Designing Geometrical Shapes with Java

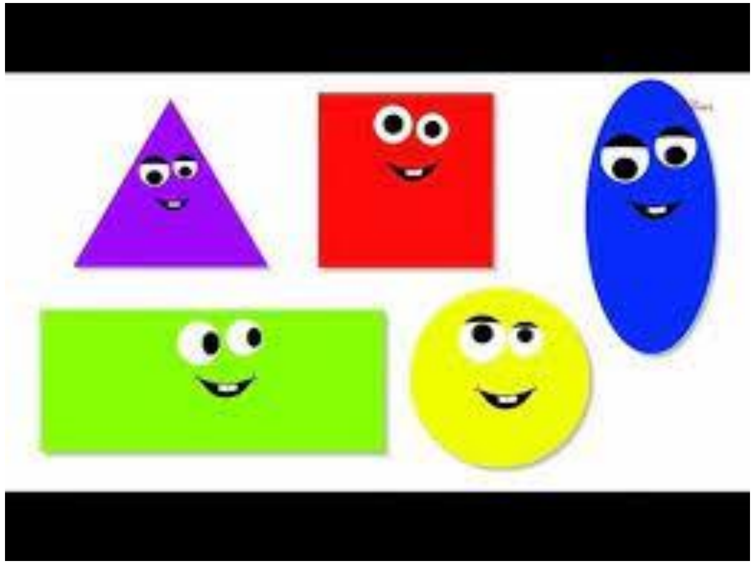
Dr. Krishnendu Guha

Assistant Professor/ Lecturer

School of Computer Science and Information Technology

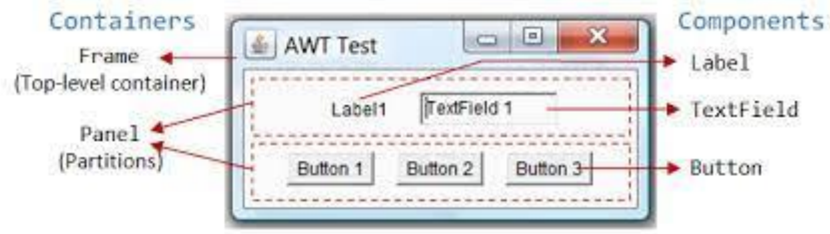
University College Cork

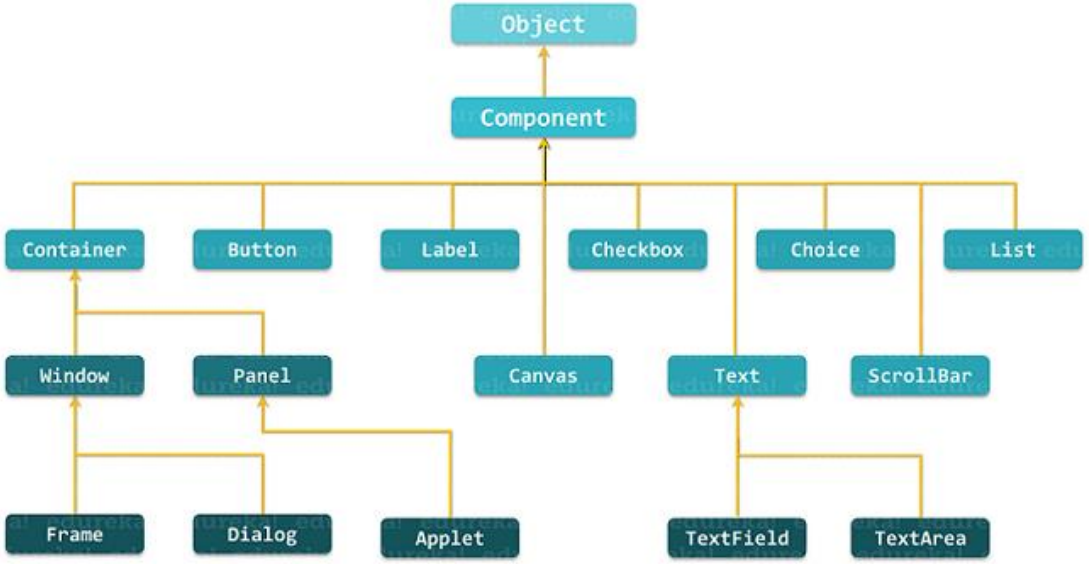
Email: kguha@ucc.ie



AWT

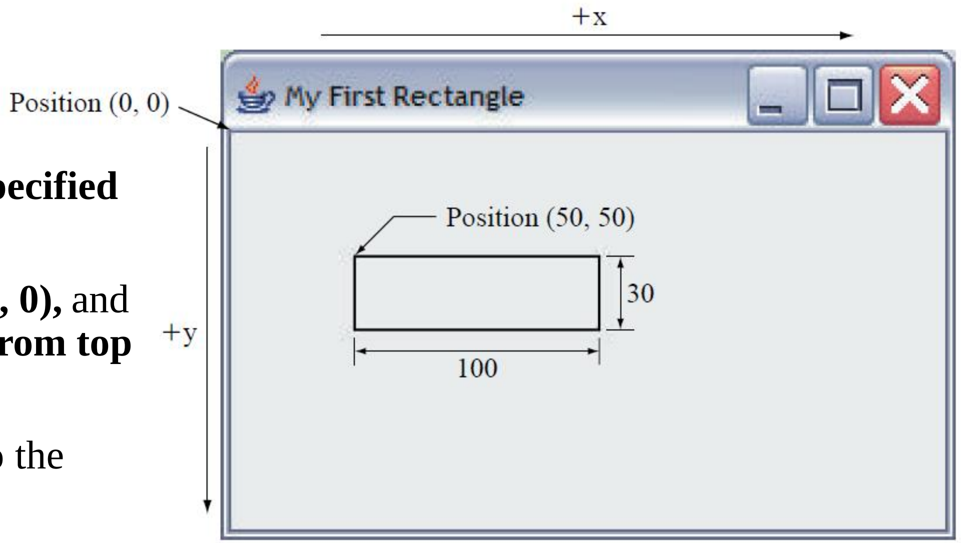
- The Full Form Of AWT is **Abstract Window Toolkit**.
