  public static void main(String[] args) throws Exception {  try {  ServerSocket ss = new ServerSocket(5555);  System.out.println("Server is waiting for the client.....");  Socket s = ss.accept();  System.out.println("CONNECTION ESTABLISHED !!!");  InputStreamReader isr = new InputStreamReader(s.getInputStream());  BufferedReader br = new BufferedReader(isr);  String str = br.readLine();  System.out.println("Message from Client: "+str);  PrintWriter pw = new PrintWriter(s.getOutputStream(), true);  pw.println("Hi Client!");  pw.close();  }  catch(Exception e) {  System.out.println("An error occured.."+e);  }  }  } |

**OUTPUT:**



**RESULT:**

The program is successfully executed and the output is verified.

**PROGRAM NO: 6**

**AIM:**

To implement client server communication using DatagramSocket - UDP.

**ALGORITHM:**

Step.1: Start the program.

Step.2: Create the Client application:

2.1: Create a DatagramSocket object to carry the packet to the destination and to receive it whenever the server sends any data.

2.2: Create the packet for sending/receiving data via a DatagramSocket.

DatagramPacket(byte buf[], int length, InetAddress inetaddress, int port):-

2.3: **Invoke a send() or receive() call on socket object.**

2.4: Close the connection.

Step.3: Create the Server application:

3.1: Create a DatagramSocket object to listen at the port specified.

3.2: Create the packet for sending/receiving data via a DatagramSocket.

3.3: **Invoke a send() or receive() call on socket object.**

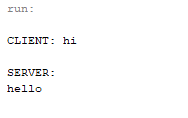
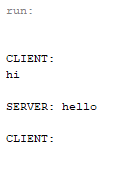
3.4: Close the connection.

Step.4: Stop the program.

**PROGRAM CODE:**

|  |  |
| --- | --- |
| CO6Q6Client.java | //Client Server communication using DatagramSocket - UDP  package javaprograms;  import java.io.IOException;  import java.net.InetAddress;  import java.net.SocketException;  import java.net.DatagramSocket;  import java.net.DatagramPacket;  import java.util.Scanner;  public class CO6Q6Client {  public static void main(String[] args) throws SocketException, IOException {  InetAddress IP = InetAddress.getByName("localhost");  DatagramSocket csocket = new DatagramSocket();  while(true) {  byte[] sendbuffer = new byte[1024];  byte[] receivebuffer = new byte[1024];  System.out.println("\n\nCLIENT: ");  Scanner sc = new Scanner(System.in);  String clientData = sc.nextLine();  sendbuffer = clientData.getBytes();    DatagramPacket sndpkt = new DatagramPacket(sendbuffer, sendbuffer.length, IP, 9876);  csocket.send(sndpkt);;    if(clientData.equalsIgnoreCase("Bye")) {  System.out.println("Connection dropped by Client. .!");  break;  }  DatagramPacket rcvpkt = new DatagramPacket(receivebuffer, receivebuffer.length);  csocket.receive(rcvpkt);  String serverData = new String(rcvpkt.getData());  System.out.print("\nSERVER: "+ serverData);  }  csocket.close();  }  } |
| CO6Q6Server.java | package javaprograms;  import java.io.IOException;  import java.net.InetAddress;  import java.net.SocketException;  import java.net.DatagramSocket;  import java.net.DatagramPacket;  import java.util.Scanner;  public class CO6Q6Server {  public static void main(String[] args) throws SocketException, IOException {  DatagramSocket sSocket = new DatagramSocket(9876);  while(true) {  byte[] sendbuffer = new byte[1024];  byte[] receivebuffer = new byte[1024];  DatagramPacket rcvdpkt = new DatagramPacket(receivebuffer, receivebuffer.length);  sSocket.receive(rcvdpkt);  InetAddress IP = rcvdpkt.getAddress();  int portNo = rcvdpkt.getPort();  String clData = new String(rcvdpkt.getData());  System.out.println("\nCLIENT: " + clData);  System.out.println("\nSERVER: ");  Scanner sc = new Scanner(System.in);  String serData = sc.nextLine();  sendbuffer = serData.getBytes();  DatagramPacket sendpkt = new DatagramPacket(sendbuffer, sendbuffer.length, IP, portNo);  sSocket.send(sendpkt);    if(serData.equalsIgnoreCase("Bye")) {  System.out.println("Connection dropped by Server. . !");  break;  }  }  sSocket.close();  }  } |

**OUTPUT:**



**RESULT:**

The program is successfully executed and the output is verified.