

Lesson 2-2: Strings, Graphics Revisited

Computer Science 46A: Introduction to Programming

San José State University

Announcements

- Homework #1 will be uploaded to Canvas
- Lab #2 is tomorrow (2/7)

Learning Objectives

By the end of this lesson, you should be able to:

1. Execute common string methods
2. Describe the components of a method declaration
3. Utilize common string accessor methods

String Methods

The **String** Class

- In the previous example, we looked at the **Day** class which we provided for our program **DayProg**
- In Java, there are several classes which are automatically imported for any program
- Example: the **String** class
- For these classes, how can you see the Constructors and Methods?
- Answer: [API Documentation](#)

Five Common `String` Methods

- `public int length()`
- `public String toUpperCase()`
- `public String toLowerCase()`
- `public String replace(char oldChar, char newChar)`
- `public String replace(String oldString, String newString)`
- `public char charAt(int index)`

String Method: length()

- The length method tells you how many characters are in your string
- For example:

```
String testString = "Hello";  
System.out.print(testString.length());
```

Output:
5

Question:

What is the output of the following code?

```
String testString = "Hello, again!";  
System.out.print(testString.length());
```

a) 10

b) 12

c) 13

d) 15

String Methods: toUpperCase() and toLowerCase()

The toUpperCase() and toLowerCase() methods will change the case of letters in your string

```
String testString = "Hello!";  
System.out.print(testString.toUpperCase());  
System.out.print(testString.toLowerCase());
```

Output:

HELLO!

hello!

String Method: replace()

The replace method allows you to change a subset of your string to a different value

```
String testString = "Hello!";  
System.out.print(testString.replace('H','Y'));  
System.out.print(testString.replace("ll", "x"));
```

Output:

Yello!

Hexo!

The replace method is an example of an overloaded method – it can take two different types of inputs

Question: charAt()

The charAt method will return the character at a given point in your string

What is the output of the following code?

```
String testString = "Hello";  
System.out.print(testString.charAt(4));
```

A) H

B) e

C) l

D) o

In Java, we start counting with 0

H	e	l	l	o
0	1	2	3	4



0 will return the first letter

```
String testString = "Hello";  
System.out.print(testString.charAt(0));
```

Output:

H

Question: String Replace

What is the output of the following code?

```
String testString = "Hello!";  
System.out.println(testString.replace('H','Y'));  
System.out.print(testString);
```

A) Yello!
Hello!

B) Yello!
Yello!

All string
methods are
accessors!

Tester Programs

- A tester program is a separate program that you can use to test how your program works
- In the beginning of this class, we will edit or create tester programs to test out different methods
- Later on, when you write your own classes, you will use tester programs that I provide in order to test your code

Participation Exercise 2-2a: **StringTester**

Create a Tester class to call methods from the String class

```
Java
Expected: Java
CS 46A
Expected: CS 46A
4
Expected: 4
6
Expected: 6
JAVA
Expected: JAVA
cs 46a
Expected: cs 46a
SE 46A
Expected: SE 46A
CS46A
Expected: CS46A
Java
Expected: Java
CS 46A
Expected: CS 46A
```

A Tester class displays the output of a called method

And a print-out of the expected output

If the two match, then you know the program is working as expected

A Return to the Graphics Package

Participation Exercise 2-2b Description:

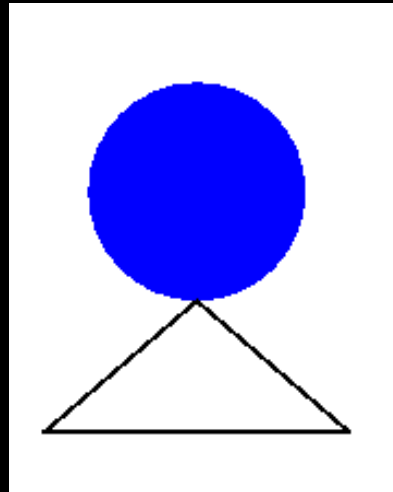
CircleOnTriangle

Goal: Write a program to draw a blue circle on top of a triangle.

