Methods & Graphics

GEEN163 Introduction to Computer Programming

"First we thought the PC was a calculator. Then we found out how to turn numbers into letters with ASCII - and we thought it was a typewriter. Then we discovered graphics, and we thought it was a television. With the World Wide Web, we've realized it's a brochure."

Douglas Adams

Making an Applet GUI

1. Create a class that extends JApplet and implements java.awt.event.ActionListener
2. Create component objects
3. Specify the layout manager
4. Add the component objects to the container
5. Add an action listener to the components that respond to user input
6. Create an actionPerformed method

Change of Thinking

- Applets do not have a main method
  - The init method is used to initialize the GUI
- Input is usually from JTextfield objects
  - java.util.Scanner is not used
- Output is usually to JLabel objects
  - System.out is not used

Interesting Objects

Many GUI applets can be created with a few simple components

- **JLabel** – Displays text in the GUI
- **JButton** – Creates a button you can press
- **JTextField** – Creates an input text box

- All of these objects are from the package javax.swing

JLabel

- A javax.swing.JLabel object is used to put text or images in a GUI
- JLabel objects usually do not have a border so you do not notice their edges
- When creating a GUI, use JLabel objects to specify any text that should be displayed

JButton

- A javax.swing.JButton object creates a button that a user can press to tell the program to do something
- The text of a JButton is often “OK” or “cancel”, but you can put anything on the button
- You need to call addActionListener for a JButton so the actionPerformed method will be called when the button is pressed

JTextField

- javax.swing.JTextField objects create a box in the GUI for users to enter text
- When creating a GUI, JTextField objects are often used to get user input
- If you add an action listener to a JTextField, the actionPerformed method will be called if the user press enter in the text box

Text in a Component

- You can set the text in many GUI components when you create them
- The constructor method can have a String specifying the text to be displayed

```
javax.swing.JLabel cow = new
                        javax.swing.JLabel( “Moo”);
javax.swing.JButton bull = new
                        javax.swing.JButton(“Hit me”);
```

Changing a Component's Text

- The setText method can be used to change the text of any object that has text, such as JButtons, JLabels and JTextfields

```
javax.swing.JLabel  msg   =
        new javax.swing.JLabel("Do something");

msg.setText("Don't do something");
```

GUI Output

- When a program with a GUI wants to display a result, it usually puts the value in a JLabel object.

```
JLabel aardvark = new JLabel();
double answer = some equation;
aardvark.setText("The answer is "+answer);
```

Getting the Text

- You can get the text from a GUI component with the getText() method
- This is useful to get the text a person entered in the JTextField

```
javax.swing.JTextField llama = new
                        javax.swing.JTextField();
// in the actionPerformed method
String answer = llama.getText();
```

Which objects can use an action listener?

1. JLabel
2. JButton
3. JTextField
4. JLabel & JButton
5. JTextField & JButton

String and Numbers

- A String can contain any character on the keyboard, including the numbers

```
String myGrade = "4.0";
double myScore = 4.0;
```

- You cannot do arithmetic with Strings
- A string containing numerical characters must be converted to a double or an int for use in any calculations

Converting Strings to Numbers

- A String that contains numerical characters can be converted using

```
double cow = Double.parseDouble( “2.25” );
String four = “4”;
int goat = Integer.parseInt( four );
```

- These methods will throw an exception if the string does not contain a number

Inputting Numbers

- The getText() method returns a String
- If a number is entered in a JTextField, it will have to be converted to an int or double

```
JTextField gekko = new JTextField();

String dragon = gekko.getText();
double lizard = Double.parseDouble( dragon );
```

The getText method returns a

1. int
2. double
3. String
4. All of the above
5. None of the above

Example GUI part 1

```
/*  GUI Kilometer to mile converter */
import javax.swing.*;
public class KmMile extends JApplet
    implements java.awt.event.ActionListener {
   final double MILEPERKM = 0.621371192;
   JTextField inKm    = new JTextField();
   JLabel instruct = new JLabel("Enter kilometer");
   JButton  goButton = new JButton("Convert");
   JLabel   answer    = new JLabel();
```

