Methods of the **`Graphics`** class in Java

University of Mount Union

CSC 120

Day 4

Drawing or Filling a Rectangle or Square:

g.drawRect(*over*, *down*, *width*, *height*);

draws an outline of a rectangle in the current drawing color

pixels inside the rectangle are unchanged

over, *down*: screen position of upper-left corner of shape

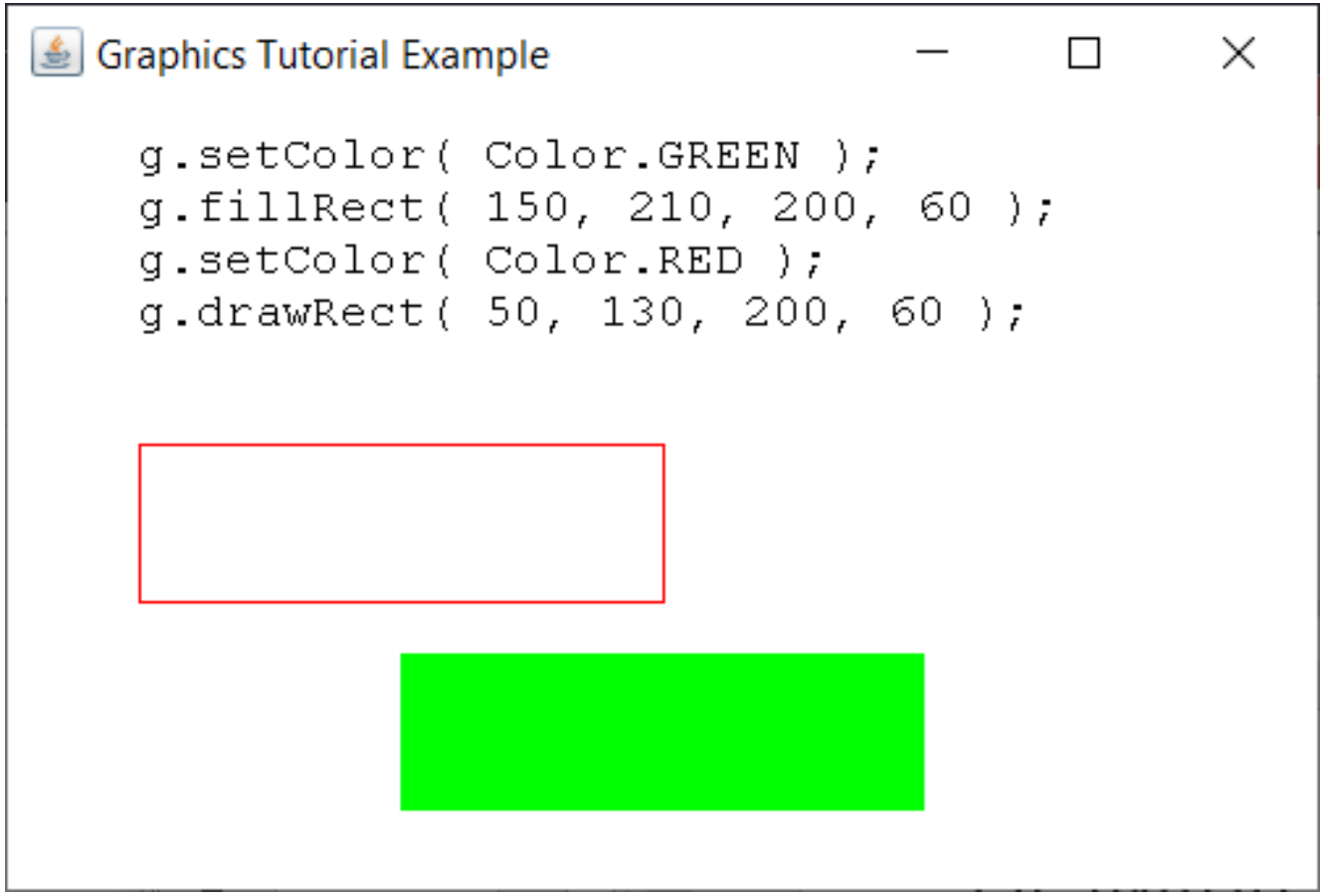
width: width of shape

height: height of shape

g.fillRect(*over*, *down*, *width*, *height*);

fills a rectangle in the current drawing color, including interior

Drawing or Filling a Rectangle or Square:



Drawing or Filling an Oval or Circle:

g.drawOval(*over*, *down*, *width*, *height*);

draws an outline of an oval in the current drawing color that is inscribed in an invisible bounding box for the shape

over, *down*: screen position of upper-left corner of bounding box for the shape

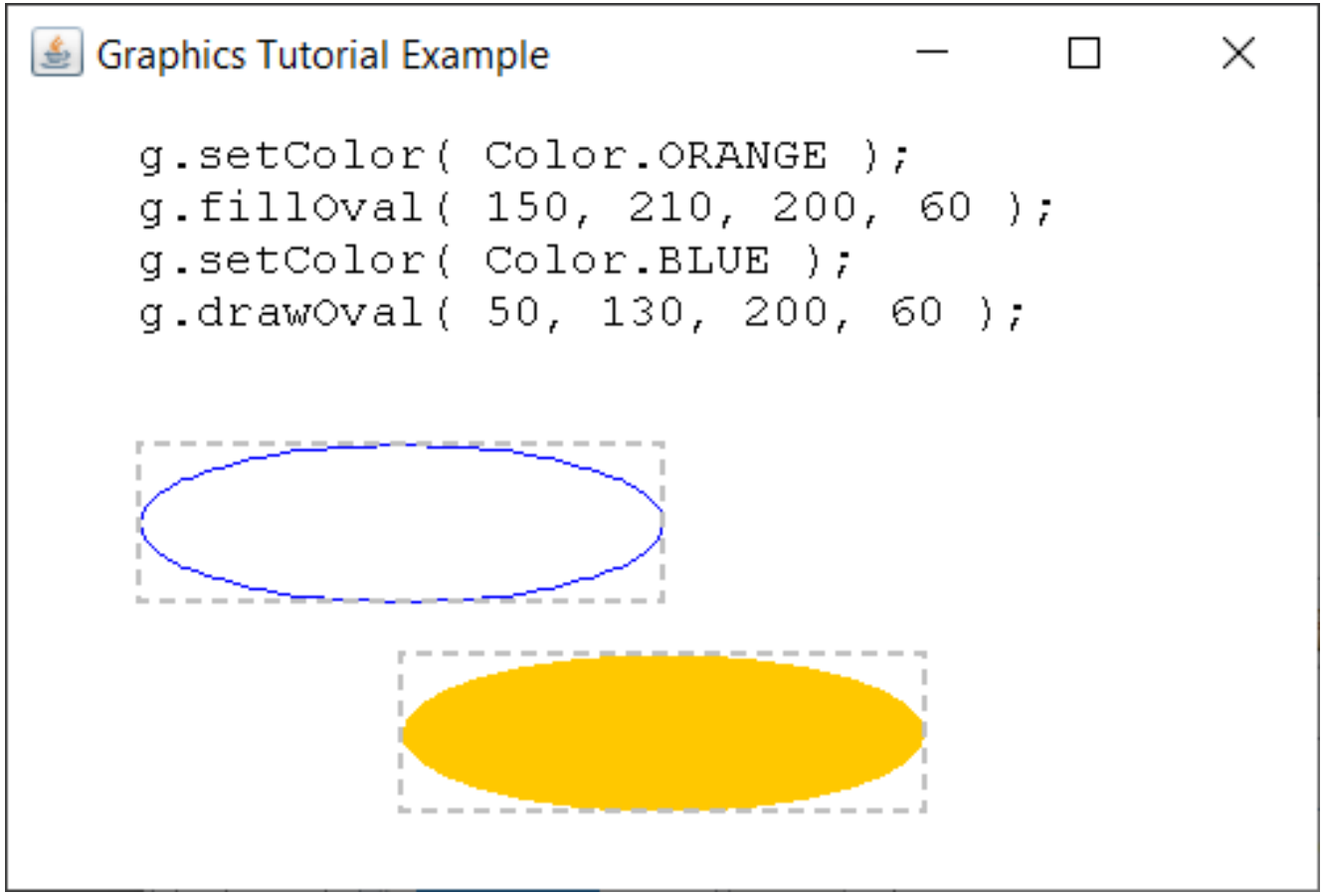
width: width of bounding box

height: height of bounding box

