Week 2 - Friday

COMP 1800

Last time

- What did we talk about last time?
- Selection statements (`if`)
- `random` library
- Monte Carlo approximation

Questions?

Else

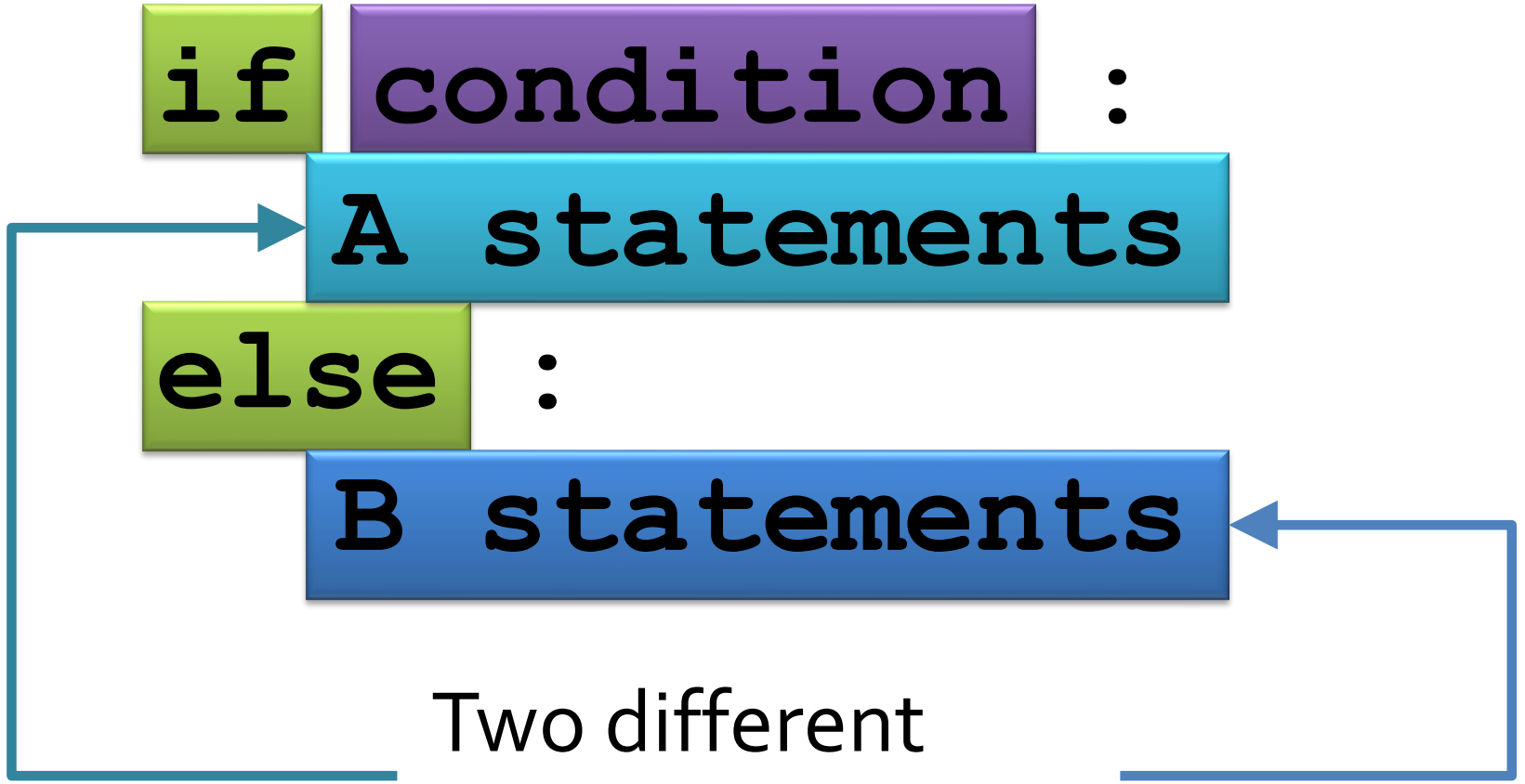
Either/Or

- Sometimes you have to make a decision
- If a condition is true, you go one way, if not, you go the other
- For example:
  - If I win the lottery,
    - I'll buy a Lamborghini
  - Otherwise,
    - I'll cry myself to sleep

Exclusivity

- Note the nature of this kind of condition
  - Both outcomes cannot happen
  - Either you buy a Lamborghini or cry yourself to sleep, never both
- For these situations, we use the **`else`** construct

Anatomy of an `if-else`



else example

```
if balance < 0:
  print('You are in debt!')
else:
  print('You have $' + str(balance))
```

if and elif

- What if you have a list of mutually exclusive conditions?
- You can tie all the possibilities together starting with **`if`**, then for each additional condition, you use **`elif`** to check it, and then you can optionally end with an **`else`** if none of the other conditions were met

```
if index == 1:
    print('First')
elif index == 2:
    print('Second')
elif index == 3:
    print('Third')
else:
    print(str(index) + "th")
```

