

# CSCI 1370

MARCH 20, 2017

ADMINISTRATIVA

**Exam #2** will be returned by next Monday.

**Exam #3** pushed back to April 12 (see course website).

After this week, **Group Meetings** will be personal responsibility.

That is, everyone gets +4 free points towards group work (no photo documentation required). **But means you are accountable to yourself (and group members).**

# DISCUSSION: CEILING FUNCTION

(REPL.IT EXERCISE 7B)

**Ceiling function** should take in a number,  $x$ , and compute the closest whole number,  $f$ , where  $f \geq x$ .

Necessary “trick”: **casting**.

Think back to how the computer interprets and processes data based on its *type*...In this case, how can we discard decimal information?

# CONTROL STRUCTURES: WRAP-UP

# CONDITIONALS

There is one primary **conditional** structure (**if**), and complimentary components (**else if, else**).

The universal component of a conditional: **logical expression** (comparison of two entities).



# EXERCISE: MAX FUNCTION

---

- Write a function that:
  - takes two numeric parameters
  - and returns the max.

# EXERCISE: RANGE FUNCTION

---

- Write a function that:
  - takes three numeric parameters – call them **x**, **y**, and **z**
  - and returns true if the value of **x** falls between **y** and **z**.

ITERATION

There are two primary **loop** structures (`for`, `while`).

Regardless, there are **three** universal components of a loop: **initialization**, **exit condition**, and **update**.

# EXERCISE: WHILE LOOP

- With a partner, write what gets printed to the screen when the following code snippet is run:

```
int i = 1;
while (i < 10)
{
    cout << i << endl;
    ++i;
}
```

# EXERCISE: FOR LOOP

- With a partner, rewrite the following code snippet using a **for** loop structure.

```
int i = 1;
while (i < 10)
{
    cout << i << endl;
    ++i;
}
```

# EXAMPLE: USER-CONTROLLED LOOP

- Beyond counters, we can also loop based on various conditions. For example:

```
char c = 'y';
while (c == 'y')
{
    cout << "Keep looping?";
    cin >> c;
}
```

# EXERCISE: USER-CONTROLLED LOOP

- With a partner, rewrite the following code snippet to loop as long as the user wants.

```
int i = 1;
while (i < 10)
{
    cout << i << endl;
    ++ i;
}
```



# EXERCISE: PRIME FUNCTION

---

- Write a function that:
  - takes a numeric parameter – call it **x**
  - and returns true if **x** is a prime number.

# EXERCISE: ASCII ART

- Write code that produces the following console content when compiled and run:

```
*  
* *  
* * *  
* * * *  
* * * * *
```