- The Abstract Window Toolkit is **a platform-dependent API used to develop GUI (Graphical User Interface) or window-based applications** in Java.
- Basically, the AWT is a **member of the Java Foundation Classes** – the approved API implementing a GUI for a Java program.





Graphics Object

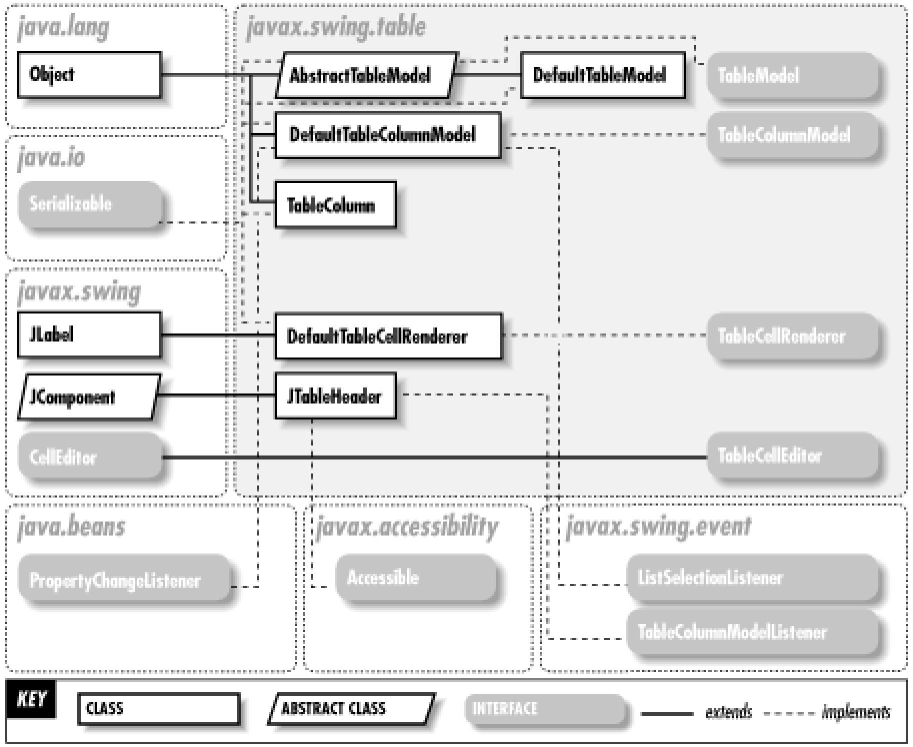
- We can draw geometric shapes on a frame window by calling appropriate **methods of the Graphics object**.
- java.awt.Graphics
- For example, if **g is a Graphics object**, then we can write
  - g.drawRect(50, 50, 100, 30);