Back to Monte Carlo

Recall the `random` library

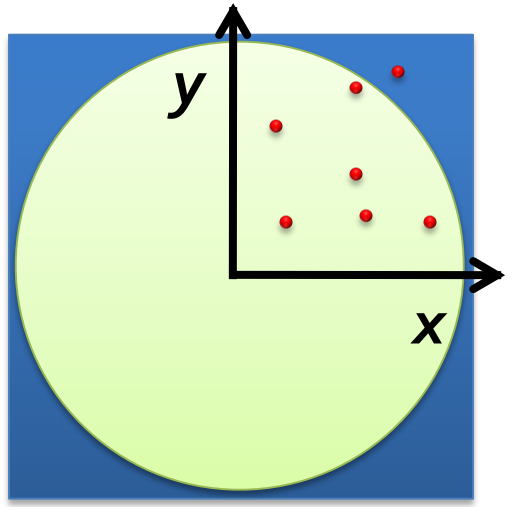
- The **`random`** library lets us produce random numbers
- It has two functions that will be useful to us:
  - **`randint(a, b)`**: Returns a random integer **`n`** where **`a`** ≤ **`n`** ≤ **`b`**
  - **`random()`**: Returns a random floating-point value from [0, 1)
- To use them, import **`random`** and then call the functions qualified by **`random`** followed by a period:

```
import random

dice = random.randint(1, 6)
percentage = random.random()
```

Monte Carlo approximation of π

- We can do something called a Monte Carlo approximation of π
- We "throw" darts at a 1 x 1 square in the upper right corner of a circle with radius 1
- We count the ones that fall inside the circle and divide by the total darts thrown
- That fraction is an estimation of the area of one fourth of the circle
- By multiplying by 4, we approximate π



Monte Carlo approximation function

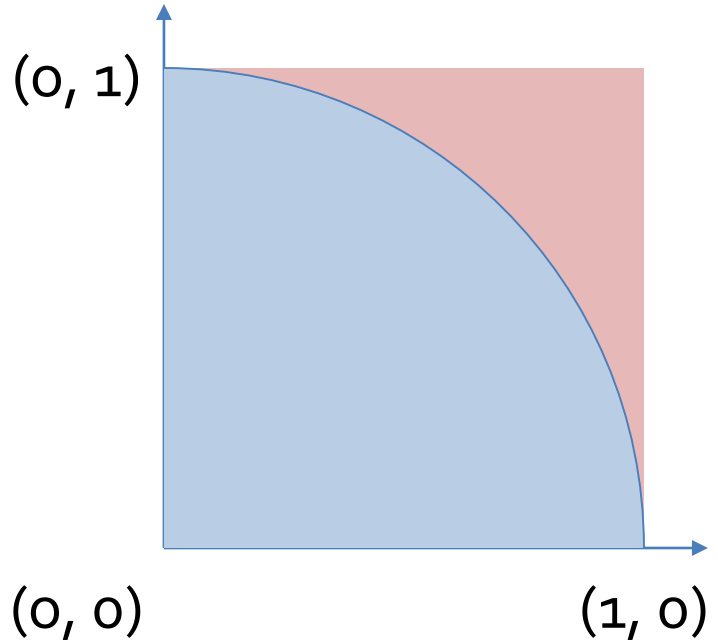
- Here is a function that performs the Monte Carlo approximation:

```
import random

def monteCarlo(darts):
    hits = 0
    for i in range(darts):
        x = random.random()
        y = random.random()
        if x*x + y*y <= 1.0: # see if dart is in circle
            hits += 1
    return 4.0 * hits / darts
```

Drawing the Monte Carlo approximation

- We can use the **`turtle`** package to draw the darts that we "throw" when doing the Monte Carlo approximation
- We need to do two things:
  - Scale the screen so that it's the right size for the dots we're drawing
    - We want the screen to go from (0, 0) up to (1, 1)
  - Draw a blue dot when the dart is in the circle and a red dot when it's not



Scaling the screen

- How far do the coordinates stretch on the normal turtle screen?
  - Who knows?!
- However, we can get the screen with the following command (as long as we already imported **`turtle`**)

```
screen = turtle.Screen()
```

- Then, we can set the minimum x, minimum y, maximum x, and maximum y with the **`setworldcoordinates()`** method

```
screen.setworldcoordinates(0, 0, 1, 1) #(0,0) to (1,1)
```

Drawing dots

- To draw a dot at a location, use the **`goto()`** method followed by the **`dot()`** method
- Assuming you have a **`turtle`** object named **`yertle`**:

```
yertle.goto(x, y)
yertle.dot()
```