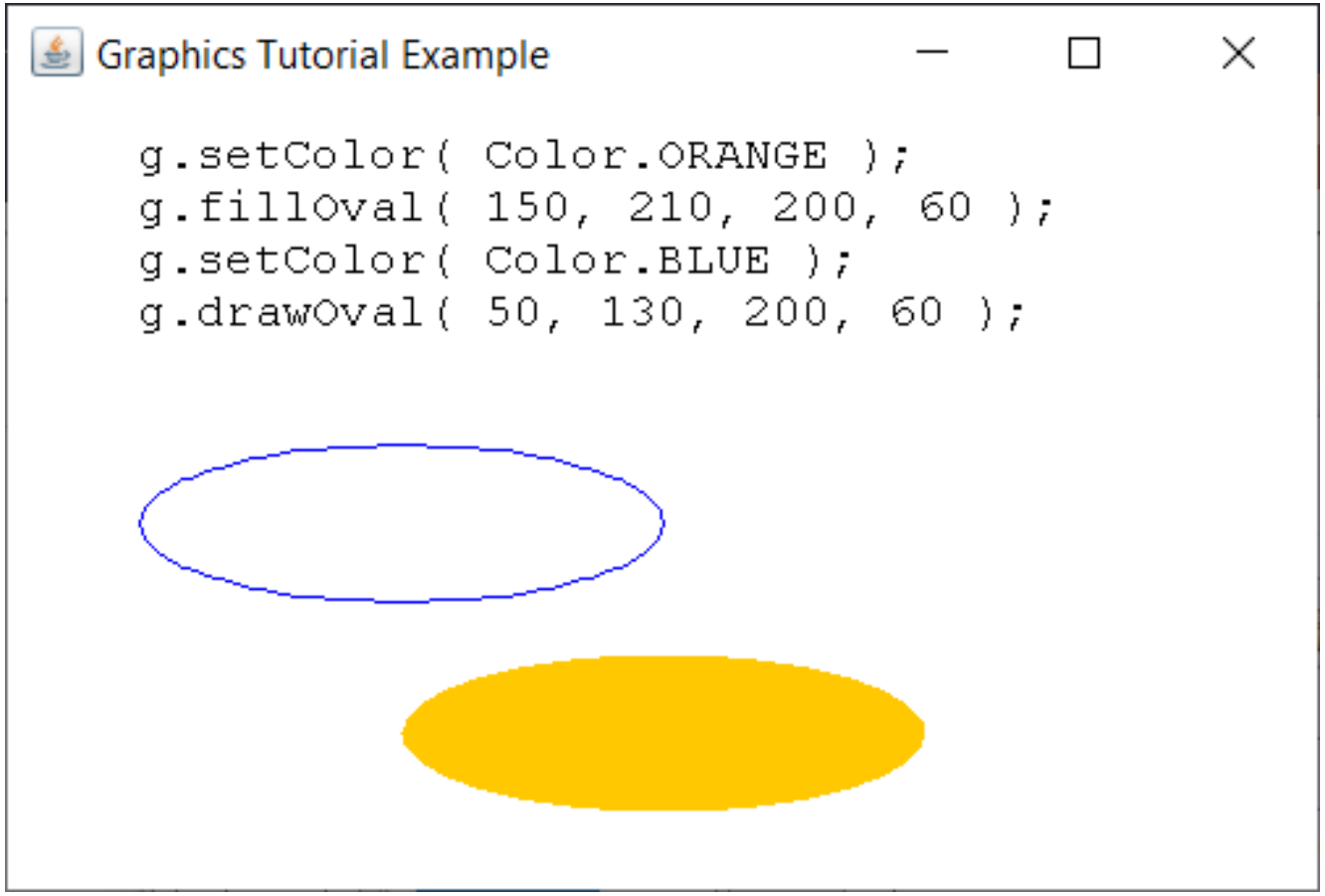
g.fillOval(*over*, *down*, *width*, *height*);

fills an oval in the current drawing color, including interior pixels

Drawing or Filling an Oval or Circle:



Drawing or Filling an Oval or Circle: (this is how it would appear on the screen)



Drawing a Line and Displaying a Text String:

g.drawLine(*x1*, *y1*, *x2*, *y2*);