- To **display a rectangle 100 pixels wide and 30 pixels high** at the **specified position** (50, 50).
- The **top left corner, just below the window title bar, is position (0, 0),** and the **x value increases from left to right** and **the y value increases from top to bottom.**
- Notice that the direction in which the y value increases is opposite to the normal two-dimensional graph.
- The **area of a frame which we can draw** is called the ***content pane of a frame***.
- The **content pane excludes the area of a frame** that *excludes the regions such as the border, scroll bars, the title bar, the menu bar, and others*

Swing

- Swing is a GUI widget toolkit for Java.
- It is part of **Oracle's Java Foundation Classes (JFC)** – an API for providing a graphical user interface (GUI) for Java programs.
- Swing was **developed to provide a more sophisticated set of GUI components than the earlier Abstract Window Toolkit (AWT).**



java.lang
Object

java.io
Serializable

javax.swing
JLabel
JComponent
CellEditor

javax.swing.table
AbstractTableModel
DefaultTableModel
TableModel
DefaultTableColumnModel
TableColumnModel
TableColumn
DefaultTableCellRenderer
TableCellRenderer
JTableHeader
TableCellEditor

java.beans
PropertyChangeListener

javax.accessibility
Accessible

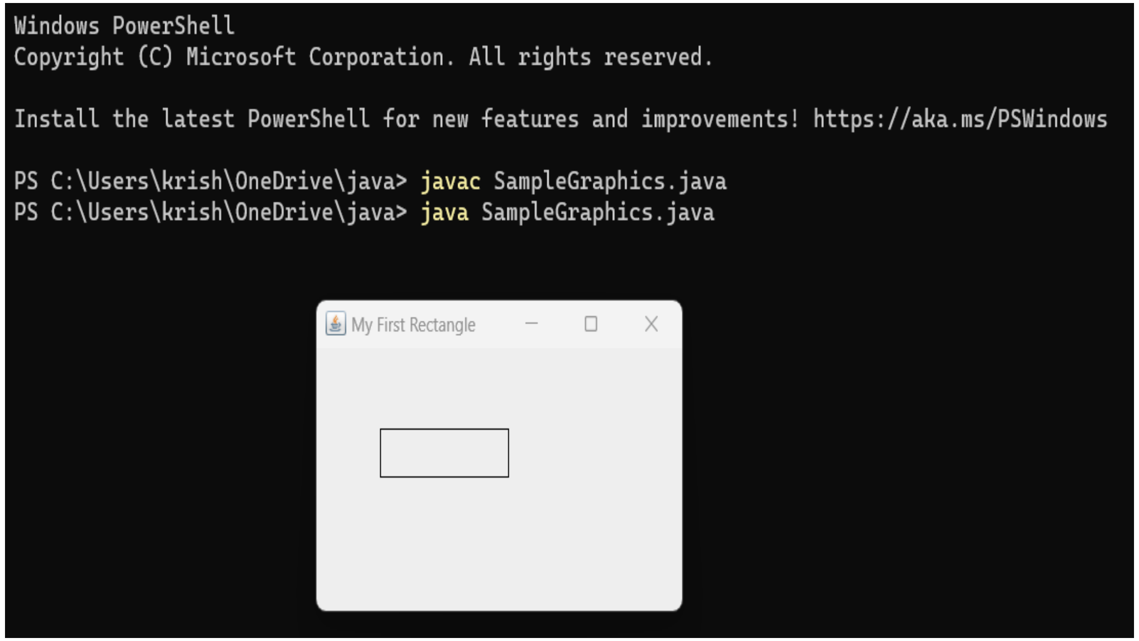
javax.swing.event
ListSelectionListener
TableColumnModelListener

