Week 2 - Monday

COMP 1800

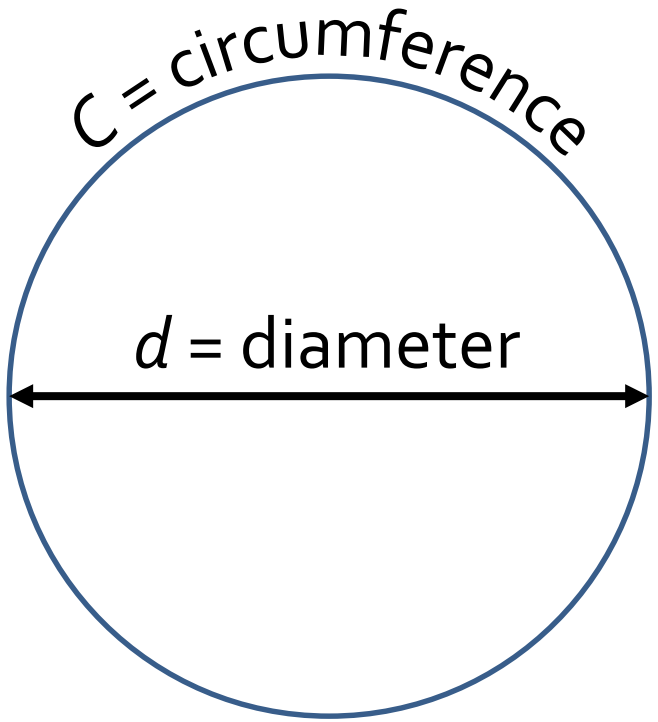
Last time

- What did we talk about last time?
- Functions
- For loops
- Work time for Assignment 1

Questions?

What is π?

- The ratio of the circumference to the diameter of any circle
  - $\pi = \frac{C}{d}$
- Ancient Greek mathematician Archimedes tried to calculate its value in the third century BCE
- Even though we have calculated trillions of digits of π, there are always more, and it's hard to figure them out



3.1415926535 8979323846 2643383279 5028841971 6939937510 5820974944
5923078164 0628620899 8628034825 3421170679 8214808651 3282306647
0938446095 5058223172 5359408128 4811174502 8410270193 8521105559
6446229489 5493038196 4428810975 6659334461 2847564823 3786783165
2712019091 4564856692 3460348610 4543266482 1339360726 0249141273
7245870066 0631558817 4881520920 9628292540 9171536436 7892590360
0113305305 4882046652 1384146951 9415116094 3305727036 5759591953
0921861173 8193261179 3105118548 0744623799 6274956735 1885752724
8912279381 8301194912 9833673362 4406566430 8602139494 6395224737
1907021798 6094370277 0539217176 2931767523 8467481846 7669405132
0005681271 4526356082 7785771342 7577896091 7363717872 1468440901
2249534301 4654958537 1050792279 6892589235 4201995611 2129021960
8640344181 5981362977 4771309960 5187072113 4999999837 2978049951
0597317328 1609631859 5024459455 3469083026 4252230825 3344685035
2619311881 7101000313 7838752886 5875332083 8142061717 7669147303
5982534904 2875546873 1159562863 8823537875 9375195778 1857780532
1712268066 1300192787 6611195909 2164201989

Attempts to approximate π

- The number π is irrational, meaning that it's neither a terminating decimal nor one that repeats with a simple pattern
- In base 10, we can only ever write an approximation of π
- In ancient times, Archimedes knew it was around 3.14
- A Chinese mathematician named Zu Chongzhi had it correct to seven digits after the decimal point in 480 CE
- In the early 16th century, it was known to 100 digits
- With electronic computers, the number of digits known in the 20th century went from 2,037 in 1949 to over 206 billion in 1999
- The 2022 record holder is exactly 100 trillion digits

math module

- In order to do some of our π calculations, we're going to need more than just basic arithmetic
- The math module has standard functions like square root and trigonometry
- To use it, you have to import it:

```
import math
```

Some `math` functions

Return type	Name	Job
Integer	`ceil( x )`	Find the ceiling of x
Integer	`floor( x )`	Find the floor of x
Floating-point	`fabs( x )`	Find the absolute value of x
Floating-point	`sin( theta )`	Find the sine of angle theta
Floating-point	`cos( theta )`	Find the cosine of angle theta
Floating-point	`tan( theta )`	Find the tangent of angle theta
Floating-point	`exp( a )`	Raise *e* to the power of a (e^a)
Floating-point	`log( a )`	Find the natural log of a
Floating-point	`pow( a, b )`	Raise a to the power of b (a^b)
Floating-point	`sqrt( a )`	Find the square root of a
Floating-point	`degrees( radians )`	Convert radians to degrees
Floating-point	`radians( degrees )`	Convert degrees to radians

