**Term-work 2**

**Complete problem definition of the Term-work**

**Create** an appropriate GUI which allows the user to select an item from the menu. When draw menu item is selected, draws the selected shape(Allowed shapes are: Circle, Rectangle & Triangle) in drawing area by getting appropriate dimensions of the selected shape from the user through keyboard entry, using the concept of ABSTRACT CLASS, INHERITANCE and DYNAMIC DISPATCH features of JAVA Programming Language.

The code must be robust for all possible erroneous input conditions, displaying appropriate error messages in message window specially designed for them.

**Expected Learning**: Abstract class, Inheritance, Runtime polymorphism, AWT, Event Handling, Exception Handling

**Source code/Program**

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

import java.applet.\*;

import java.awt.\*;

import java.awt.event.\*;

/\*\*

\*

\* @author kishan

\*/

public class shapeDraw4 extends Applet implements ItemListener, ActionListener {

/\*\*

\* Initialization method that will be called after the applet is loaded into

\* the browser.

\*/

Checkbox circle, square, triangle;

Panel mainPanel, cb, tb;

CardLayout cardL0;

TextField a, b, c, d;

String msg, a1, b1, c1, d1;

int a2, b2, c2, d2;

boolean circ, squr, tri;

String [] xset;

String [] yset;

String [] zset;

int [] xPoints = new int[5];

int [] yPoints = new int [5];

Label xpoints, ypoints;

Label top, left, width, height;

@Override

public void init() {

// TODO start asynchronous download of heavy resources

cardL0 = new CardLayout();

mainPanel = new Panel();

mainPanel.setLayout(cardL0);

circle = new Checkbox("Circle",null,false);

square = new Checkbox("Square",null,false);

triangle = new Checkbox("Triangle",null,false);

cb = new Panel();

cb.add(circle);

cb.add(square);

cb.add(triangle);

top = new Label("TOP ", Label.RIGHT);

left = new Label("LEFT ", Label.RIGHT);

width = new Label("WIDTH ", Label.RIGHT);

height = new Label("HEIGHT ", Label.RIGHT);

xpoints = new Label("xPoints",Label.RIGHT);

ypoints = new Label("yPoints",Label.RIGHT);

a = new TextField(10);

b = new TextField(10);

c = new TextField(10);

d = new TextField(10);

tb = new Panel();

tb.add(xpoints);

tb.add(top);

tb.add(a);

tb.add(ypoints);

tb.add(left);

tb.add(b);

tb.add(width);

tb.add(c);

tb.add(height);

tb.add(d);

mainPanel.add(cb,"checkbox");

mainPanel.add(tb,"TextField");

add(mainPanel);

circle.addItemListener(this);

square.addItemListener(this);

triangle.addItemListener(this);

a.addActionListener(this);

b.addActionListener(this);

c.addActionListener(this);

d.addActionListener(this);

}

@Override

public void itemStateChanged(ItemEvent ie){

circ = circle.getState();

squr = square.getState();

tri = triangle.getState();

if(circ == true || squr == true){

xpoints.setVisible(false);

ypoints.setVisible(false);

cb.setVisible(false);

tb.setVisible(true);

}

if(tri == true){

top.setVisible(false);

left.setVisible(false);

width.setVisible(false);

height.setVisible(false);

cb.setVisible(false);

tb.setVisible(true);

c.setVisible(false);

d.setVisible(false);

}

repaint();

}

@Override

public void actionPerformed(ActionEvent ae){

if(tri == true){

a1 = a.getText();

xset = a1.split(",",-2);

xPoints[0] = Integer.parseInt(xset[0]);

xPoints[1] = Integer.parseInt(xset[1]);

xPoints[2] = Integer.parseInt(xset[2]);

b1 = b.getText();

yset = b1.split(",",-2);

yPoints[0] = Integer.parseInt(yset[0]);

yPoints[1] = Integer.parseInt(yset[1]);

yPoints[2] = Integer.parseInt(yset[2]);

}

else{

a1 = a.getText();

b1 = b.getText();

c1 = c.getText();

d1 = d.getText();

a2 = Integer.parseInt(a1);

b2 = Integer.parseInt(b1);

c2 = Integer.parseInt(c1);

d2 = Integer.parseInt(d1);