KEY
CLASS
ABSTRACT CLASS
INTERFACE
extends
implements

Draw a rectangle on a frame window's content pane

```
import javax.swing.*; //for JFrame
import java.awt.*; //for Graphics and Container
class SampleGraphics {
public static void main( String[] args ) {
        JFrame win;
        Container contentPane;
        Graphics g;
        win = new JFrame("My First Rectangle");
        win.setSize(300, 200);
        win.setLocation(100,100);
        win.setVisible(true);
        contentPane = win.getContentPane();
        g = contentPane.getGraphics();
        g.drawRect(50,50,100,30);
}
}
```

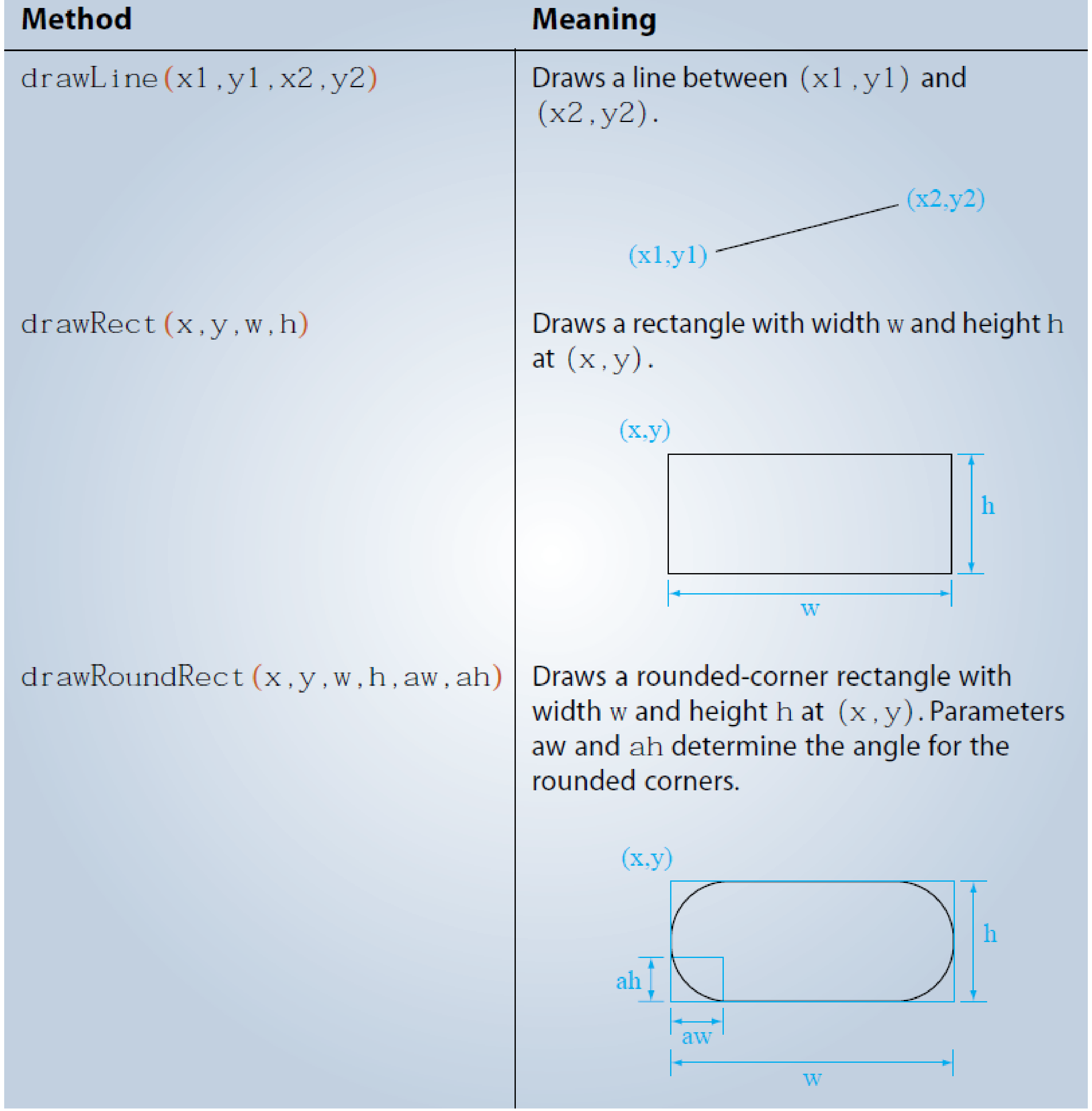
To draw on the content pane of a frame window, first we must get the content pane’s Graphic object.
Then we call this Graphics method to draw geometric shapes.