Using `math` functions

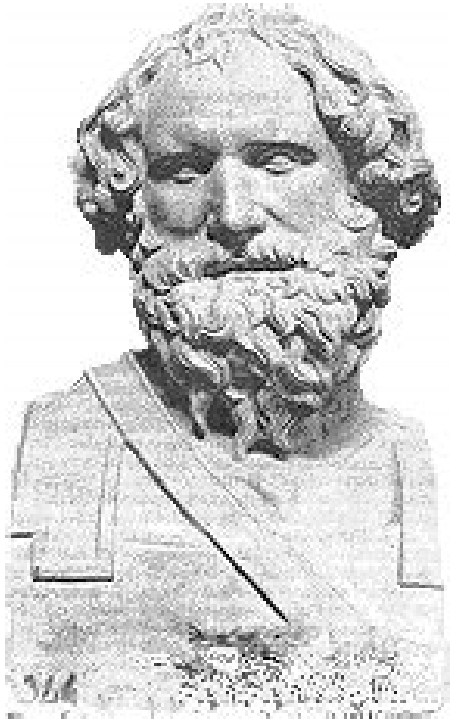
- After importing **`math`**, you still say **`math.`** before the name of a function
- For example, to compute the cosine of 2.6 radians, you can do the following:

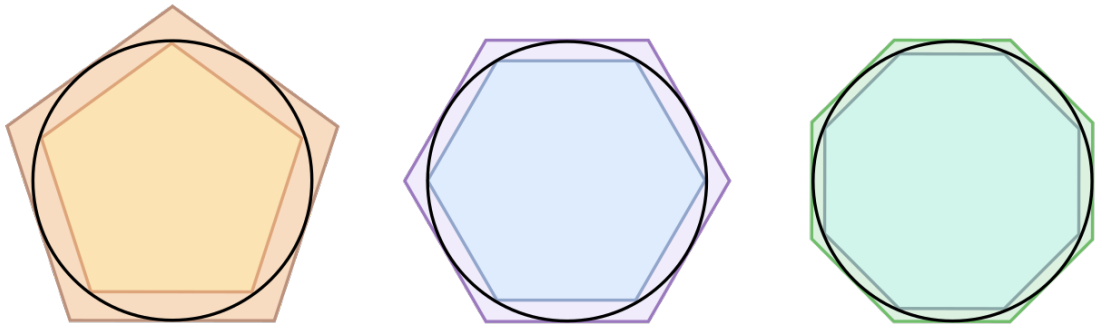
```
import math
result = math.cos(2.6)
```

- Note that all the trigonometry functions take radians, not degrees

Archimedes' method

- Start with a unit circle
- Draw polygons inside the circle
- Calculate their perimeters
- As the number of sides gets larger, the perimeters more closely approximate the circumference of the circle