- *Before* drawing a dot, you can set the color to draw by calling the **`color()`** method with a color:

```
yertle.color('blue')
```

- Before drawing dots, put the turtle's tail up with the **`up()`** method so that it doesn't draw lines everywhere

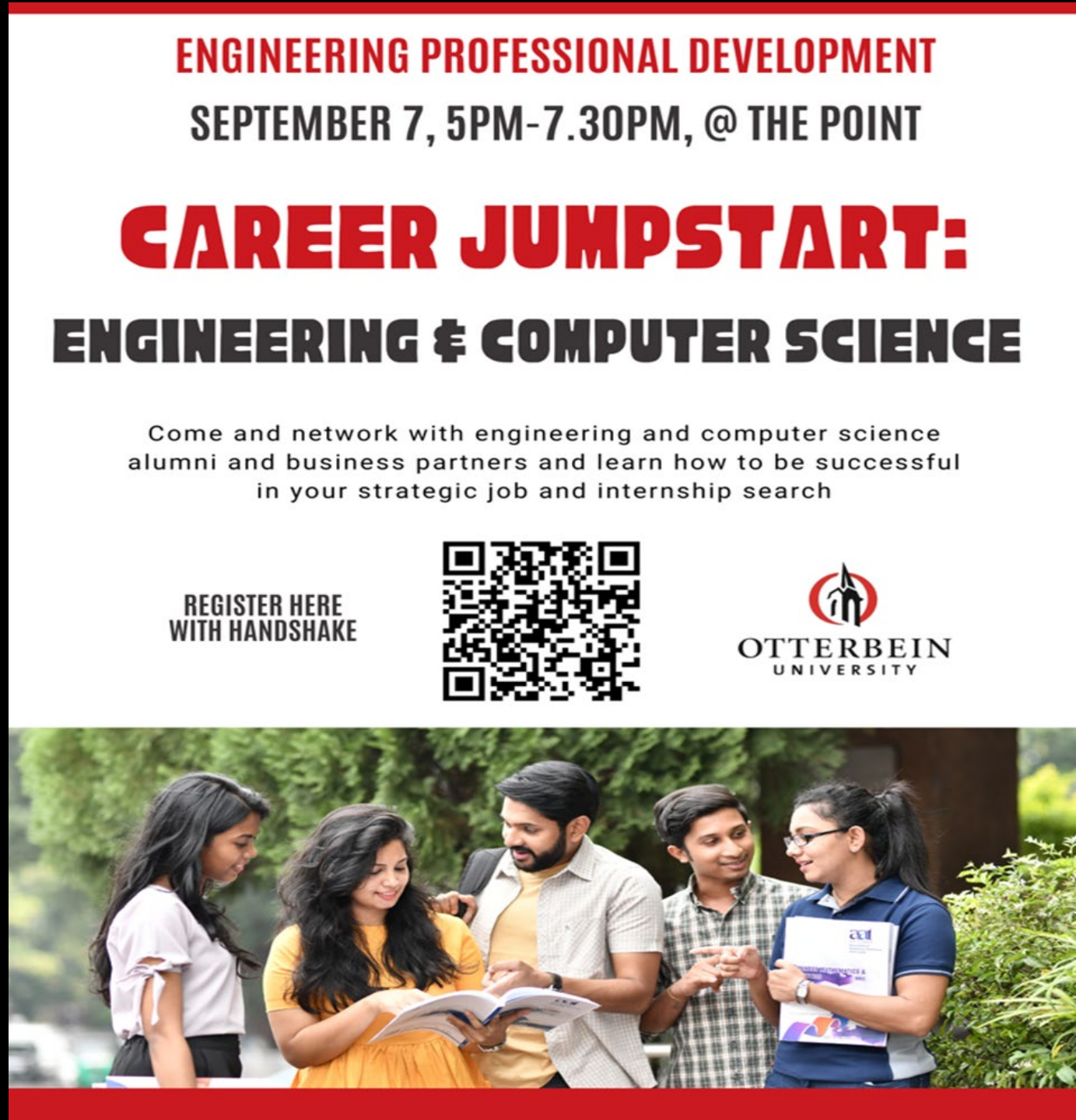
Work Time for Assignments 1 and 2

Upcoming

Next time…

- Strings
  - Concatenation
  - Repetition
  - Indexing
  - Slicing
  - Searching

- 20 employers in the fields of Engineering and Computer Science
- 20 alumni members attending
- Free professional LinkedIn headshots
- Plenty of food and great conversations
- Build new connections on LinkedIn
- Door prizes
- Network with people in your field
- Learn about possible internships
- Gain new insights about your major
- Required event for all sophomores



Reminders

- Read Section 3.2 of the textbook
- Finish Assignment 1
  - Due tonight by midnight!
- Work on Assignment 2
  - Due next Friday by midnight