Example GUI part 2

```
public void init() {
    setSize(150, 75);
    java.awt.Container pane = getContentPane();
    BoxLayout where = new
            BoxLayout(pane,BoxLayout.Y_AXIS);
    setLayout( where );
    pane.add( instruct );
    pane.add( inKm );
    pane.add( goButton );
    pane.add( answer );
    goButton.addActionListener( this );
    inKm.addActionListener( this );
}
```

Example GUI part 3

```
public void actionPerformed(
        java.awt.event.ActionEvent thing) {
   String textKm = inKm.getText();
   double km = Double.parseDouble( textKm );
   answer.setText(km * MILEPERKM + " miles");
}
```

Programming Assignment

- Write a program to calculate the monthly payments for a ten year loan
- The program should have a Graphical User Interface
- Upload the Java source code to Blackboard by 4:00pm on Friday, Sept 14

No main Method

- Applets do not need a main method
- When an applet runs in a browser, the browser creates an object of the applet and then calls the **`init`** method
- There are many methods for an applet including **`start, stop`** and **`paint`**

paint Method

- The web browser calls the paint method when it wants to display an applet's graphics

```
public void paint( Graphics peacock ) {
    peacock.setColor(java.awt.Color.BLUE);
    peacock.fillRect( 10, 20, 100, 150 );
}
```

- The paint method may be called many times, particularly when you resize the window

Graphics Methods

- There are many methods that can be used to draw simple shapes on the screen
- Most Graphics methods do not return a value
- You can set the color to be used to draw a shape. This color will be used for all shapes until you change the color

- *The Graphics methods are **not** an effective tool for drawing cool interactive graphics*

Colors

- The java.awt.Color class defines colors
- You can define a color using RGB values or you can use a predefined constant
- Some Color constants are:
  - java.awt.Color.BLUE
  - java.awt.Color.GREEN
  - java.awt.Color.ORANGE
  - java.awt.Color.GRAY
  - java.awt.Color.YELLOW

Graphics Coordinates

- The screen has a coordinate system with the origin in the upper left corner
- Coordinates are given in pixels (Picture Elements)
- An X coordinate specifies the distance from the left edge
- A Y coordinate specified the distance from the top edge
- The screen size depends on the device

setColor

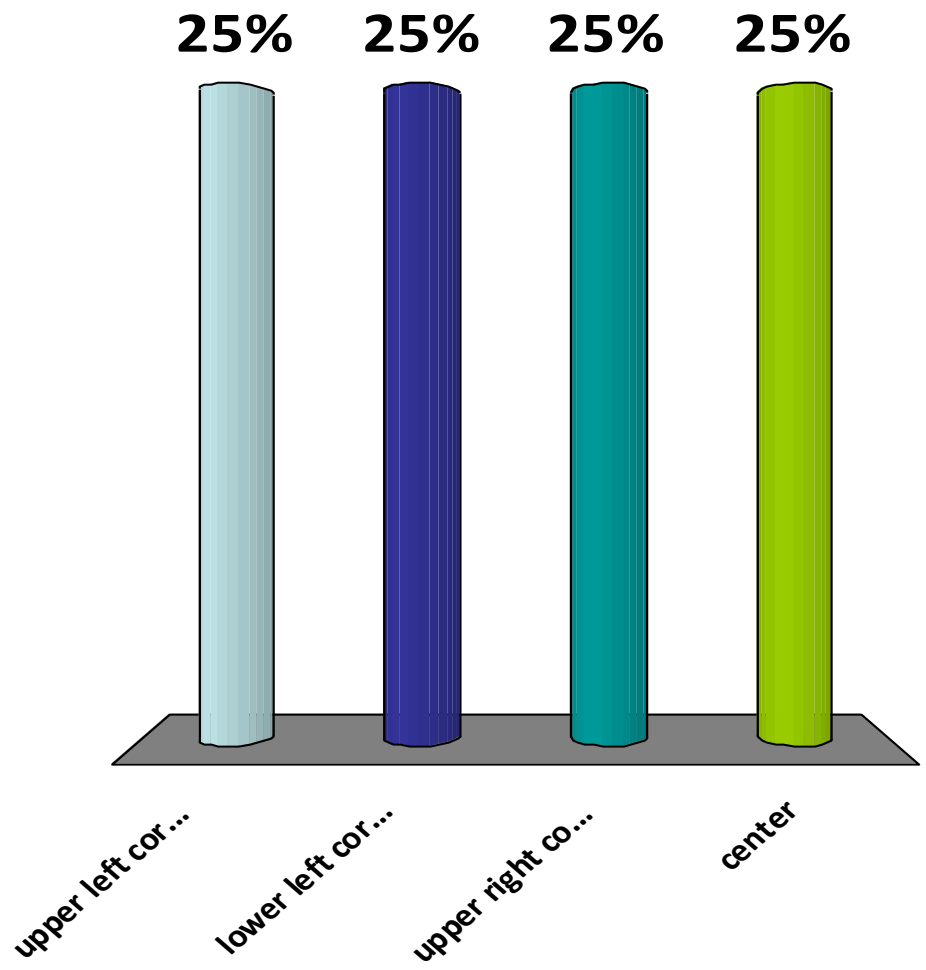
- The setColor method of java.awt.Graphics objects sets the color that will be used when drawing shapes

```
gobj.setColor(java.awt.Color.BLUE);
```