The circle should be centered at (45,45) and have a radius of 25

The triangle should have a height of 30 and a base of 70

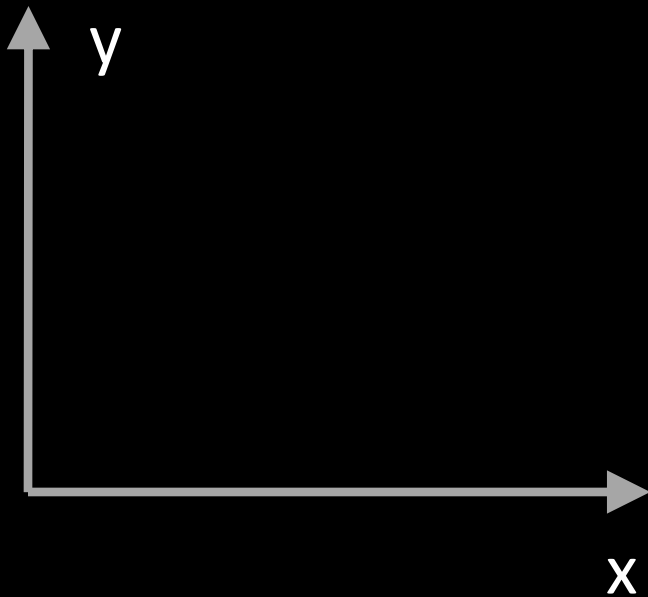
Expected output:



Note: this may seem like a silly example, but all computer graphics on apps, video games, etc call on pre-written classes to render the images that you see on your screen!

Reminder: A Note on Coordinates

In your math class:



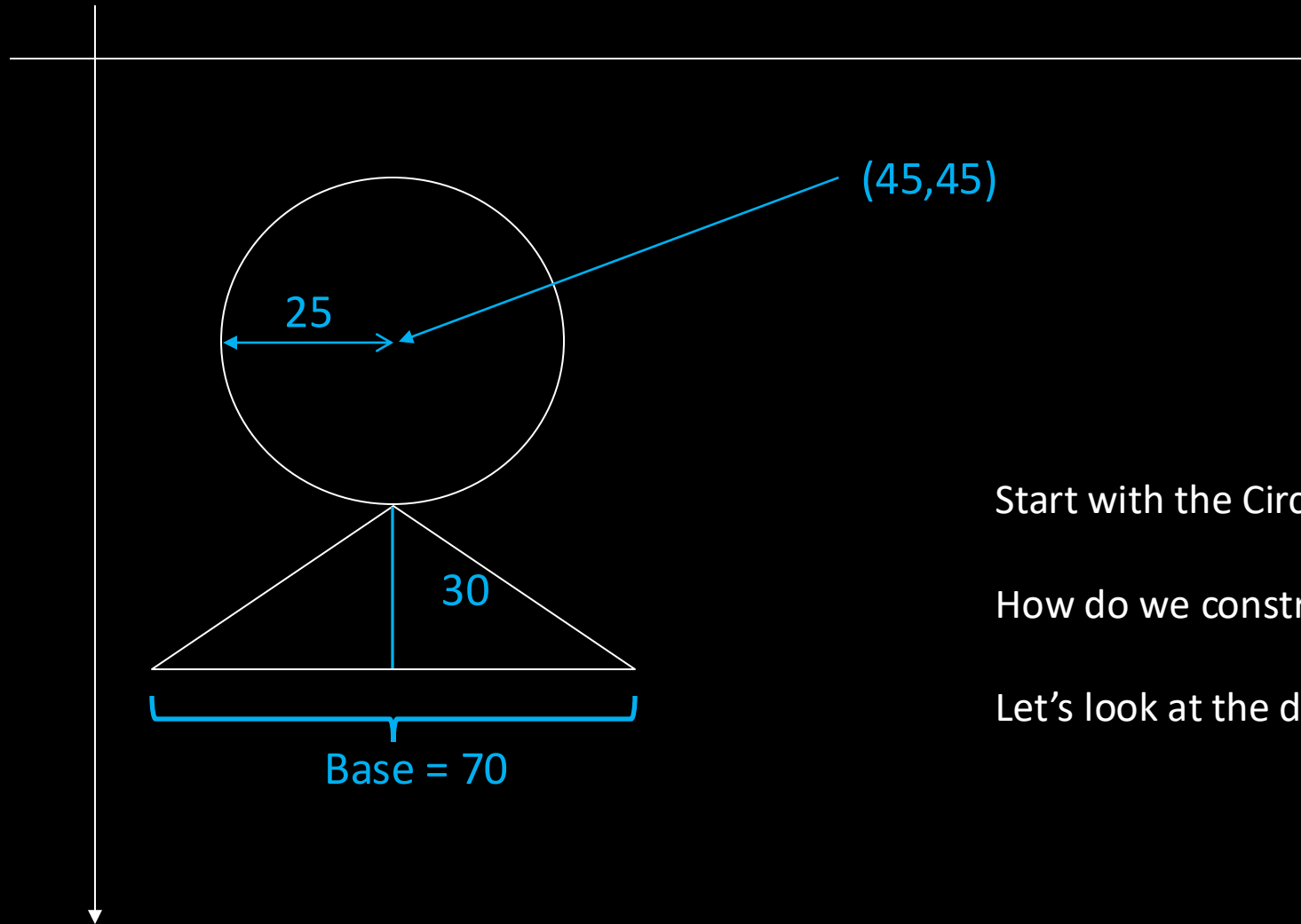
Cartesian grid, y-direction is upwards

In your CS class:



Computer Science visualization,
y-direction is downwards
(across most programming languages!)

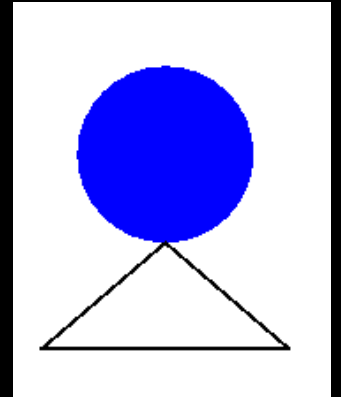
Mapping out the Dimensions



Start with the Circle

How do we construct an Ellipse object?

Let's look at the documentation!



Circle:
radius: 25
center: (45, 45)

Triangle:
height: 30
base: 70

Question: Ellipse Constructor

Which four arguments are passed to the Ellipse constructor?

