

16.5 Drawing Rectangles

The `drawRect()` method is used to display a rectangle.

(a) To Draw a Hollow Rectangle

Syntax is :

```
drawRect(int x, int y, int width, int height);
```

Where,

`x` : the x coordinate of the rectangle

`y` : the coordinate of the rectangle

`width` : the width of the rectangle

`height` : the height of the rectangle

Example :

The statement

```
g.drawRect(30, 30, 40, 50);
```

will draw a rectangle starting at (30, 30) having a width of 40 pixels and a height of 50 pixels. Note that `g` is a graphics object.

(b) To draw a Solid Rectangle

To draw a solid box we should make use of the `fillRect()` method. Syntax is:

```
fillRect(int x, int y, int width, int height);
```

The rectangle is filled using the graphics context's current color.

(c) **To draw rounded corners follow or solid rectangle**

We can also draw rounded rectangles (rectangles with rounded corners) by using `drawRoundedRect()` and `fillRoundRect()` methods. Syntax of the two method are :

```
drawRoundRect(int x, int y, int width, int height, int arcWidth, int arcHeight);  
fillRoundRect(int x, int y, int width, int height, int arcWidth, int arcHeight);
```

Where,

x : (i the x coordinate of the rectangle

y : the y coordinate of the rectangle

width : the width of the rectangle

height : the height of the rectangle

arcWidth : the horizontal diameter of the arc at the four corners.

arcHeight : the vertical diameter of the arc at the four corners.

Example

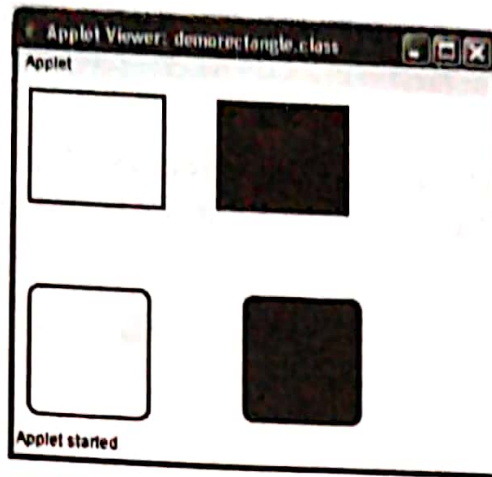
//Code of demorectangle.java

```
import java.awt.*;  
import java.applet.*;  
public class demorectangle extends Applet  
{  
    public void paint(Graphics g)  
    {  
        g.drawRect(10,10,100,80);  
        g.fillRect(150,10,100,80);  
        g.drawRoundRect(10,150,90,90,15,15);  
        g.fillRoundRect(170,150,90,90,15,15);  
    }  
}
```

//Code of demorectangle.html

```
<HTML>  
  <BODY>  
    <APPLET CODE ="demorectangle.class"  
      Width = 400  
      Height = 280>  
  </APPLET>  
</BODY>  
</HTML>
```


Output :



16.6 Drawing Circle and ELLIPSES

To draw a circle or an ellipse, use **drawOval()** method. The Syntax is :

✓ **drawOval(int x, int y, int width, int height);**

Where,

x : the x coordinate of the upper left corner of the oval.

y : the y coordinate of the upper left corner of the oval.

width : the width of the oval.

height : the height of the oval.

To draw a solid oval filled with the current color use the **fillOval()** method. Syntax is :

✓ **fillOval(int x, int y, int width, int height);**

Example :

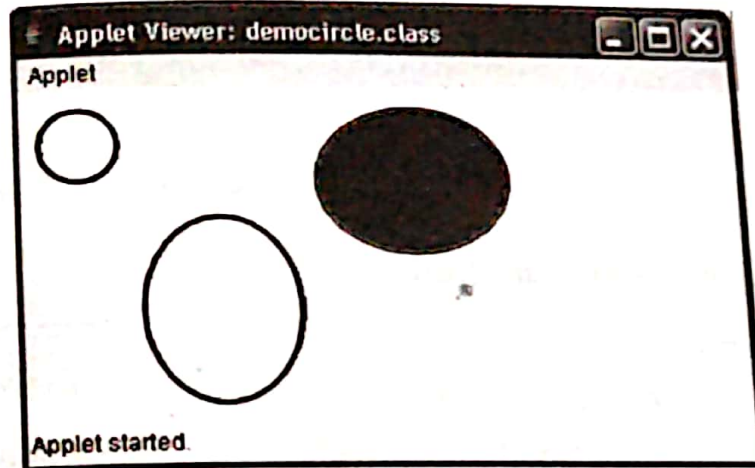
//Code of democircle.java

```
import java.awt.*;
import java.applet.*;
public class democircle extends Applet
{
    public void paint(Graphics g)
    {
        g.drawOval(10,10,40,40);
        g.drawOval(60,70,80,100);
        g.setColor(Color.red);
        g.fillOval(150,10,100,80);
    }
}
```

//Code of democircle.html

```
<HTML>
  <BODY>
    <APPLET CODE = "democircle.class"
      Width = 400
      Height = 180>
  </APPLET>
</BODY>
</HTML>
```

Output :



16.7 Drawing Arcs

An arc is a part of an oval. Arcs can be drawn with **drawArc()** and filled with **fillArc()**. Syntax is :

drawArc(int x, int y, int width, int height, int startAngle, int arcAngle); or
fillArc(int x, int y, int width, int height, int startAngle, int arcAngle);

Where,

x : the x coordinate of the upper-left corner of the arc.

y : the y coordinate of the upper-left corner of the arc.

width : the width of the arc.

height : the height of the arc.

startAngle : the beginning angle.

arcAngle : the angular extent of the arc.

Example :

//Code of demoarc.java

```
import java.awt.*;
import java.applet.*;
public class demoarc extends Applet
{
    public void paint(Graphics g)
    {
        g.drawArc(10,10,80,80,0,270);
        g.setColor(Color.red);
        g.fillArc(150,10,100,80,0,180);
    }
}
```

//Code of demoarc.html

```
<HTML>
  <BODY>
    <APPLET CODE = "demoarc.class"
      Width = 400
      Height = 180>
    </APPLET>
  </BODY>
</HTML>
```

Output :