- where **`gobj`** is an object of the type java.awt.Graphics

Location 0,0 is

1. upper left corner
2. lower left corner
3. upper right corner
4. center

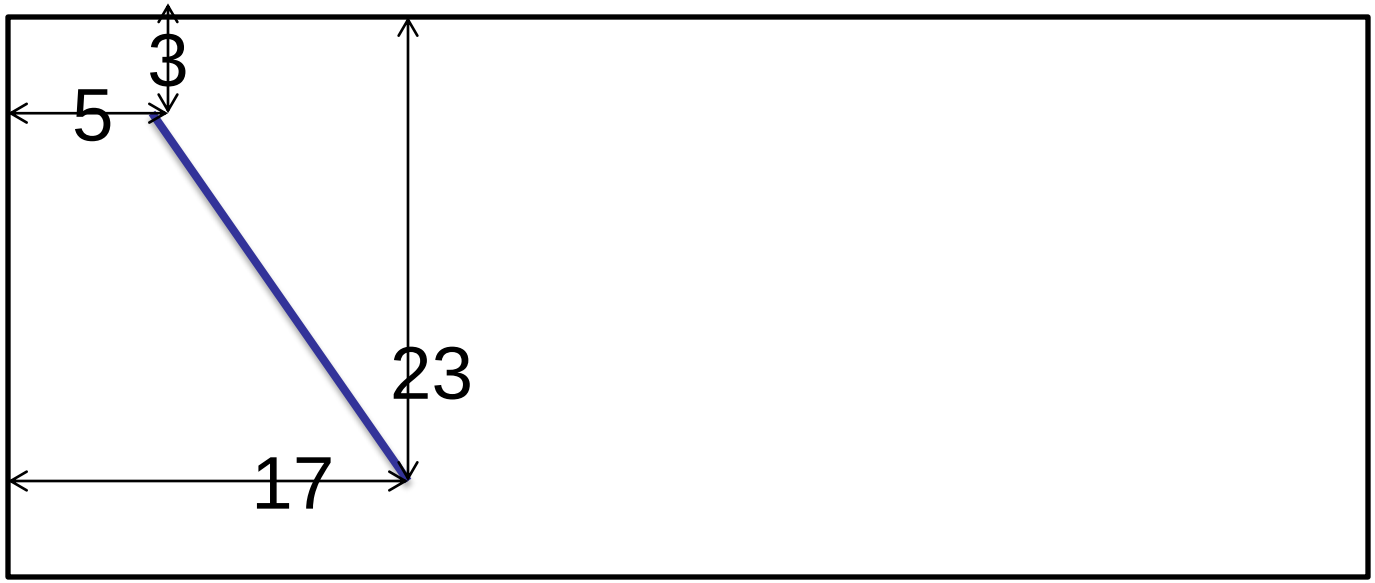


drawLine

- The drawLine method of java.awt.Graphics objects draws a line from x_1,y_1 to x_2,y_2

```
gobj.drawLine(5, 3, 17, 23);
```