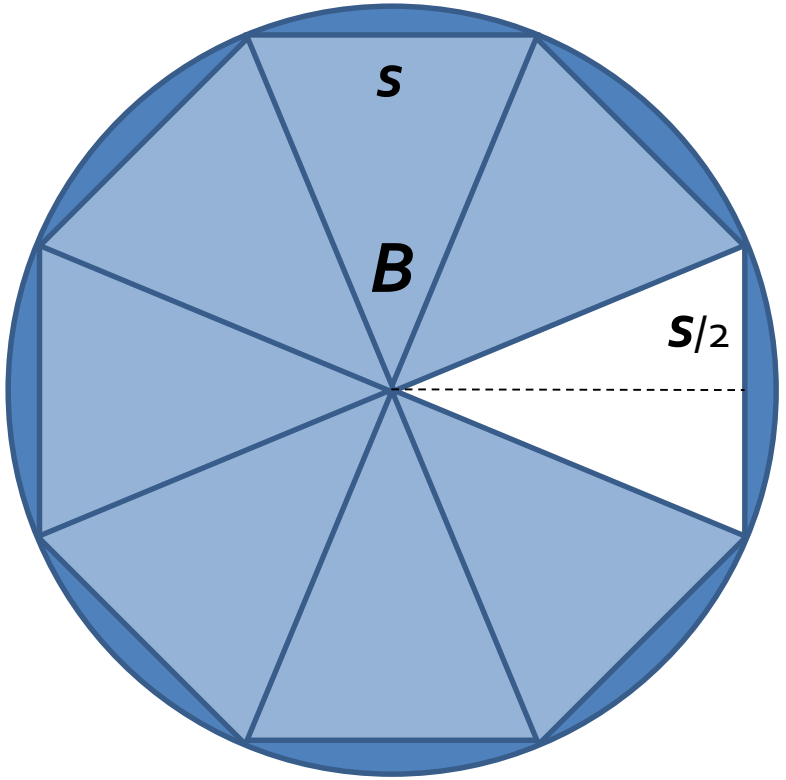




Math behind Archimedes

- An ***n***-sided polygon inside of a unit circle is made up of ***n*** triangles, fit together like a pizza
- Angle ***B*** is 360 / ***n***
- Side ***s*** is opposite ***B***
- Each triangle is made up of two right triangles with a height of ½***s***
- Because it's a unit circle, ***r*** = 1, so $\frac{s}{2} = \sin(\frac{B}{2})$
- The total perimeter is:

$$2n\frac{s}{2} = n \cdot \sin(\frac{B}{2})$$



Writing a function

- We can write a function that will take different values for sides, the number of sides that the polygon will have
- Each one will give us different approximations to π
- Let's complete the following function definition, using the math on the previous slide:

```
def archimedes(sides):
```

`return` statements

- Like most code in Python, the code inside of a function executes line by line
- Of course, you are allowed to put loops inside functions
- You can also put in **`return`** statements
- A function will stop executing and jump back to wherever it was called from when it hits a **`return`**
- The **`return`** statement is where you put the value that will be given back to the caller

The `range()` function

- The **`range()`** function produces a sequence of values that a variable will take on
- With only a single parameter **`n`**, the sequences of numbers is 0, 1, 2,…,**`n`** – 1 (but not **`n`**)

```
for i in range(100):
```

- With two parameters, **`a`** and **`b`**, the sequence starts at **`a`** and goes up to **`b`** – 1 (but not **`b`**)

```
for i in range(10,20):
```

- With three parameters, **`a`**, **`b`**, and **`step`**, the sequence starts at **`a`** and goes almost up to (but not including) **`b`**, taking steps of size **`step`**

```
for i in range(10,20,5):
```

Various approximation values

- Since we have a function, we can put it inside a **`for`** loop to see how the approximation gets better with larger numbers of sides
- We can use the third kind of **`range`** statement, starting at 8, going up to (but not including) 100, and jumping in steps of 8 as well

```
for sides in range(8, 100, 8):
  print (sides, archimedes(sides))
```

Improved approximation

- Archimedes' real method is supposed to put a polygon inside *and* outside the circle
- The average of their perimeters should be a better approximation of π than either one alone
- Let's make an improved version of our function that finds a better approximation

```
def betterArchimedes(sides):
```

Patterns

- A **design pattern** is a problem-solving technique in coding in which there is a standard way of do something that is used a lot
- **Accumulator Pattern**
  - Produce a result by iterating over a sequence of values and accumulate their sum (or other aggregation) along the way
- This example finds the sum of numbers from 1 up to 10:

```
>>> acc = 0
>>> for x in range(1, 11):
        acc = acc + x
>>> acc
    55
```

Gottfried Wilhelm von Leibniz

- Leibniz invented calculus at the same time as Newton
- He created an approximation for π:
  - $\frac{\pi}{4} = \frac{1}{1} - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \cdots$
  - $\pi = \frac{4}{1} - \frac{4}{3} + \frac{4}{5} - \frac{4}{7} + \frac{4}{9} - \cdots$
- How can we do this in Python?
- We need an Accumulator Pattern
- We also need a variable for the denominator that increases by 2 every time so that it's always odd
- We also need a way to subtract every other term

John Wallis

- John Wallis was a 17th century British mathematician who is believed to have come up with the ∞ symbol for infinity
- Of course, he also found an approximation for π
  - $\frac{\pi}{2} = \frac{2}{1} \cdot \frac{2}{3} \cdot \frac{4}{3} \cdot \frac{4}{5} \cdot \frac{6}{5} \cdot \frac{6}{7} \cdot \frac{8}{7} \ldots$

Wallis approximation

- $\frac{\pi}{2} = \frac{2}{1} \cdot \frac{2}{3} \cdot \frac{4}{3} \cdot \frac{4}{5} \cdot \frac{6}{5} \cdot \frac{6}{7} \cdot \frac{8}{7} \cdots$

