Midterm
- You mostly did well!
  - MC Curve: 50,55 – A
    - 40,45 - B
    - 30, 35 - C
- No lecture the day before Thanksgiving. I will be at lab hours, and I will do a little review project with whoever shows up.

Graphics module
- Not a build-in part of Python
- Need to include a module to get graphics functions.
- Ours is in `ecs10graphics.py`.

Importing
- `import ecs10graphics as gr`
- Graphics functions are, eg., `gr.line()`.
- Why do we need the prefix at all? Importing lots of modules, if two of them have functions named `line()` we need to be able to tell the difference.

The Window
- Begin any graphics program with:
  ```python
  gr.begin_graphics()
  ```
- This creates the graphics window.
- We can tailor the window.
  ```python
  gr.begin_graphics(400,600,gr.Color.pink,"Picture")
  ```

Function Arguments
- `open("dataFile.txt", "r")`
- `len(myList)`
- `int("45")`
- `gr.begin_graphics(400,600)`
- The stuff in the parenthesis. Input to the function.
- Some Python functions allow arguments or no arguments, or different numbers of arguments.
- Order of arguments is important.
### Arguments By Name

gr.begin_graphics(500, 500, title="Line Drawing", background = gr.Color.blue)

- Some functions allow you to give arguments by name, rather than order.
- Good for functions with lots of arguments, many of which will often be left out.

### How do you find out...

- What the arguments to a function are?
- What functions are available in a module?
  1. Search for documentation. A link to documentation for ecs10graphics is on the Web page, with the notes for this lecture.
  2. After importing the module, you can use dir() or help() to get a list of what’s in it:

```
  dir(gr)
  help(gr)
```

### How do you find out...

- Find out about specific functions with help()

```
  help(gr.begin_graphics)
```

### The display/animation loop

- alive = True
- while alive:
  - alive = gr.sleep(.05)

- Our Python program is running in one window.
- It needs to stop and let the other window with the graphics in it run (so it can display itself, respond to the user killing it, whatever).
- This gr.sleep() command stops our program for a bit.

### gr.sleep()

- gr.sleep() pauses this program to allow graphics window to be moved, killed, etc – normal Windows behavior.
- Argument is number of seconds to pause. Here, 5/100.
- This window and the graphics window are taking turns running.
- If window is killed while the program is asleep, sleep returns False. Otherwise returns True.

### A big X

- Let's draw something
  - gr.set_Color(gr.Color.purple)
  - Picks up the purple crayon
  - gr.line(200,25,200,575,lineWidth=6)
  - Draws the line from point (200,25) to point (200,575).
  - Could specify (x,y) with floats as well.
The window

Points

- A point in the window is a place in the window described by its x and y.
- (25,25) is in the lower left, (575,575) is the upper right.
- Since a point is an ordered list of two integers (x and y), it makes sense to represent a point with a list:
  
  ```python
  point = [25,25]
  ```

Pentagon

- To make a pentagon, we need to produce points on a circle.
- How do you produce such points?
- Geometry of a circle:

  Imagine a stick (the radius) sweeping out the circle as it rotates around the center. One point on the circle for each angle.

Points on a circle

- (x,y) is a point on the circle.
- (cx,cy) is the center.
- a is the angle (between 0 and 2Pi)
- Trigonometry! Why did you sleep through that class?

  ```python
  x = cx + radius * cos(a)
  y = cy + radius * sin(a)
  ```

How to get sin() and cos()?

- The math module!
  
  ```python
  import math
  ```

  Then you can call:

  ```python
  math.sin(a)
  math.cos(a)
  ```

  a should be a float, between 0 and 2*math.pi

Draw some points on a circle
Connect the dots…

Enjoy

- Try drawing something!
- Next time, we’ll start using graphics as an example for building our own functions, and for other ways of adding things to Python.