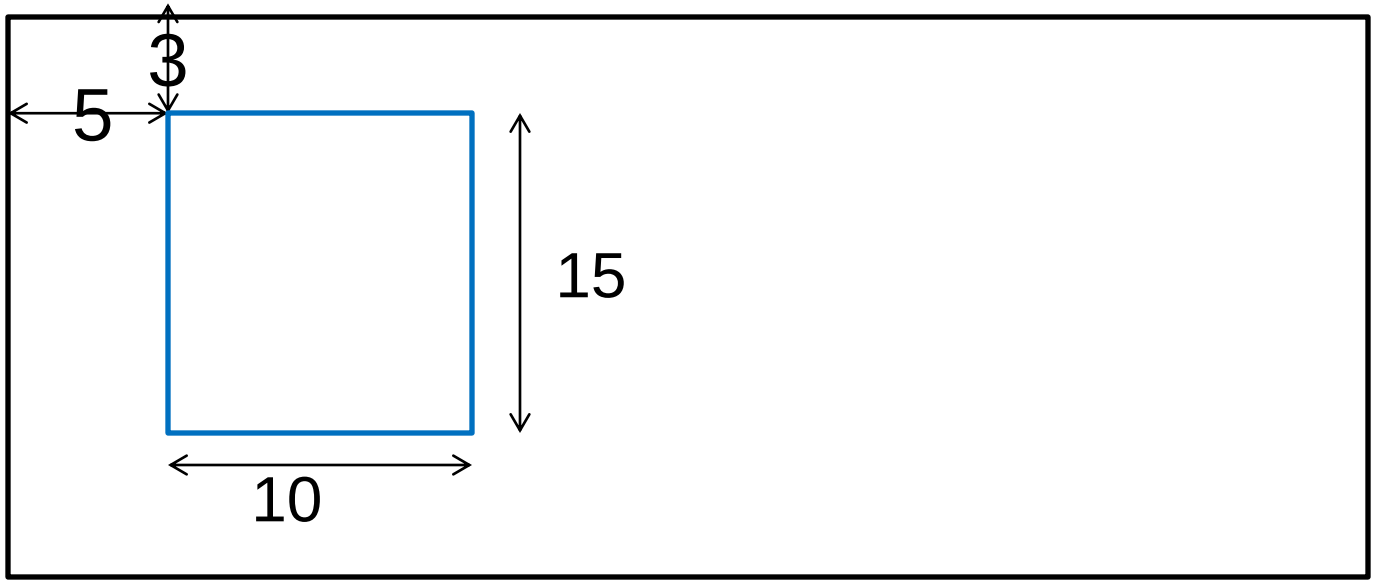
- The line is drawn using the current color



drawRect

- The drawRect method of java.awt.Graphics objects draws a rectangle of width and height whose upper left corner is x,y

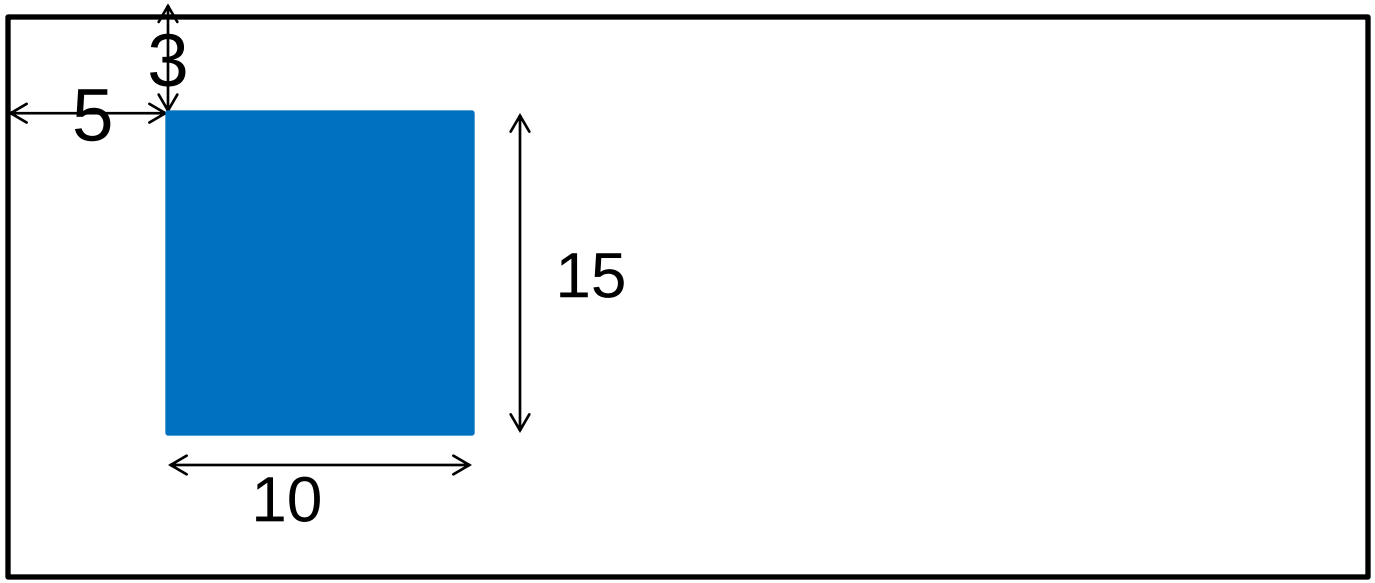
```
gobj.drawRect( x, y, width, height);
gobj.drawRect( 5, 3, 10, 15 );
```