}

tb.setVisible(false);

repaint();

}

@Override

public void paint(Graphics g){

//Panel draw=new Panel();

if(circ == true){

g.drawOval(a2, b2, c2, d2);

}

else if(squr == true){

g.drawRect(a2, b2, c2, d2);

}

else{

g.drawPolygon(xPoints, yPoints, 3);

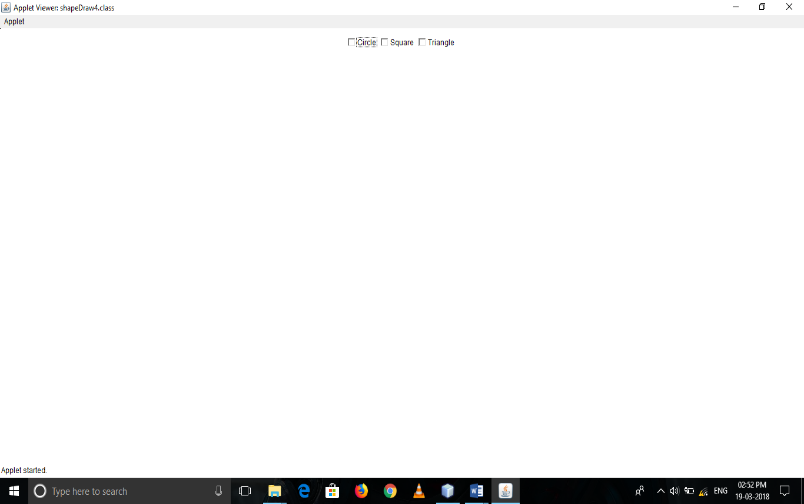
}

}

// TODO overwrite start(), stop() and destroy() methods

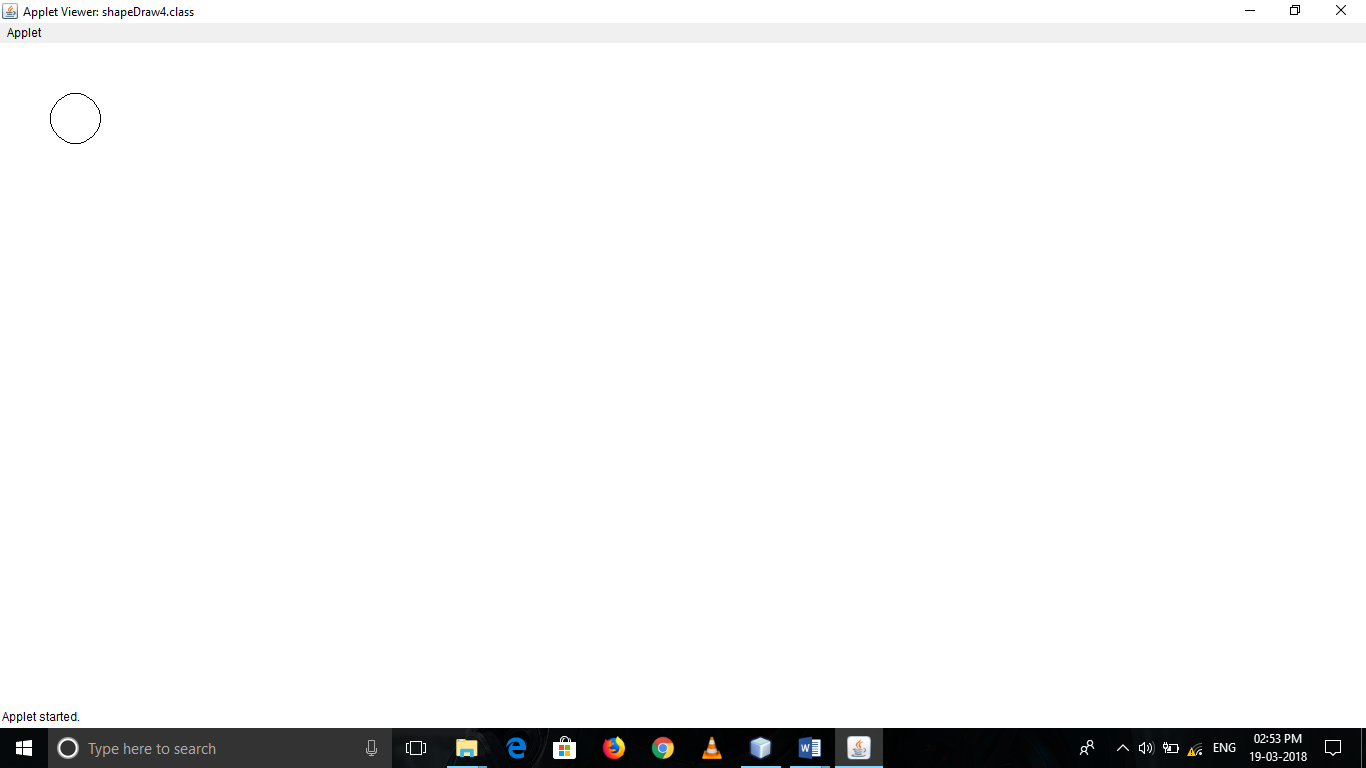
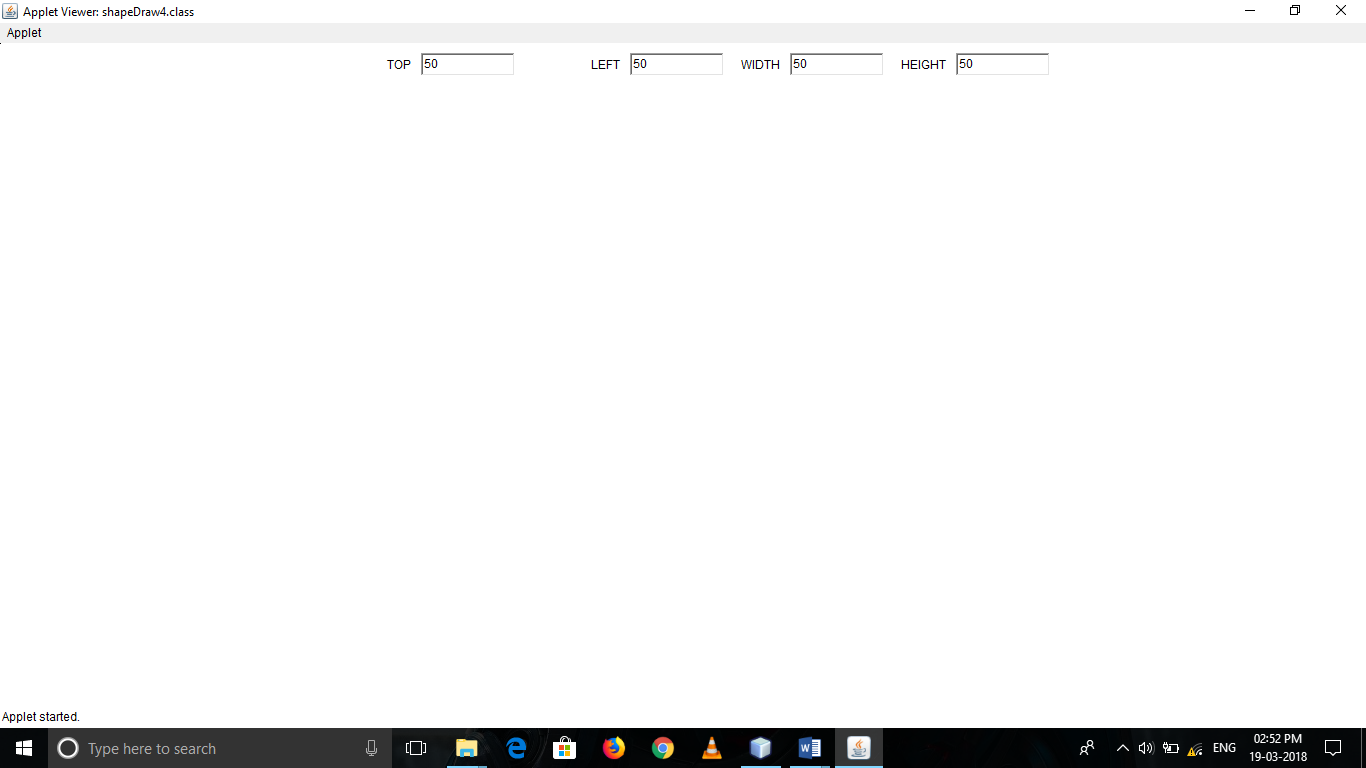
}