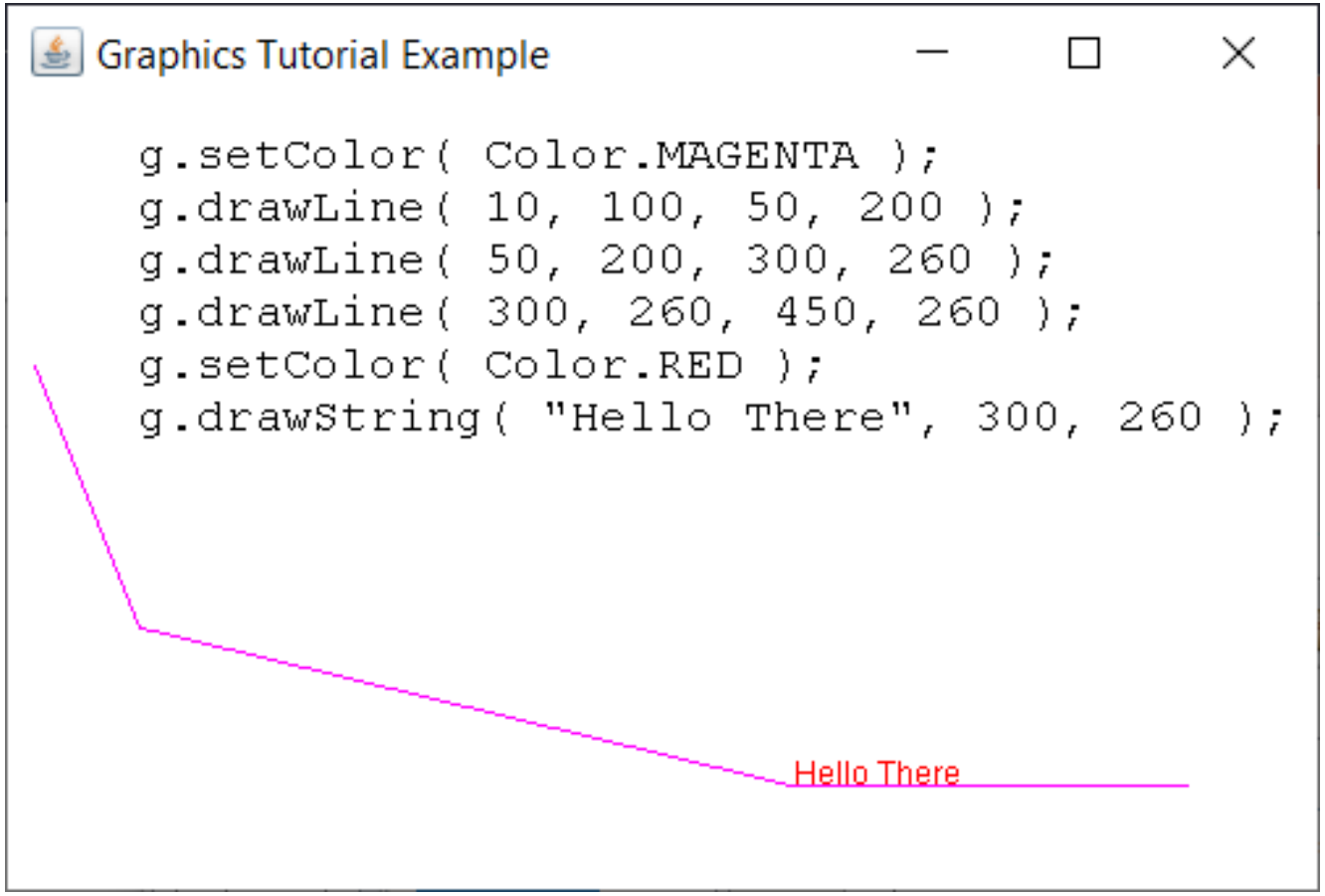
draws a line between the point (*x1*, *y1*) and the point (*x2*, *y2*) in the current drawing color

g.drawString(*text*, *over*, *down*);

displays the *text* on the screen at the point (*over*, *down*)

(*over*, *down*) is the lower-left corner of the text box

Drawing a Line and Displaying a Text String:



Drawing or Filling an Arc:

g.drawArc(*over*, *down*, *width*, *height*, *startAngle*, *arcAngle*);

draws an outline of an arc of an oval in the current drawing color that is inscribed in an invisible bounding box for the shape, starting at *startAngle* degrees and continuing around the oval for *endAngle* degrees