fillRect

- The fillRect method of java.awt.Graphics objects draws a rectangle of width and height whose upper left corner is x,y and fills it with the current color

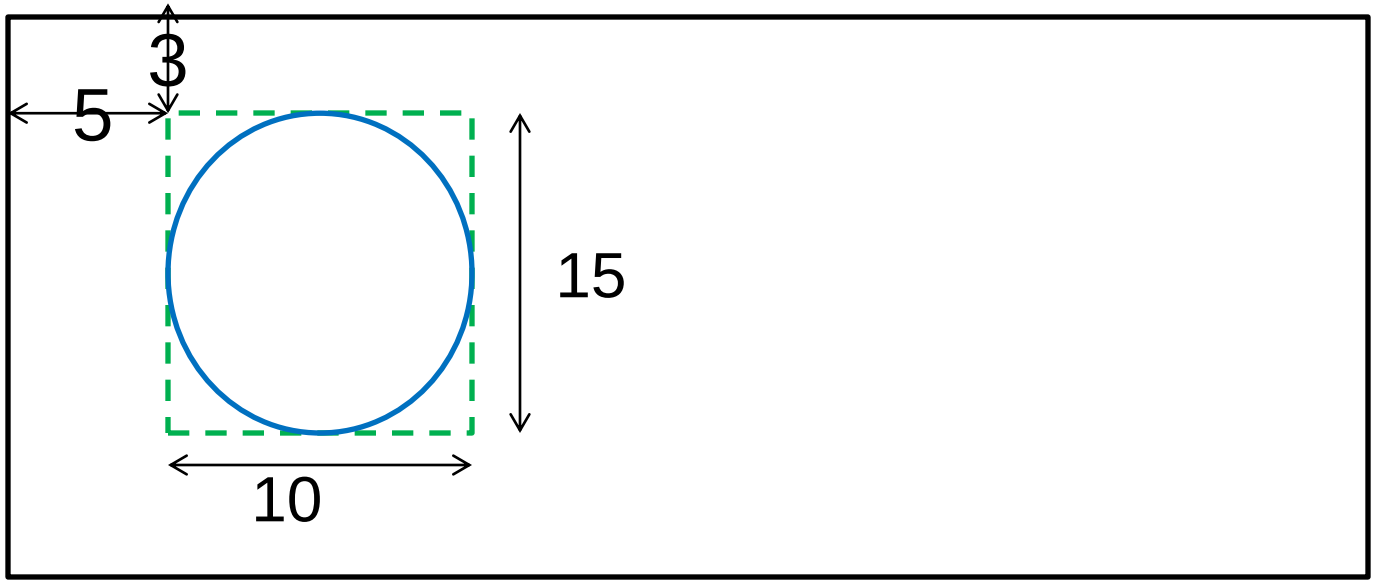
```
gobj.fillRect( 5, 3, 10, 15 );
```



drawOval

- The drawOval method of java.awt.Graphics objects draws a circle or oval to fit in a box of width and height whose upper left is x,y