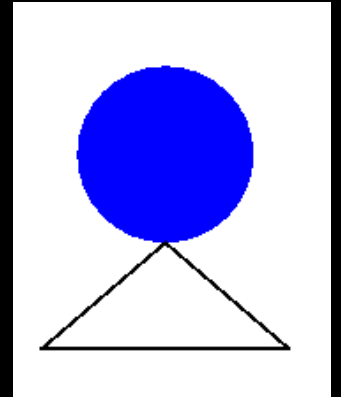
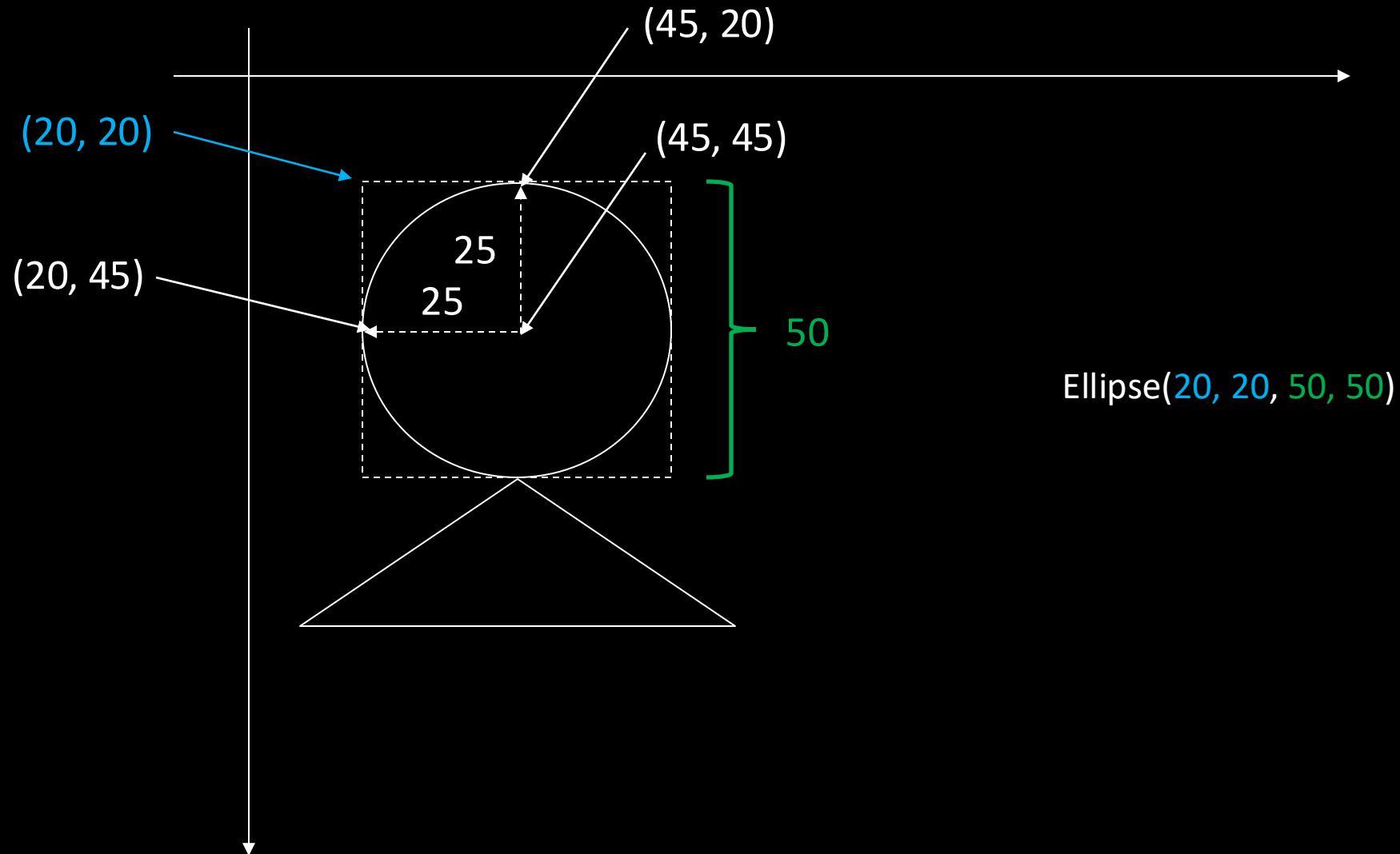
- A) Center x, center y, radius a, radius b
- B) Center x, center y, diameter a, diameter b
- C) Upper left x, Upper left y, radius a, radius b
- D) Upper left x, Upper left y, diameter a, diameter b

Documentation for Ellipse

```
/**  
    Constructs an ellipse.  
    @param x the leftmost x-coordinate  
    @param y the topmost y-coordinate  
    @param width the width of the bounding box  
    @param height the height of the bounding box  
*/  
public Ellipse(double x, double y, double width, double height)  
{  
    }
```

Question: How can we tell this is a constructor for an object in the class ellipse?

