

Government Polytechnic College, Jodhpur
Department of Computer Science (NBA Accredited)

Programme: Diploma
Course: JAVA TOOLS
Course CODE: CS-309
Max.Marks : 15

Class Test: II

Session: 2017-18
Year: IIIrd
Time: 14:45 to 15:45
Date: 27-01-2018

Instructions to candidates: Attempt Any Three Questions

Sl#	Question	Marks	CO MAPPING
1	Explain drawRect () and drawOval () method with example.	5	CO3
2	Explain drawPolygon() and fillPolygon() method with example.	5	CO3
3	Explain Frame from Abstract Window Toolkit with Example in Java.	5	CO4
4.	Explain CheckBox from Abstract Window Toolkit with Example in Java.	5	CO4

SOLUTIONS:

Q1. Explain drawRect() and drawOval() method with example.

Sol: Drawing Rectangle:

The drawRect() method is used to draw an rectangle.

Void drawRect(int *top*, int *left*, int *width*, int *height*)

The upper-left corner of the rectangle is at top left. The dimensions of the rectangle are specified by width and height.

Example:

```
import java.awt.*;
import java.applet.*;
public class Rectangle extends Applet {
    public void paint (Graphics g) {
        g.drawRect (10,10,60,50);
    }
}
```

Drawing Ellipses :

To draw an ellipse drawOval() method is used.

Void drawOval(int top, int left, int width, int height)

The ellipse is drawn within a bounding rectangle whose upper-left corner is specified by top, left and whose width and height are specified by width and height.

Example:

```
import java.awt.*;
import java.applet.*;
public class Ellipses extends Applet {
    public void paint (Graphics g) {
        g.drawOval (10,10,50,50);
    }
}
```

Q2. Explain drawPolygon() and fillPolygon() method with example.

Sol: It is possible to draw arbitrarily shaped figures using drawPolygon() and fillPolygon() methods.

Void drawPolygon()(int x[], inty[], int numpoints)

Void fillPolygon()(int x[], inty[], int numpoints)

The polygon's endpoints are specified by the coordinate pairs contained within the x and y arrays. The number of points defined by x and y is specified by numpoints. There are alternative forms of these methods in which the polygon is specified by a Polygon object.

Example:

```
import java.awt.*;
import java.applet.*;
public class HourGlass extends Applet {
    public void paint(Graphics g) {
        int xpoints[] = { 30,200,30,200,200,30};
        int ypoints[] = {30,30,200,200,30};
        int num = 5;
        g.drawPolygon (xpoints, ypoints, num);
    }
}
```

Q3. Explain Frame from Abstract Window Toolkit with Example in Java.

Sol: If you want to create a frame window the constructors used to create frame are:

Frame()

Frame (String Title)

The first constructor simply creates a frame window without any title, while the second one creates a frame with the title specified as string in the argument. Some of the methods are :

- (a) Void setSize(int width, int height): It is used to set the dimension of the window. The new dimension in the form of width and height is passed as arguments.
- (b) Dimension getSize() : It is used to return the current size of the window, contained within the width and height fields of Dimension object.
- (c) Void setVisible(Boolean flag) : it is used to make the window visible after its creation. The component is visible only if the argument passed is true.

Example:

```
import java.awt.*;
public class FrameDemo extends Frame {
    public FrameDemo (String title) {
        Super(title);
    }
    public static void main(String[] args) {
        FrameDemo frameDemo = new FrameDemo("Demo Frame");
        frameDemo.setSize(200, 300);
        frameDemo.setVisible(true);
    }
}
```

Q4. Explain CheckBox from Abstract Window Toolkit with Example in Java.

Sol: Checkboxes are used as on-off or yes-no switches. An element –like button might be needed to trigger an event to check the state of the checkboxes. The checkboxes are the objects of checkbox class, which support the following constructors:

checkbox()

checkbox(String str)

checkbox(String str, Boolean on)

checkbox(String str, CheckBoxGroup cbg, Boolean on)

The first form of constructor creates a checkbox whose label is blank initially and the state of the checkbox is unchecked. In the other forms of constructors, string argument account for the label of the checkbox, Boolean arguments (true/false) accounts for whether the checkbox will be created checked or unchecked.

Example:

```
import java.applet.*;
import java.awt.*;
import java.awt.event.*;
Public class CheckboxClass extends Applet implements ItemListener {
    String msg = "";
    Checkbox Win98, winNT, solaris, mac;
    Public void init() {
        Win98 = new Checkbox ( "Windows 98/XP", null, true);
        winNT = new Checkbox ("Windows NT/2000");
        solaris = new Checkbox ("Solaris");
        mac = new Checkbox ("MacOS");
        add(Win98);
        add(winNT);
        add(solaris);
        add(mac);
        Win98.addItemListener (this);
        winNT.addItemListener (this);
        solaris.addItemListener (this);
        mac.addItemListener (this);
    }
    Public void itemStateChanged(ItemEvent ie) {
        Repaint();
    }
    Public void paint (Graphics g) {
        msg = "Current State: ";
        g.drawString (msg, 6, 80);
        msg = " Windows 98/XP: " + Win98.getState();
        g.drawString (msg, 6, 100);
        msg = " Windows NT/2000: " + winNT.getState();
        g.drawString (msg, 6, 120);
        msg = " Solaris: " + solaris.getState();
        g.drawString (msg, 6, 140);
        msg = " MacOS: " + mac.getState();
        g.drawString (msg, 6, 160);
    }
}
```