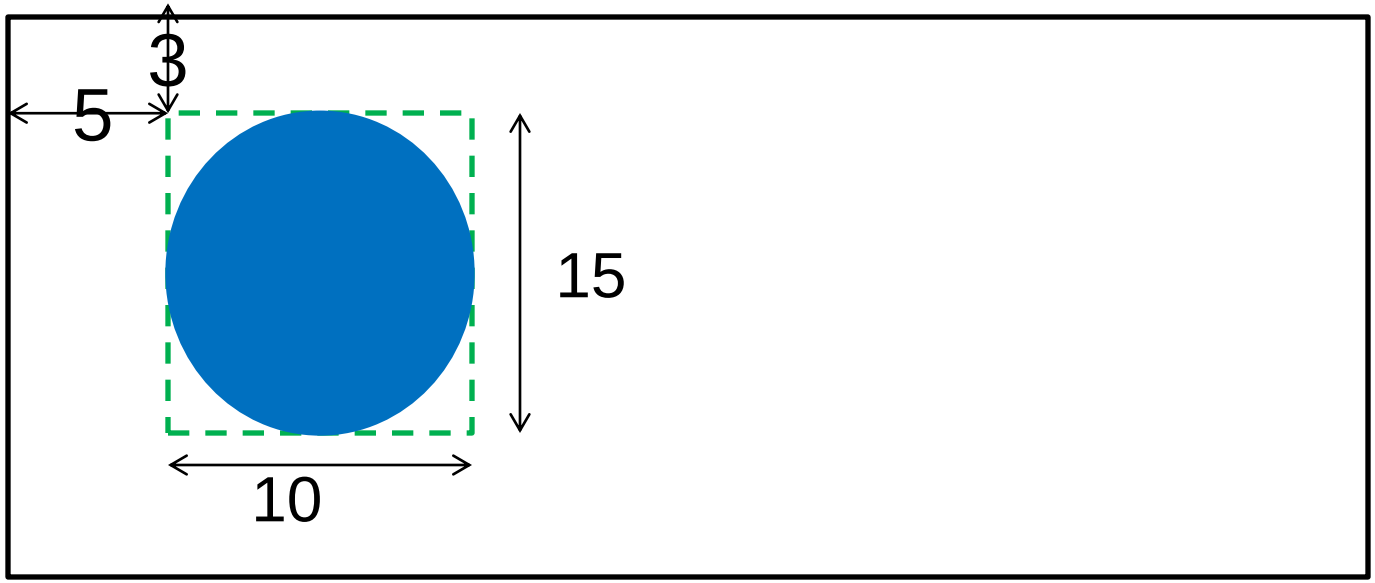
```
gobj.drawOval( x, y, width, height );
gobj.drawOval( 5, 3, 10, 15 );
```



fillOval

- fillOval is just like drawOval, but it colors in the circle with the current color

```
gobj.fillOval( 5, 3, 10, 15 );
```



