Incremental Development

Chapter 6

Today’s Outline

- Finish up last lecture on “recursion” and the “return statement”.
- Practice Debugging.
- Incremental Development Strategy.

Incremental Development

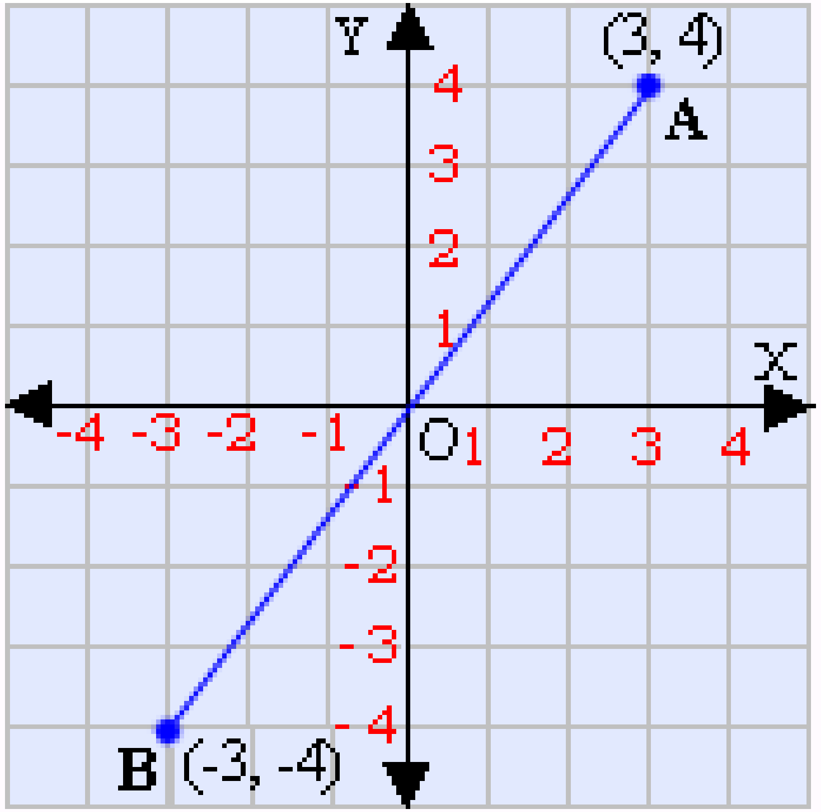
- What is it and why is it important?
  - The goal of incremental development is to avoid long debugging sessions by adding and testing only a small amount of code at a time.
- Example using the Distance Theorem

$$distance = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

- Explain this in plain language?
  - distance = the square root of the **change in x squared**, plus the **change in y squared.**

Incremental Development

- What is the distance theorem? Think coordinates…



Coordinates:

-3,-4 and 3,4.

Therefore…

$(3 - -3)^2 + (4 - -4)^2$

Incremental Development

- Example using the Distance Theorem

$$distance = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

- Let’s start coding! Start by defining the function.

```
def distance(x1, y1, x2, y2):
    return 0.0
```

Incremental Development

- So we’ve coded the following...

```
def distance(x1, y1, x2, y2):
    return 0.0
```

- Call your function using sample values. What happens and why?

```
print distance(1, 2, 4, 6):
```

- After each incremental change, we test the function again.

Incremental Development

- Add code to find the differences in x and y coordinates. How can we test this?

```
def distance(x1, y1, x2, y2):
    dx = x2 - x1
    dy = y2 - y1
    print "dx is", dx #Testing!
    print "dy is", dy

    #call function
```

- What's the output? Is the code functioning?

Incremental Development

- Add code to find the differences in x and y coordinates. We can take out the previous print statement.

```
def distance(x1, y1, x2, y2):
    dx = x2 - x1
    dy = y2 - y1
    dsquared = dx**2 + dy**2
    # Another check!
    print "dsquared is: ", dsquared
```

Incremental Development

- Add code to find the differences in x and y coordinates.

```
def distance(x1, y1, x2, y2):
    dx = x2 - x1
    dy = y2 - y1
    dsquared = dx**2 + dy**2
    result = math.sqrt(dsquared)
    return result
```

- Did we forget anything?

Incremental Development

- Forgot to import the math module! So add that, and try again.

```
import math
def distance(x1, y1, x2, y2):
    dx = x2 - x1
    dy = y2 - y1
    dsquared = dx**2 + dy**2
    result = math.sqrt(dsquared)
    return result
```

Incremental Development

- Start with a working skeleton program and make small incremental changes.
- When there is an error: it’s probably in the last change.
- Use temporary variables to refer to intermediate values so that you can easily inspect and check them.
- Once the program is working, relax, and play around with your options.

Summary

- Always document purpose, inputs, outputs!
- Incremental development is a good way to debug as you go and find out if your script is working.