over, *down*: screen position of upper-left corner of bounding box for the shape

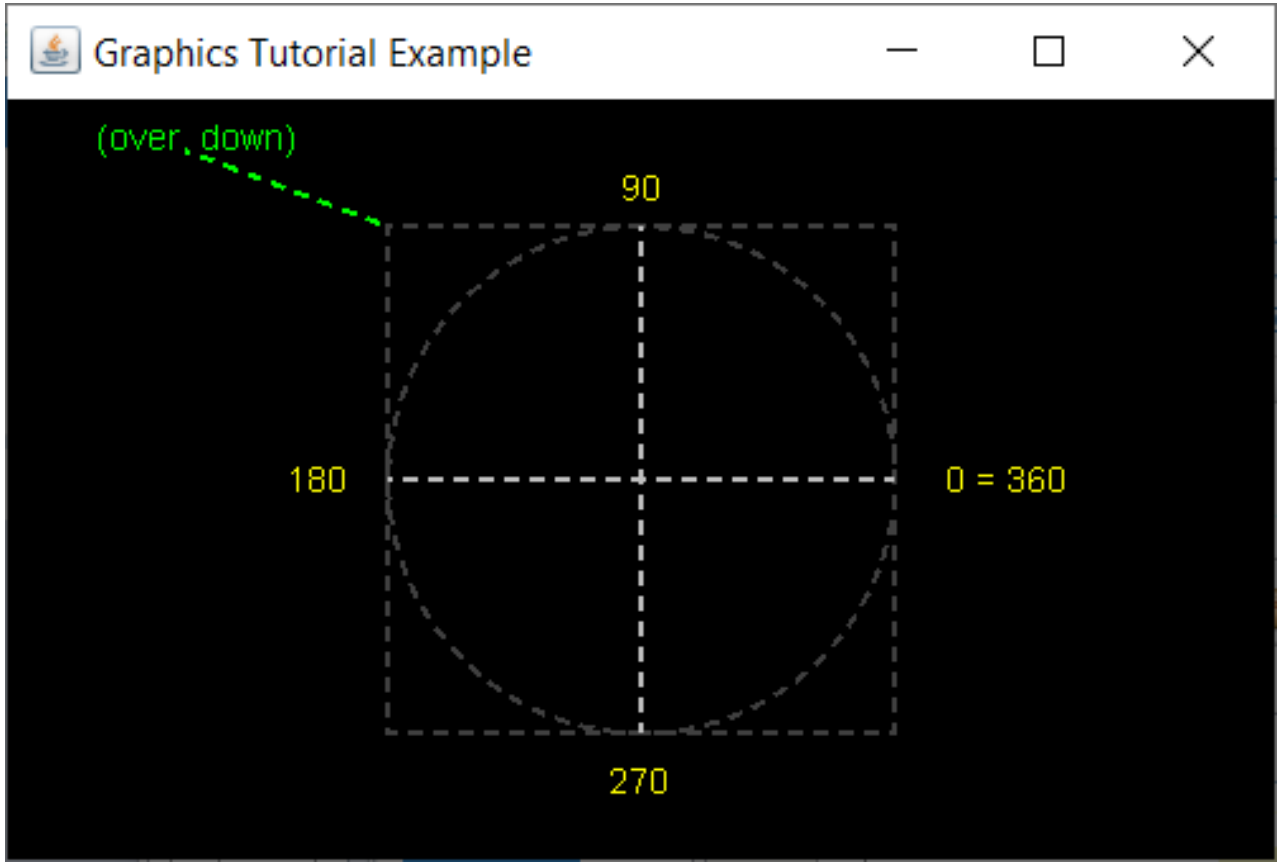
width: width of bounding box

height: height of bounding box

g.fillArc(*over*, *down*, *width*, *height*, *startAngle*, *arcAngle*);

Angle Definition for Java Arcs:

Imagine an invisible set of axes inside the invisible bounding box



0 degrees = EAST on a compass

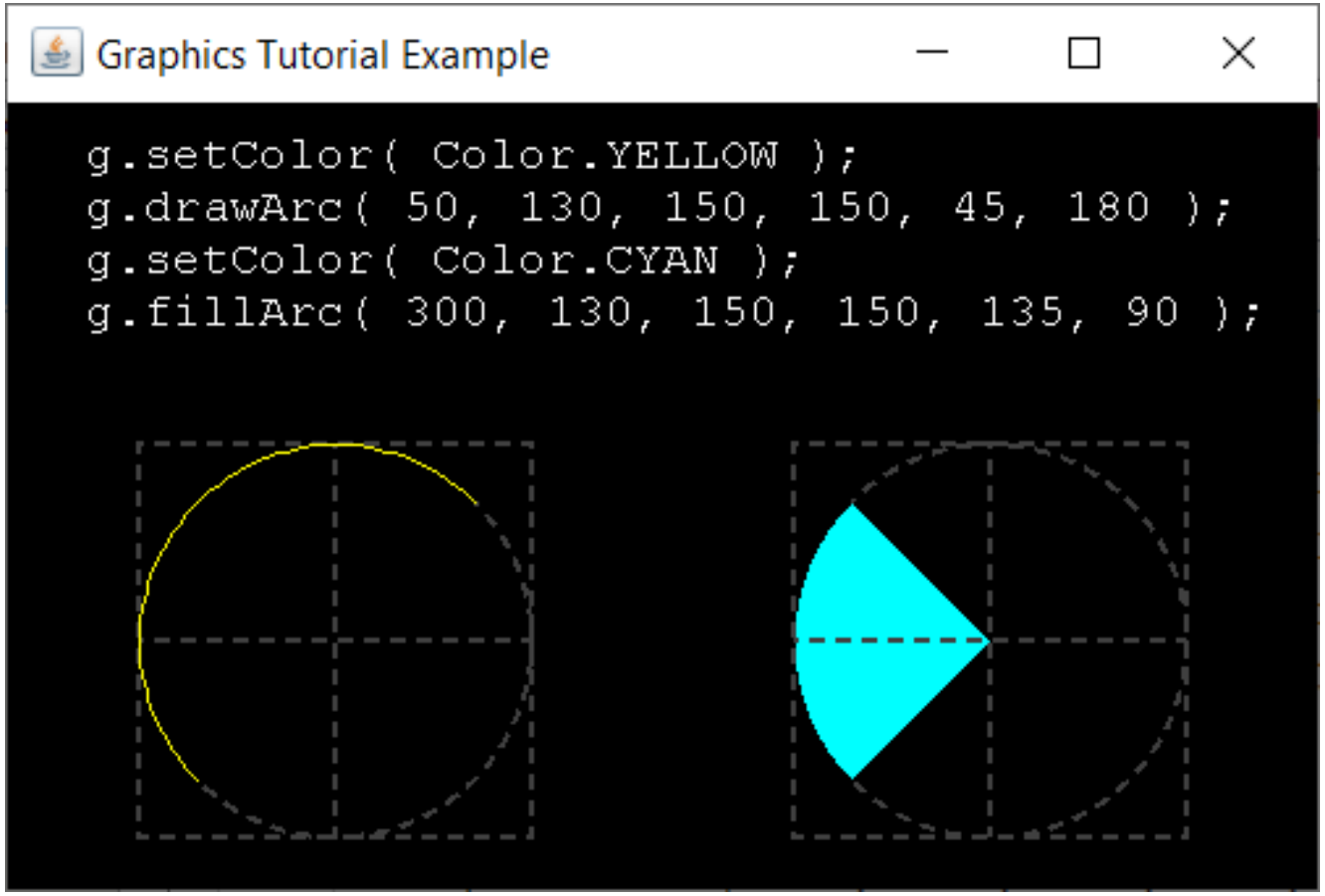
90 degrees = NORTH

180 degrees = WEST