**Input:** CheckBox must be ticked for circle, square and triangle

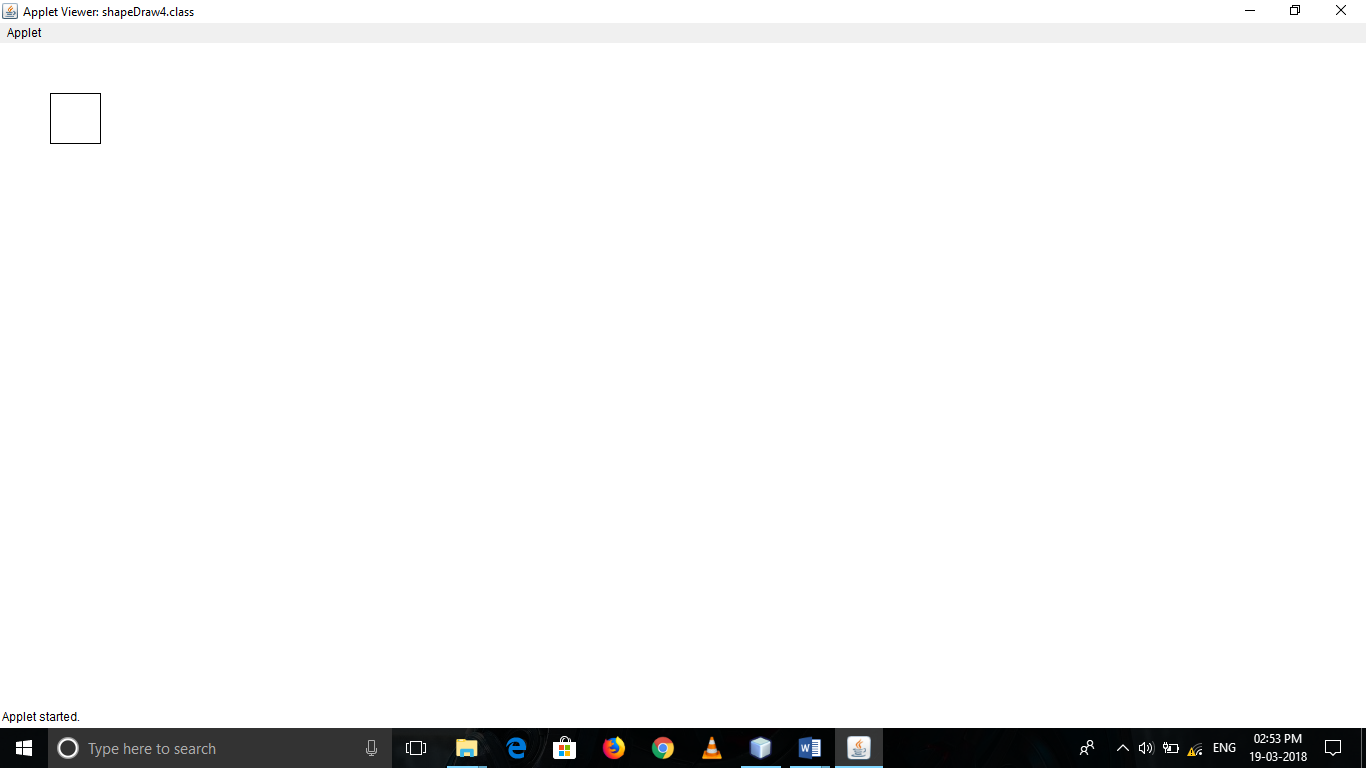


**Output with screenshot.**

//CheckBox ticked Circle



//CheckBox ticked Square



//CheckBox ticked triangle