drawArc

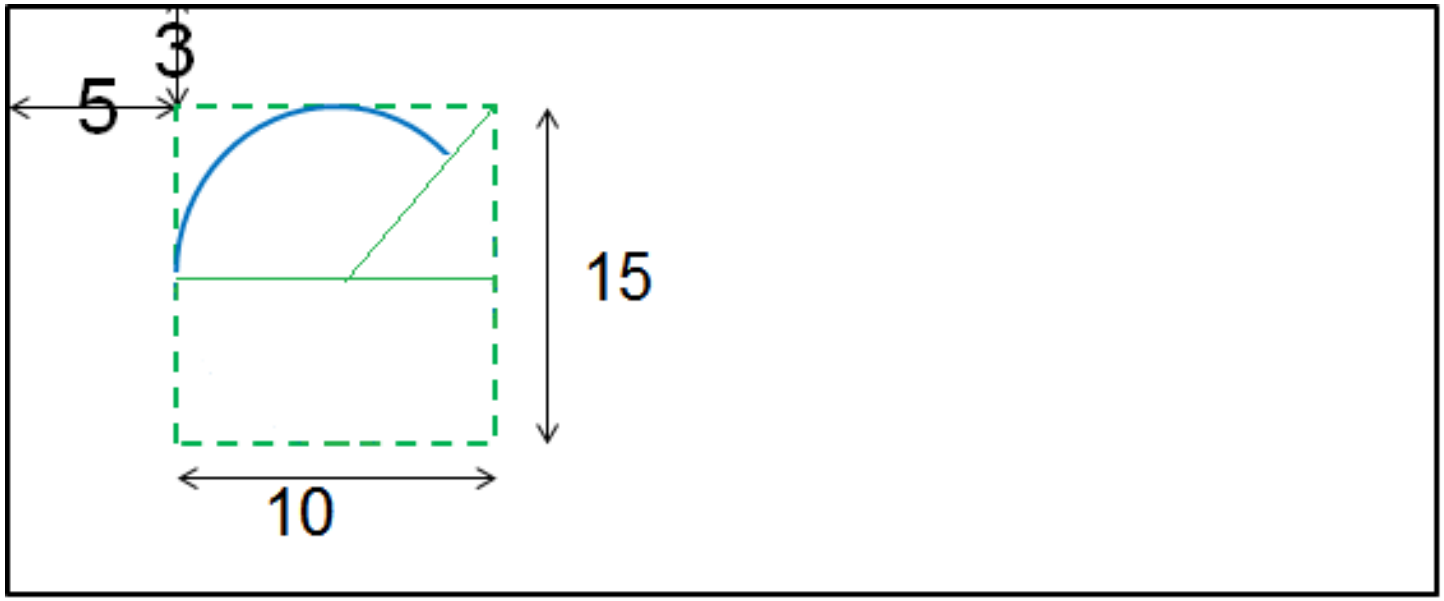
- The drawArc method of java.awt.Graphics objects draws part of an oval from start angle for arc angle degrees
- Angles are in degrees not radians
- Zero degrees in the 3 o'clock position

```
gobj.drawArc( x, y, width, height,
              startAngle, arcAngle );
```

drawArc

- The drawArc method of java.awt.Graphics objects draws part of an oval from

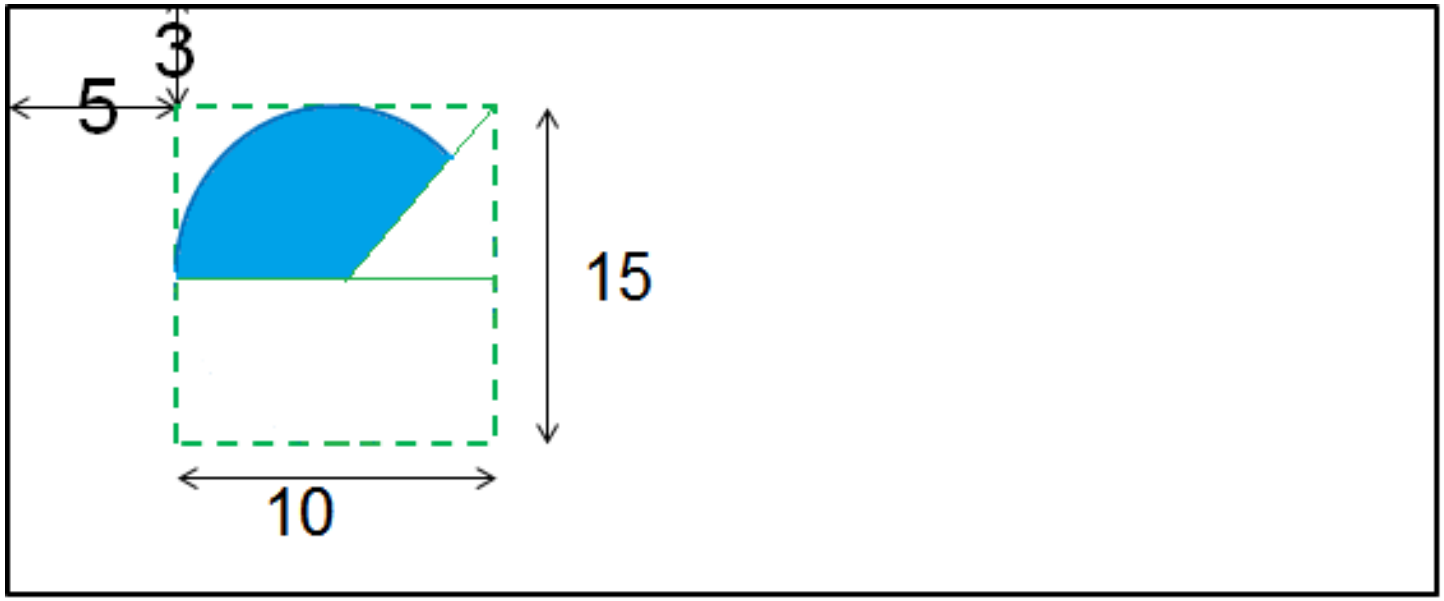
```
gobj.drawArc( 5, 3, 10, 15, 45, 135);
```