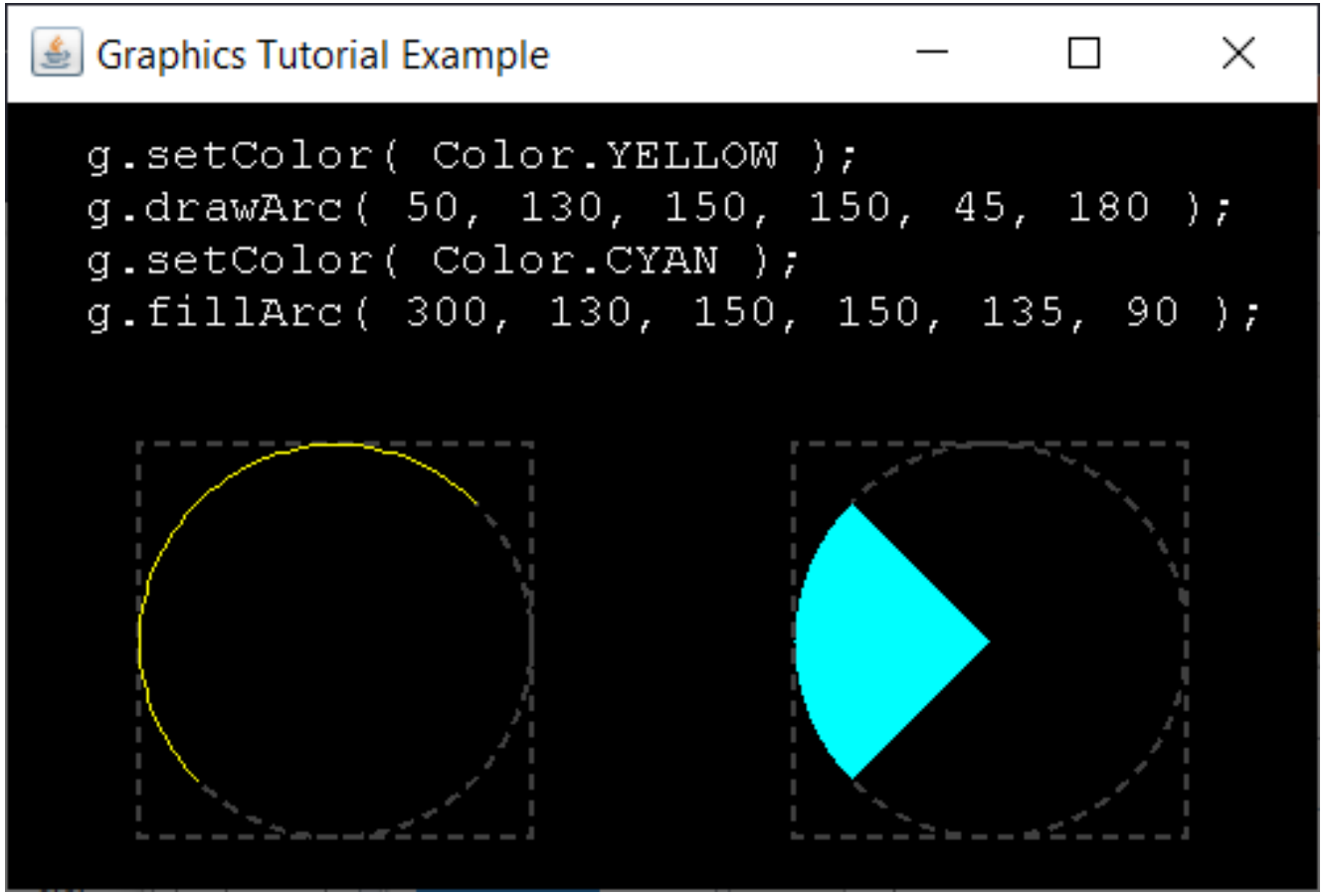
270 degrees = SOUTH

proceed COUNTER-CLOCKWISE around the oval as degrees increase

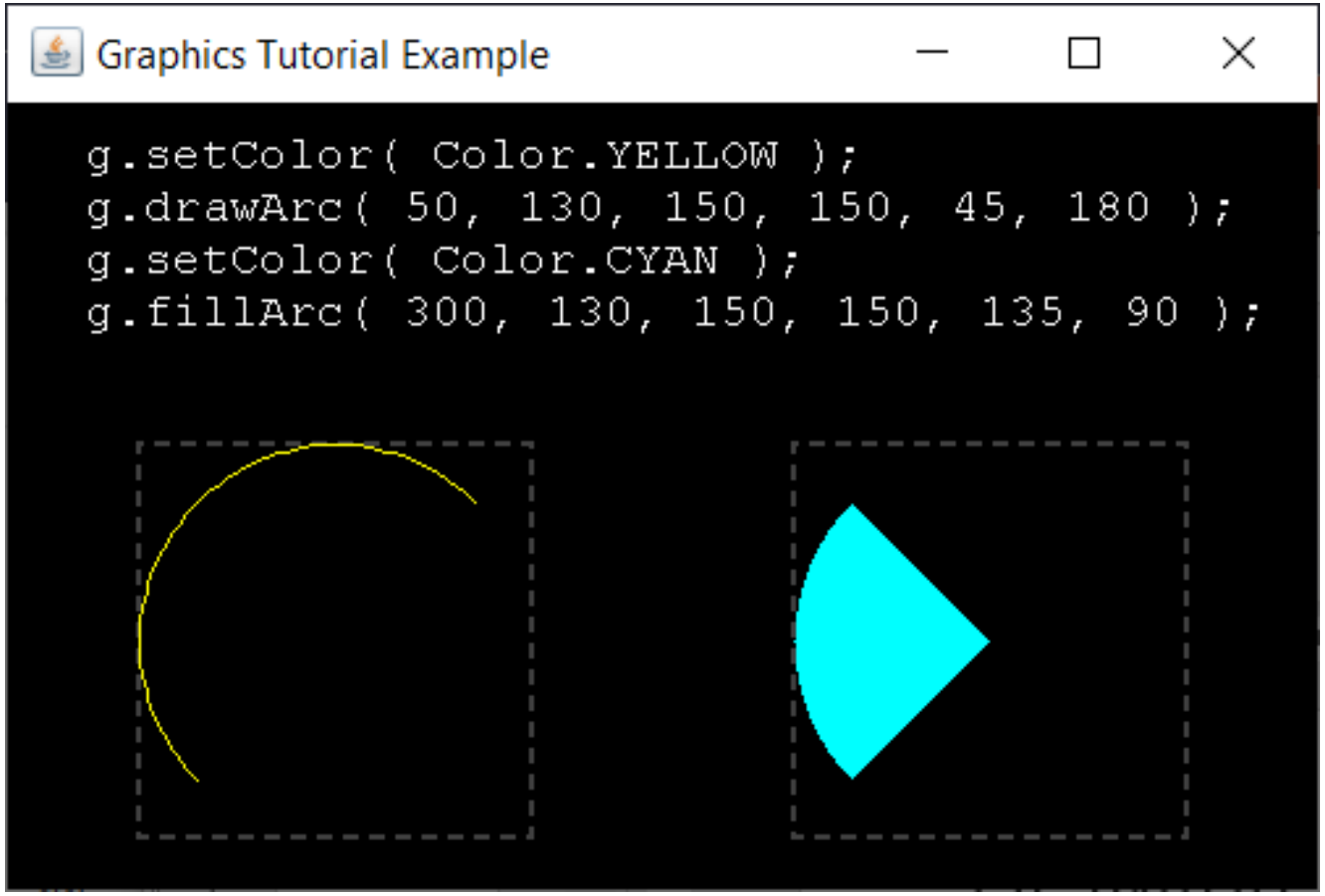
Drawing or Filling an Arc:



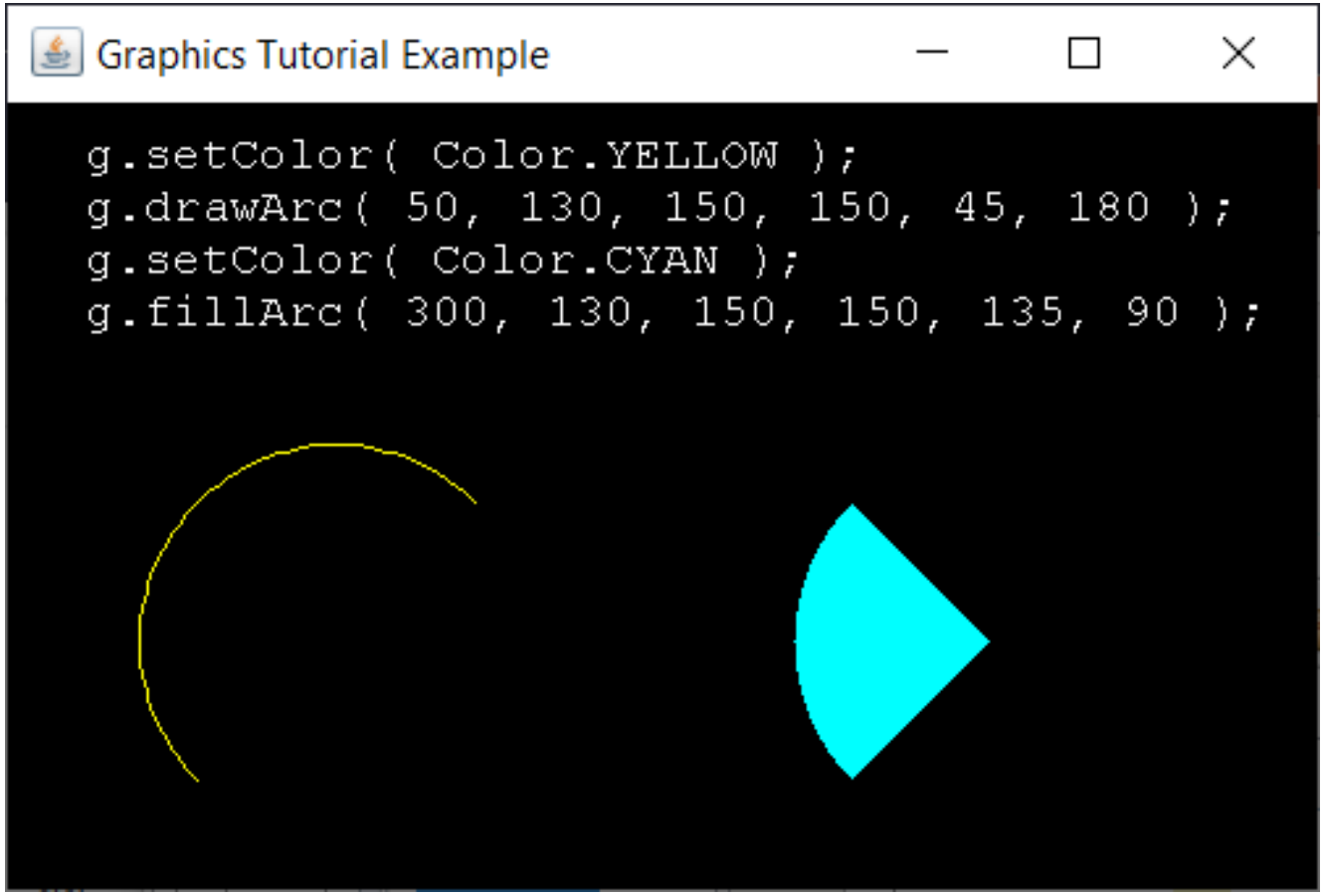
Drawing or Filling an Arc:



Drawing or Filling an Arc:



Drawing or Filling an Arc: (this is how it would appear on the screen)



Methods of the **`Graphics`** class in Java

University of Mount Union

CSC 120

Day 4