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\krish\OneDrive\java> javac SampleGraphics.java
PS C:\Users\krish\OneDrive\java> java SampleGraphics.java
```

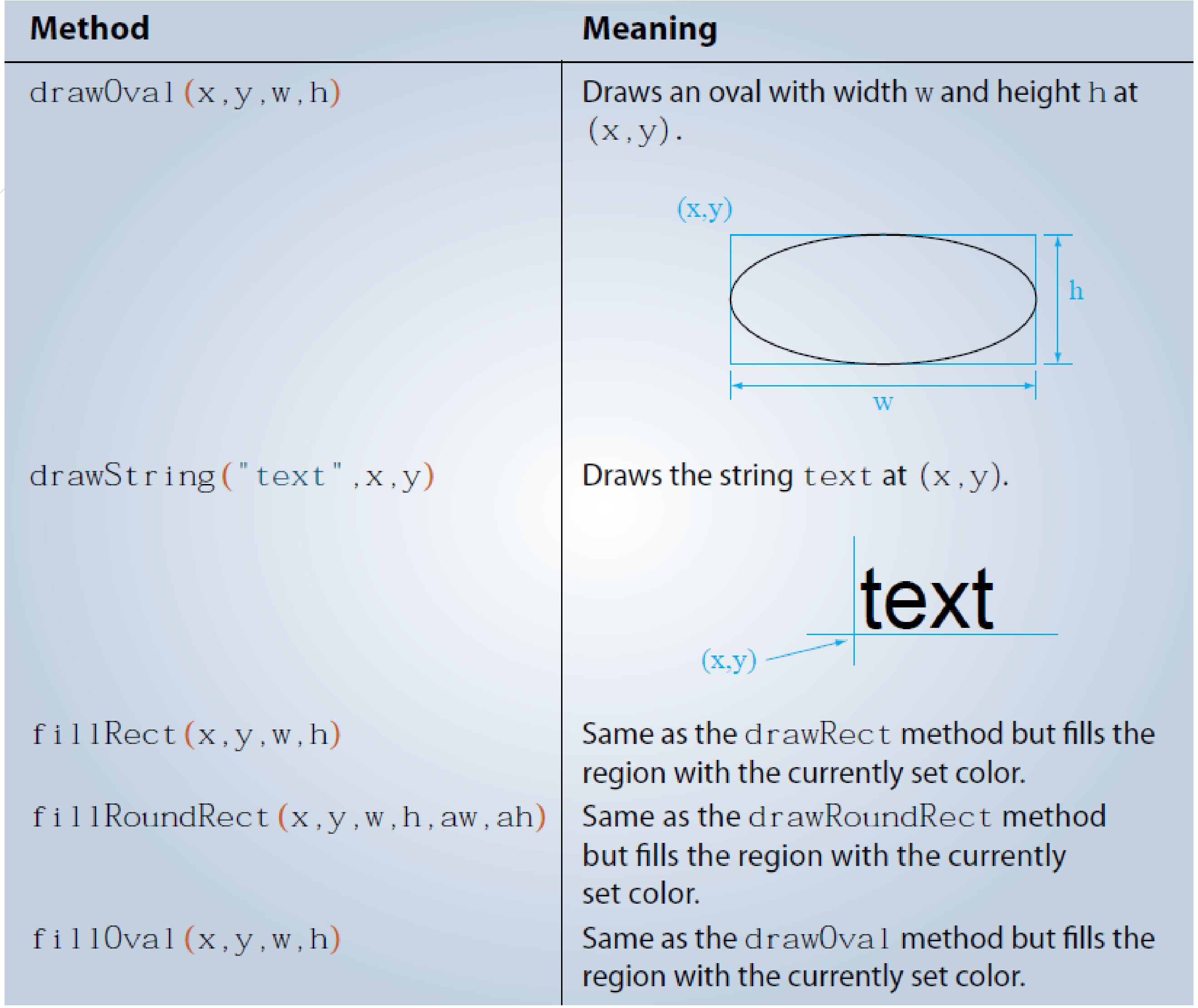
- *To draw geometric shapes on the content pane of a frame window:*
- **1.** *The content pane is declared as a* **Container,** *for example,* Container contentPane;
- **2.** *The frame window must be visible on the screen before we can get its content pane and the content pane's* **Graphics** *object.*

A partial list of drawing methods defined for the Graphics **class**



Method	Meaning
drawLine(x1,y1,x2,y2)	Draws a line between (x1,y1) and (x2,y2).
drawRect(x,y,w,h)	Draws a rectangle with width w and height h at (x,y).
drawRoundRect(x,y,w,h,aw,ah)	Draws a rounded-corner rectangle with width w and height h at (x,y). Parameters aw and ah determine the angle for the rounded corners.

(x2,y2)
(x1,y1)
(x,y)
h
w
(x,y)
h
ah
aw
w



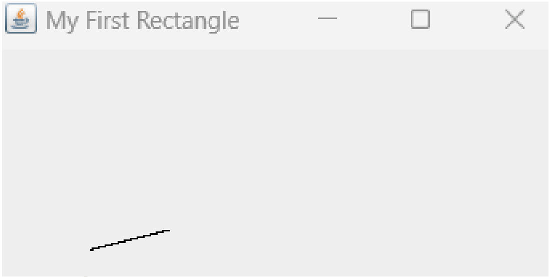
