



Software Workshop No. 1

UCSB Robotics, Fall 2020 | Alex Mei

Installation Check-In

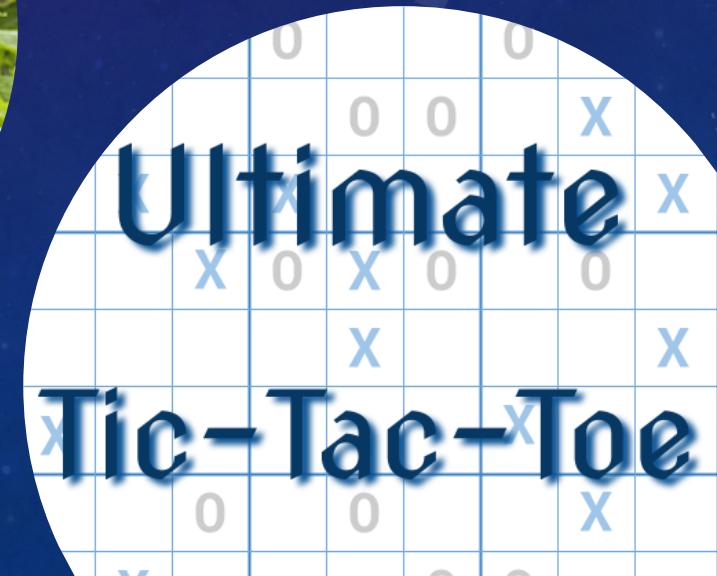
ABOUT ME

- 2nd Year CS Major
- ULA + ERSP
- Procore Intern



ABOUT ME

- Cook
- Survivor
- Games



LIL' INTROS

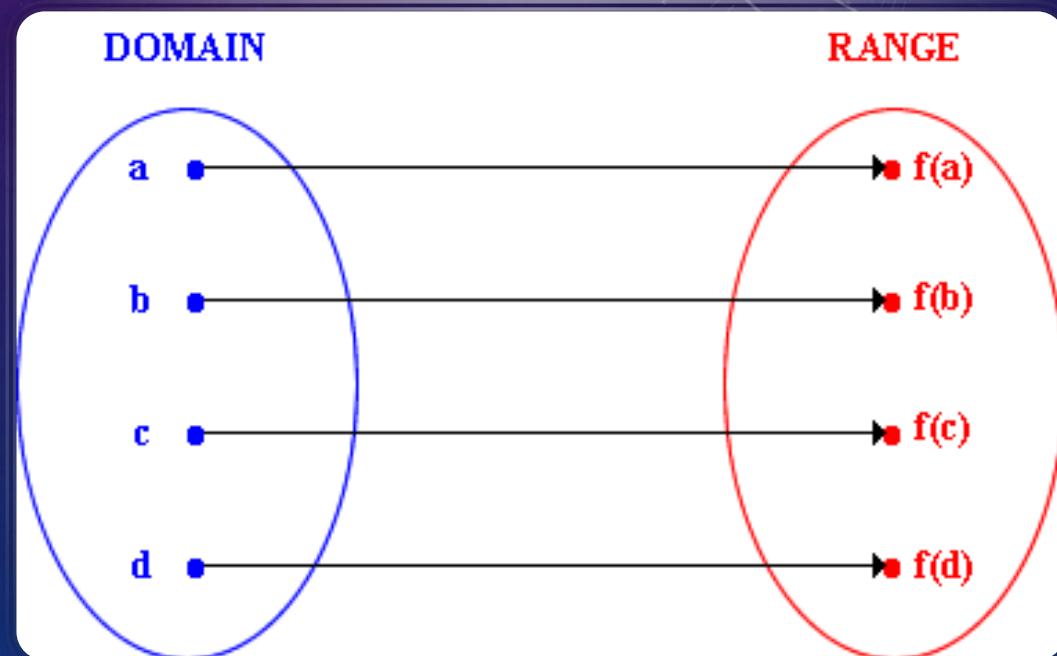
Please share the following:

- Your Name
- Area of Study
- Year
- One Thing You Enjoy



TODAY'S AGENDA

- Software in Robotics
- Standard Input & Output
- Variables and the 5 Basic Data Types
- User Defined Functions



SOFTWARE IN ROBOTICS

- Autonomy
- Remote Control



THERE'S A FIRST FOR EVERYTHING...

```
1 #include <iostream>
2 using namespace std;
3
4 int main(){
5     cout << "Hello World!" << endl;
6
7     return 0;
8 }
```

THERE'S A FIRST FOR EVERYTHING...

