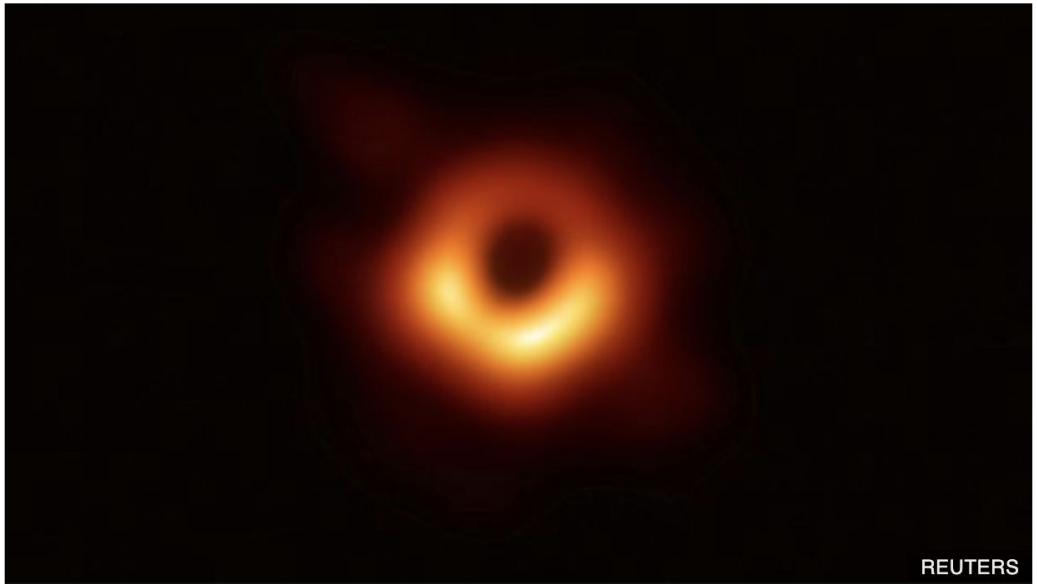
GWC Week 4

Methods (functions) and Recursion

WIT Shout-out of the Week: Katie Bouman

- American Computer Scientist specialized in computer imagery, she got an undergrad degree in Electrical Engineering from University of Michigan, then a masters from MIT in computer science
- In June of 2019 she developed an algorithm to view black holes known as “Continuous High-Resolution Image Reconstruction using Patch Priors (CHIRP)
- She was a member of the Event Horizon Telescope that produced the first image of a black hole
- She is now a professor of computing and math at California Institute of Technology



The first ever photo of a black hole, taken using a global network of telescopes

video

https://www.youtube.com/watch?v=P7n2rYt9wfU

This video was before Katie and her team were able to capture the first image of the black hole.

Warm Up!

Practice Problem

For a user input, count the number of vowels (a,e,i,o,u)

Hints: you will need one for loop that runs for the length of the string with several if statements nested inside. Use 1 string variable for the user input, and 3 int variables for the string length, vowel count, and for loop iteration.

Methods (aka Functions)

Methods (aka Functions)

```
public static int methodName(int a, int b) {
   // body
}
```


- Way to minimize the amount of code in your main() and keep your code well organized
- We've already seen several different methods
  - Console.Write()
  - Console.Read()
  - Main (string[] args)
- Methods are always followed by ()
  - Sometimes data (called PARAMETERS) are passed within the ()
  - Note: if / else-if / for are also followed by (), but are NOT methods

Modifiers

```
public static int methodName(int a, int b) {
    // body
}
```


public

The type or member can be accessed by any other code in the same assembly or another assembly that references it.

private

The type or member can be accessed only by code in the same class or struct.

protected

The type or member can be accessed only by code in the same class, or in a class that is derived from that class.

Return Type

```
public static int methodName(int a, int b) {
    // body
}
```


- Think back to data types from last week
  - int
  - double
  - bool
  - string
  - char
  - etc.
- This is what the method will send back when it is called

Method Name

```
public static int methodName(int a, int b) {
    // body
}
```


- Should be related to purpose of the method
- camelCase or use_underscores
- Will be used to access method

Parameters

```
public static int methodName( int a, int b) {
    // body
}
```


- This is the data that will be used in the method
- Must declare data type
- Can have multiple parameters
  - Just put a comma between them as shown above
- Can have 0 parameters
  - Ex: Console.ReadLine();

How to Call a Method

- Use the Method Name (write it exactly the same) and include any parameters you need
- Summing Example

Recursion

When a method calls itself (creates a loop)

Show repl and work through it by hand before running program

Practice Problems

- Create a method that takes two integers as parameters, compares them, and returns the highest value
- Create a method that recursively calls itself to increment an integer given by the user by 3 until it is greater than 100
- Write a program that takes an integer from user input (n) and either sums up all numbers from 1 to n AND calculates n!