fillArc

- The fillArc method is just like the drawArc method except it fills the arc with the current color

```
gobj.fillArc( 5, 3, 10, 15, 45, 135);
```



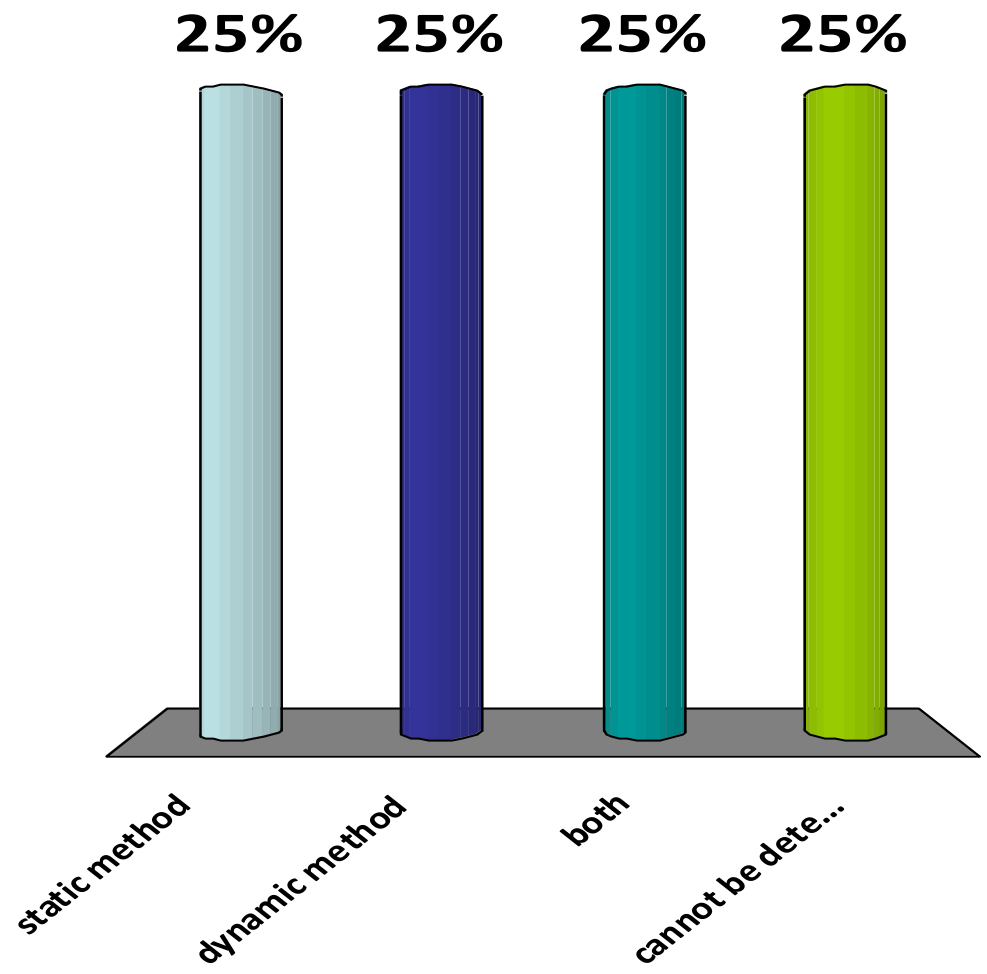
drawString

- The drawString method of java.awt.Graphics objects writes the text of a String starting at the specified x,y location

```
gobj.drawString( String, x, y);
gobj.drawString("Whatever", 5, 3 );
```

drawString is a

1. static method
2. dynamic method
3. both
4. cannot be determined



Drawing Pictures

- You can draw simple pictures using the methods of the Graphics
- If shapes overlap, the shape from the later method call will be on top