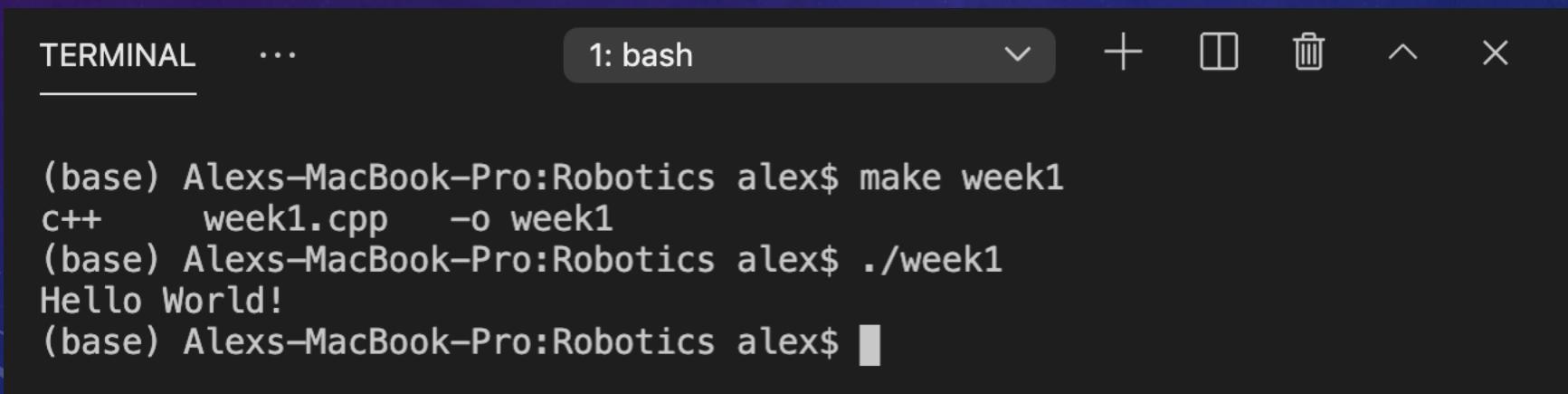
- `#include <iostream>`: imports the library for standard input and output to the console
- `using namespace std`: imports the standard namespace which has definitions for names 'cout' and 'endl'
- `int main()`: the function that gets called by default when the program is run
- `return 0`: signify end of program

STANDARD OUTPUT

- `cout`: keyword to pass data to the standard output
- `<< {data}`: adds the data following the `<<` operator to the output
- `endl`: ends the line by adding a newline to the output

COMPILING C++ CODE

- `make {filename}`: keyword shortcut to compile a single file quickly
- `./{filename}`: executes the actions of the compiled code file



```
TERMINAL ... 1: bash ▾ + ⌂ ⌍ ^ ×  
  
(base) Alexs-MacBook-Pro:Robotics alex$ make week1  
c++ week1.cpp -o week1  
(base) Alexs-MacBook-Pro:Robotics alex$ ./week1  
Hello World!  
(base) Alexs-MacBook-Pro:Robotics alex$ █
```

A screenshot of a Mac OS X terminal window titled "1: bash". The window shows a command-line session. The user first runs "make week1", which compiles a C++ program from a file named "week1.cpp" into an executable named "week1". Then, the user runs "./week1", which outputs "Hello World!" to the console. The terminal has a dark background with light-colored text and icons.

YOUR TURN...