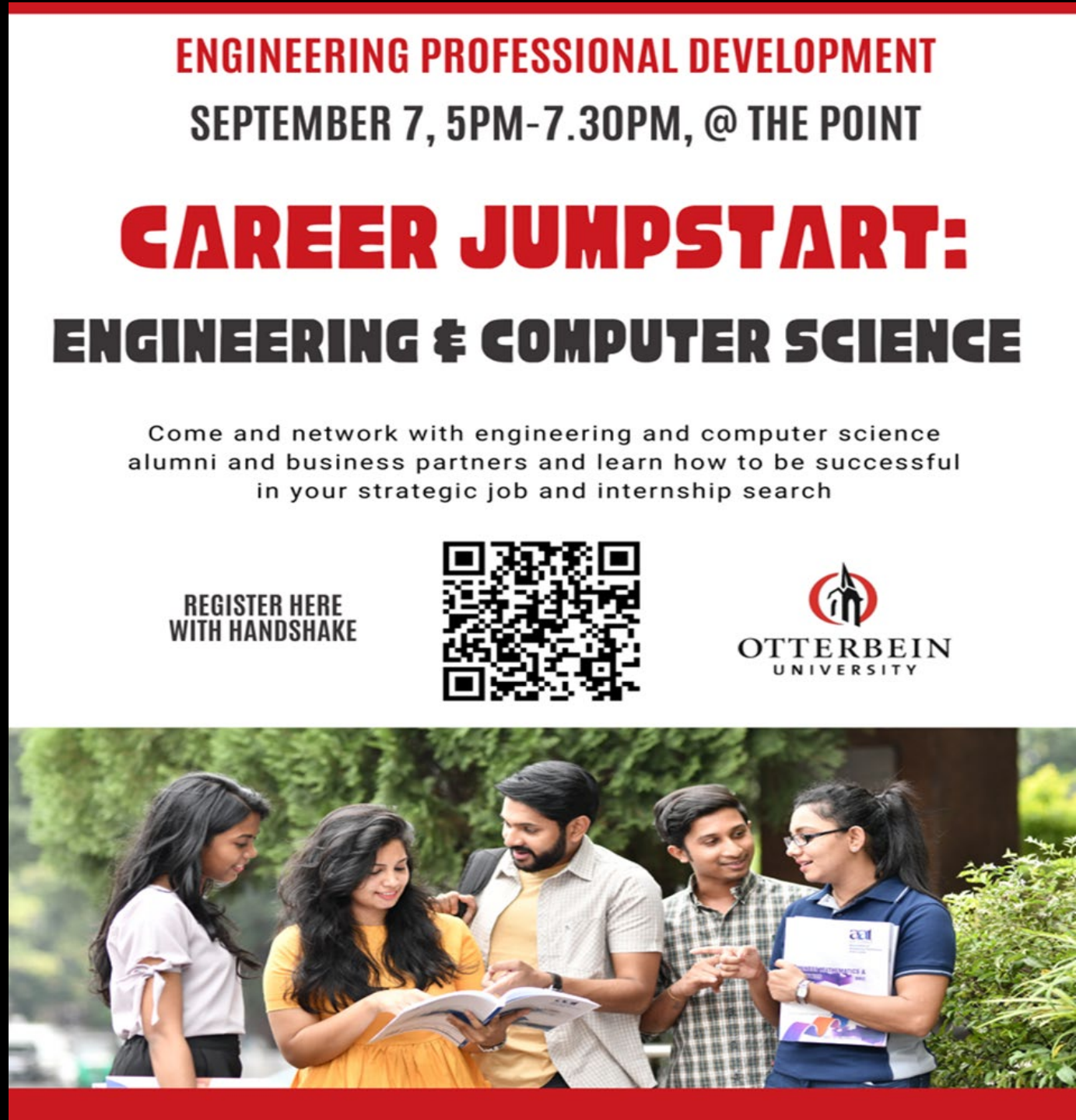
- As with Leibniz, we will use the Accumulator Pattern
- Although the Accumulator Pattern often adds things up, we'll be multiplying stuff as we go
  - When summing, we start with 0
  - When multiplying, we start with 1
- Note that both the numerator and the denominator are used twice in a row before increasing by two
  - But they change on opposite turns!

Upcoming

Next time…

- Simulation and random numbers
- We'll do a Monte Carlo approximation of π
- We'll also talk about Boolean variables and selection statements

- 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

- Keep reading Chapter 2 of the textbook
- Emergency elections for CS Club
  - Do you want to have a voice in CS Club?
  - Come vote at 4 p.m. this Wednesday, August 30 in Point 113!