Method	Meaning
drawOval(x,y,w,h)	Draws an oval with width w and height h at (x,y).
drawString("text",x,y)	Draws the string text at (x,y).
fillRect(x,y,w,h)	Same as the drawRect method but fills the region with the currently set color.
fillRoundRect(x,y,w,h,aw,ah)	Same as the drawRoundRect method but fills the region with the currently set color.
fillOval(x,y,w,h)	Same as the drawOval method but fills the region with the currently set color.

A partial list of drawing methods defined for the Graphics class

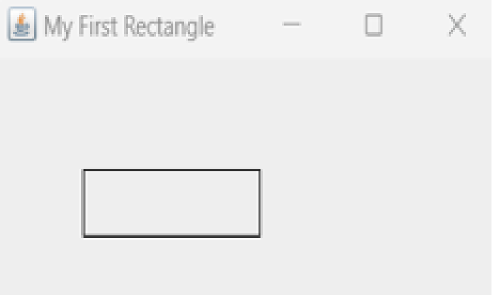
Notice the distinction between the draw and fill methods.

The draw method will draw the boundary only, while the fill method fills the designated area with the currently selected color

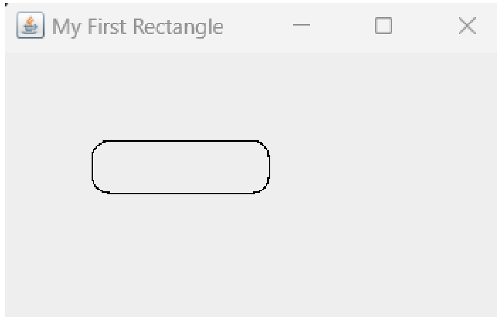
```
g.drawLine(50,50,100,30);
```



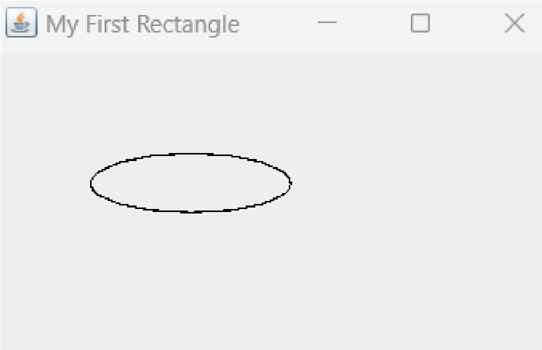
```
g.drawRect(50,50,100,30);
```



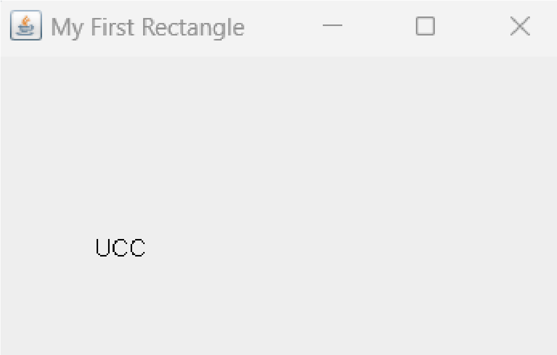
```
g.drawRoundRect(50,50,100,30,20,20);
```