Mapping out the Dimensions: The Circle



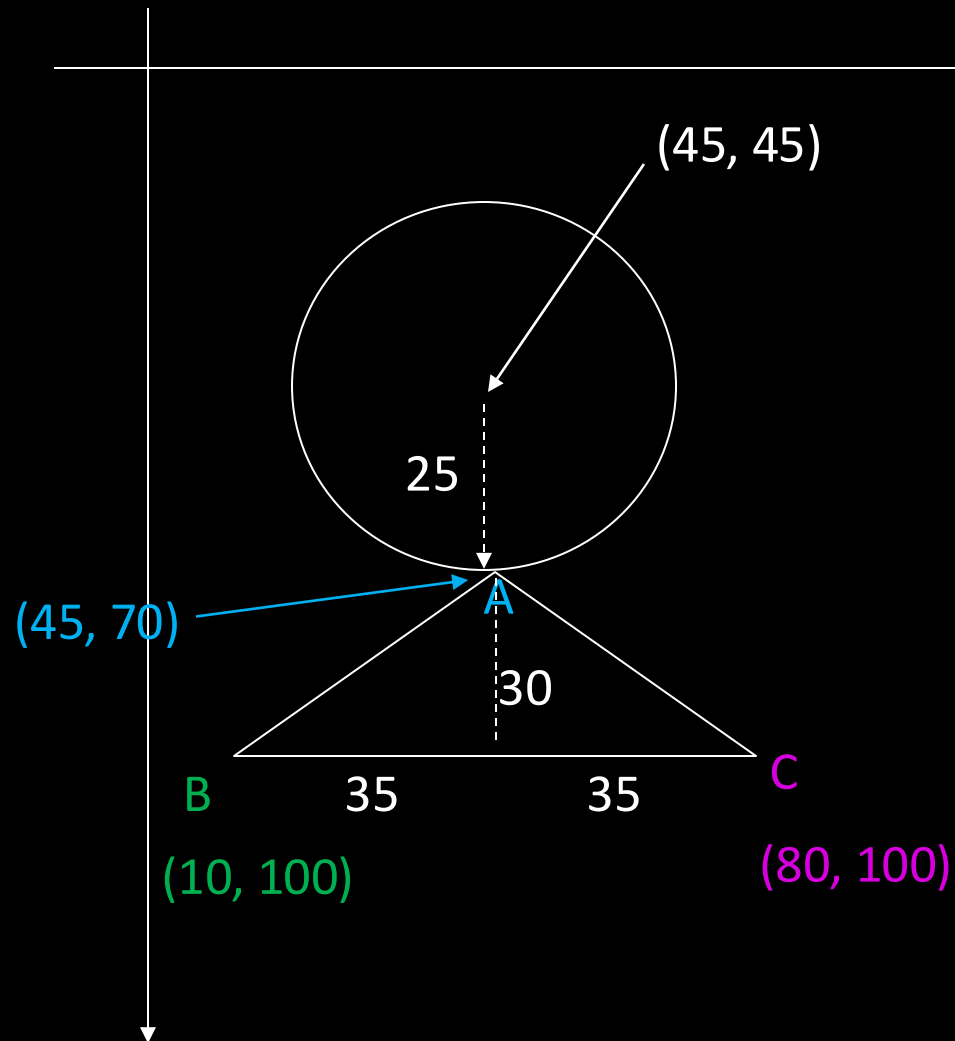
Circle:
radius: 25
center: (45, 45)

Triangle:
height: 30
base: 70

Documentation for Line

```
/**  
    Constructs a line with a given starting and ending location.  
    @param x1 the x-coordinate of the starting point  
    @param y1 the y-coordinate of the starting point  
    @param x2 the x-coordinate of the ending point  
    @param y2 the y-coordinate of the ending point  
*/  
public Line(double x1, double y1, double x2, double y2)
```

Mapping out the Dimensions: The Triangle



Three Lines:

A-B

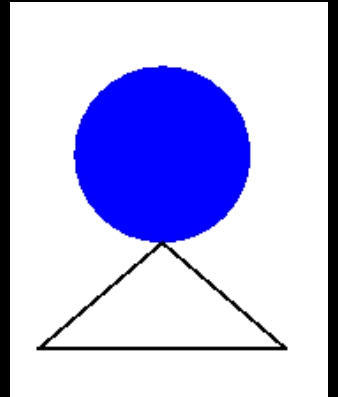
Line (45, 70, 10, 100)

A-C

Line (45, 70, 80, 100)

B-C

Line (10, 100, 80, 100)



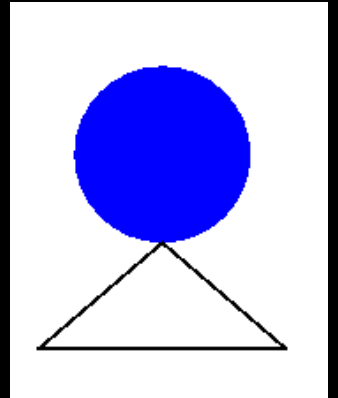
Circle:
radius: 25
center: (45, 45)

Triangle:
height: 30
base: 70

The Final Piece: Adding Color

- So far, we've determined the information to draw the circle and the triangle
- The final piece we will need is to fill in our circle with a blue color
- For a default color (black) we can use the `.fill()` method of `Ellipse`
- To change the color, we change the default color of the `Ellipse` using the `color` package, and then fill it :

```
circle.setColor(Color.BLUE);
```
- Note: you must call the `draw` method BEFORE the `fill` method



Circle:
radius: 25
center: (45, 45)

Triangle:
height: 30
base: 70

Put it all together

Follow this pseudo-code to construct your image:

1. Construct a new Ellipse object
2. Change its color, fill it, and draw it to the canvas
3. Construct a new Line object for line A-B and draw it to the canvas
4. Construct a new Line object for line A-C and draw it to the canvas
5. Construct a new Line object for line B-C and draw it to the canvas
6. Compile, debug, and run your program
7. When you finish, compare notes with your neighbor and upload to Canvas

Submit Participation Exercise 2-2a:

StringTester

Create a Tester class to call methods from the String class

```
Java
Expected: Java
CS 46A
Expected: CS 46A
4
Expected: 4
6
Expected: 6
JAVA
Expected: JAVA
cs 46a
Expected: cs 46a
SE 46A
Expected: SE 46A
CS46A
Expected: CS46A
Java
Expected: Java
CS 46A
Expected: CS 46A
```

A Tester class displays the output of a called method

And a print-out of the expected output

If the two match, then you know the program is working as expected

Submit Participation Exercise 2-2b:

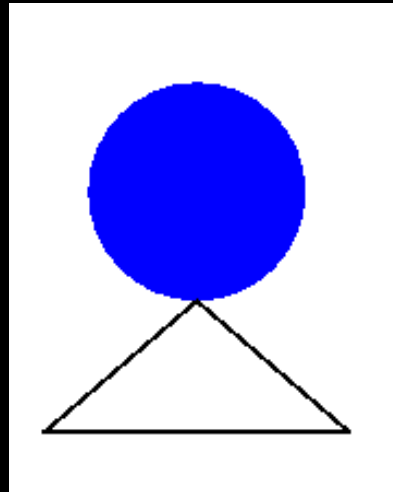
CircleOnTriangle

Goal: Write a program to draw a blue circle on top of a triangle.

The circle should be centered at (45,45) and have a radius of 25

The triangle should have a height of 30 and a base of 70

Expected output:



Note: this may seem like a silly example, but all computer graphics on apps, video games, etc call on pre-written classes to render the images that you see on your screen!

Future Problem

- In a future Homework, you will draw a Snowman like the one shown at the right
- To ensure success, be sure to follow the steps we did today:
 1. Draw a schematic diagram
 2. Determine how to construct each object by looking at its documentation
 3. Determine the coordinates of the arguments on your drawing