Write a program to output the following to standard output:

One fish, two fish, red fish, blue fish

Black fish, blue fish, old fish, new fish

```
[Running] cd "/Users/alex/Desktop/Robotics/"  
One fish, two fish, red fish, blue fish  
Black fish, blue fish, old fish, new fish  
  
[Done] exited with code=0 in 0.4 seconds
```

STANDARD INPUT

```
C++ week1.cpp > ...
```

```
1 #include <iostream>
2 #include <string>
3 using namespace std;
4
5 int main(){
6     cout << "Type what your heart desires: ";
7     string data;
8     cin >> data;
9     cout << "You have inputted: " << data << endl;
10
11 }
12 }
```

```
(base) Alexs-MacBook-Pro:Robotics alex$ make week1
c++    week1.cpp   -o week1
(base) Alexs-MacBook-Pro:Robotics alex$ ./week1
Type what your heart desires: yeehaw
You have inputted: yeehaw
(base) Alexs-MacBook-Pro:Robotics alex$ □
```

STANDARD INPUT

- `cin`: keyword to pass data from the standard input
- `>> {variable}`: stores the data from standard input (up to the first space) into the string variable
- Note: the string library must be included to use string variables
- Note: the code must be run via terminal to input data
- Note: the VSCode 'run' button will not allow keyboard inputs

YOUR TURN...

Write a program to prompt the user to enter their name.

Then, print out "Hello" followed by their name 3 times.

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
(base) Alexs-MacBook-Pro:Robotics alex$ make week1
c++    week1.cpp   -o week1
(base) Alexs-MacBook-Pro:Robotics alex$ ./week1
Enter your name: Alex
Hello Alex!
Hello Alex!
Hello Alex!
(base) Alexs-MacBook-Pro:Robotics alex$ █
```

VARIABLES

- Definition: a specialized box to store a specific data type
- Property: variables have a fixed data type
- Syntax: {data type} {variable name} = {value};

```
string cookie = "chocolate";
bool oatmeal = false;
int count = 10;
double weight = 10.5;
char c = 'c';
```

BASIC DATA TYPES

- string: a collection of characters enclosed by double quotes
- char: a single ASCII character enclosed by single quotes
- int: a whole number
- double: a decimal number
- bool: either true or false

A LIL' EXPERIMENT

Between the 5 basic data types, determine what data type is returned when you use the following operators on all possible combinations of pairs of the data type:

+ (addition), - (subtraction), * (multiplication), / (division), % (remainder)

ZE' RESULTS

- `string/char`: can add any combination of strings and characters together
- `int`: an arithmetic operation between two integers return an integer (Ex: $5 / 4$ returns 1)
- `double`: an arithmetic operation with at least one double will return a double (Ex: $5 / 4.0$ returns 1.25)
- `bool`: `false = 0`; `true = 1`; booleans are converted to integers when used in arithmetic
- `modulo (%)`: returns the remainder of integer division of two numbers

FUNCTIONS

- **Definition:** a definable process that can be called multiple times
- **Function Call:** a call to the function to perform the process one time
- **Function Implementation:** the definition of the process of the function
- **Parameter:** an argument to the function (may have 0 or more)
- **Return type:** the data type that a function returns
- **void:** keyword to signify that a function does not return any data

FUNCTIONS

```
C++ week1.cpp > triple(double)
```

```
1 #include <iostream>
2 using namespace std;
3
4 double triple(double x){
5     return 3 * x;
6 }
7
8 void printHello(){
9     cout << "Hello" << endl;
10 }
11
12 int main(){
13     printHello();
14     cout << triple(10.5) << endl;
15     return 0;
16 }
```

```
(base) Alexs-MacBook-Pro:Robotics alex$ make week1
c++    week1.cpp   -o week1
(base) Alexs-MacBook-Pro:Robotics alex$ ./week1
Hello
31.5
(base) Alexs-MacBook-Pro:Robotics alex$ █
```

DATA TYPE CONVERSIONS

- `int(x)`: returns the integer value of x (where x is a number/boolean)
- `double(x)`: returns the double value of x (where x is a number/boolean)
- `bool(x)`: returns false if x equals 0 and true otherwise (where x is a number)
- `atoi(s)`: returns the integer value of s (where s is a string that is convertible into an integer)
- `atof(s)`: returns the double value of s (where s is a string that is convertible into a double)
- Note: atoi and atof are defined within the `<cstdlib>` library and it must be imported for the functions to be used

YOUR TURN...

1. Write a function that returns the integer cube of a number. (Ex: `cube(10.01) = 1003`)
2. Write a function that returns the fractional remainder of a division operation. (Ex: `fraction(5, 4) = 0.25`)
(Assume the denominator is non-zero.)
3. Write a function that prints a string concatenated with "IS A GOD". (Ex: `god("Zeus")` prints "Zeus IS A GOD")