```
g.drawOval(50,50,100,30);
```

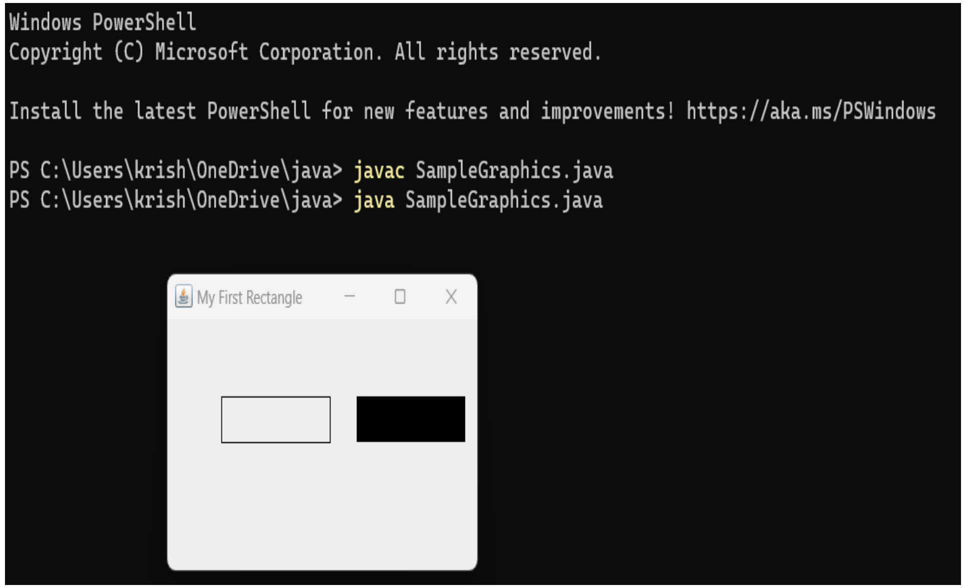


```
g.drawString(“UCC”,60,50);
```



Draw a rectangle on a frame window's content pane

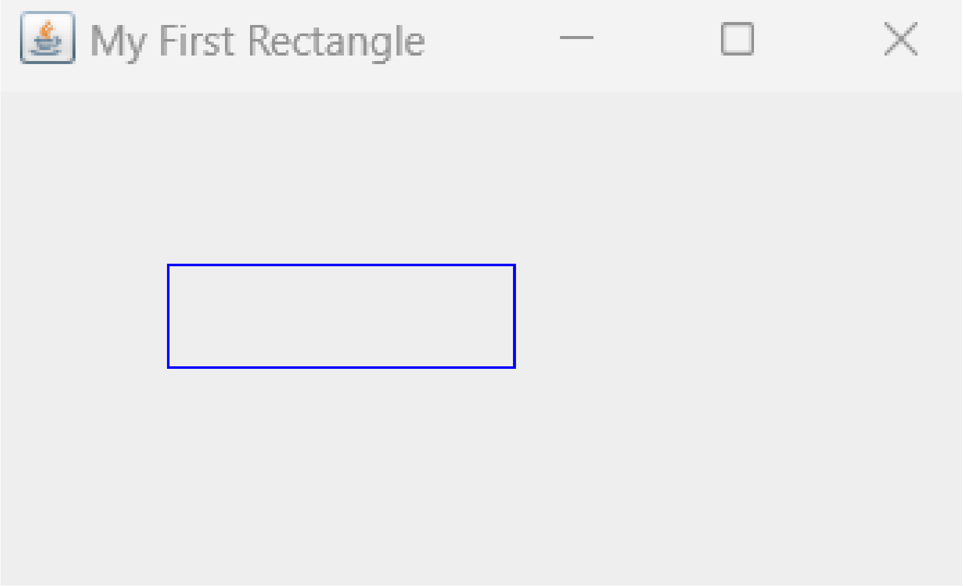
```
import javax.swing.*; //for JFrame
import java.awt.*; //for Graphics and Container
class SampleGraphics {
public static void main( String[] args ) {
        JFrame win;
        Container contentPane;
        Graphics g;
        win = new JFrame("My First Rectangle");
        win.setSize(300, 200);
        win.setLocation(100,100);
        win.setVisible(true);
        contentPane = win.getContentPane();
        g = contentPane.getGraphics();
        g.drawRect(50,50,100,30);
        g.fillRect(175, 50, 100, 30);
}
}
```



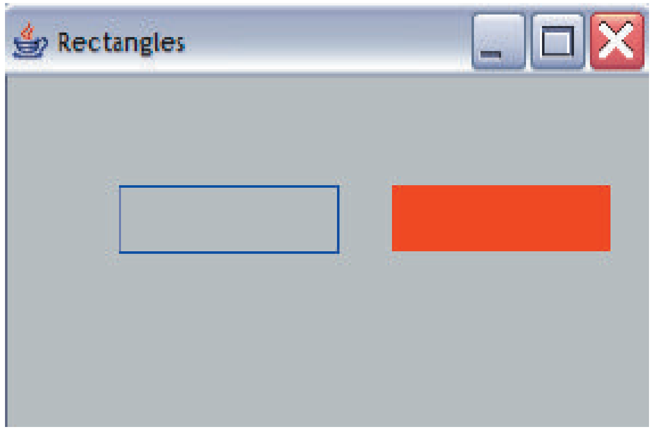
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\krish\OneDrive\java> javac SampleGraphics.java
PS C:\Users\krish\OneDrive\java> java SampleGraphics.java
```

Draw a rectangle with outline in blue

```
import javax.swing.*; //for JFrame
import java.awt.*; //for Graphics and Container
class SampleGraphics {
public static void main( String[] args ) {
JFrame win;
Container contentPane;
Graphics g;
win = new JFrame("My First Rectangle");
win.setSize(300, 200);
win.setLocation(100,100);
win.setVisible(true);
contentPane = win.getContentPane();
g = contentPane.getGraphics();
g.setColor(Color.BLUE);
g.drawRect(50, 50, 100, 30);
}
}
```



```
import javax.swing.*;
import java.awt.*;
class SampleGraphics2 {
public static void main( String[] args ) {
JFrame win;
Container contentPane;
Graphics g;
win = new JFrame("Rectangles");
win.setSize(300, 200);
win.setLocation(100,100);
win.setVisible(true);
contentPane = win.getContentPane();
contentPane.setBackground(Color.LIGHT_GRAY);
g = contentPane.getGraphics();
g.setColor(Color.BLUE);
g.drawRect(50,50,100,30);
g.setColor(Color.RED);
g.fillRect(175,50,100,30);
}
}
```



ANY
QUESTIONS ?

Thank
You