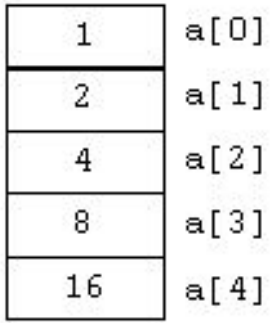
Discussion Section Week 4

More on Arrays...

Be sure to have an IDE open for any exercises

Review

- One of many data structures
- Static
- What's allowable?
- 0 indexed
- Access:
  - arrayName[elementNumber]



Warm-Up (5 Minutes to Try)

Write a program that does the following:

1) Creates an int array that will hold 10 elements
2) Using a for loop, populate the array with numbers 0-9
    a) Print out the content of your array

Solution

```cpp
#include <iostream>

int main(void)
{
    int arr[10];

    for(int i = 0; i < 10; ++i) arr[i] = i;

    for(auto j: arr) std::cout << " " << j;
    std::cout << std::endl;

    return 0;
}
```

```
TERMINAL  PROBLEMS  OUTPUT  DEBUG CONSOLE

derek@DESKTOP-3L8T6AU:/mnt/c/Users/Derek Jacobs/Desktop/CSC/TA/211$ g++ test.cpp && ./a.out
 0 1 2 3 4 5 6 7 8 9
```

What to expect today

- Argc, Argv

- Multi-layered Arrays

Argc, Argv

Argc, Argv

- A couple common ways to write “main”

```
int main(int argc, char *argv[])
```

- Arrays are just pointers
  - More on those later in the course

```
int main(int argc, char **argv)
```

What does Argc, Argv mean?

- Argument count
  - The number of arguments used when running your executable
- Argument values
  - An array of arguments passed into that executable

- Executable
- Arguments

Values of Argc, Argv

- Argc == 4

```
derek@DESKTOP-3L8T6AU:/mnt/c/Users/Derek Jacobs/Desktop/CSC/TA/211$ ./a.out argument0 argument1 argument2
```

- Argv: An array of character arrays (Keep this wording in mind)

0	1	2	3
./a.out	argument0	argument1	argument2

Exercise 1 (5 min)

Write a program that does the following:

1) Prints out the argument count of the program
2) Prints out the contents of all arguments passed to the executable
    a) Hint: You need to iterate through an array
        i) Think “while(array[element])”

- Execute your program with “./a.out argument0 argument1 argument2” to test your work

Solution

```cpp
#include <iostream>

int main(int argc, char **argv)
{
    std::cout << argc << std::endl;

    int i = 0;
    while(argv[i])
        std::cout << argv[i++] << std::endl;

    return 0;
}
```

TERMINAL PROBLEMS OUTPUT DEBUG CONSOLE

```
derek@DESKTOP-3L8T6AU:/mnt/c/Users/Derek Jacobs/Desktop/CSC/TA/211$ g++ test.cpp && ./a.out argument0 argument1 argument2
4
./a.out
argument0
argument1
argument2
derek@DESKTOP-3L8T6AU:/mnt/c/Users/Derek Jacobs/Desktop/CSC/TA/211$
```

Why are “argc” and “argv” useful?

- File parsing
- Options
  - ls -l -a
- Make them fit your own use cases
- Note: All arguments passed into argv are char* type. But, they can be changed!

Exercise 2 (10 Minutes)

Write a program that does the following:

1) Expects only 2 arguments in addition to the executable call
2) Converts those 2 arguments to integers
3) Prints out the result of raising the first number to the second power

```
derek@DESKTOP-3L8T6AU:/mnt/c/Users/Derek Jacobs/Desktop/CSC/TA/211$ ./a.out 12 2
144
```

Solution

```cpp
#include <iostream>
#include <cmath>

#define EXIT_FAIL 1

int main(int argc, char **argv)
{
    if(argc != 3)
    {
        std::cerr << "2 arguments were not passed. Exiting..." << std::endl;
        exit(EXIT_FAIL);
    }

    std::cout << pow(std::stoi(argv[1]), std::stoi(argv[2])) << std::endl;

    return 0;
}
```

Any Questions?

Multi-Dimensional Arrays

Up to This Point

[0, 1, 2, 3, 4]

[‘H’, ‘e’, ‘l’, ‘l’, ‘o’]

[1.234, 5.67, 8.90]

More than one dimension

int arr[5][4] = ….	
Index	Data
0	[1,2,3,4]
1	[5,6,7,8]
2	[9,10,11,12]
3	[13,14,15,16]
4	[17,18,19,20]

How it Works?

- First []
  - “Backbone”
- Second []
  - Size of each

```cpp
int main(int argc, char **argv)
{
    int arr[3][5] = {0};
    for (int i = 0; i < 3; ++i)
    {
        for (int j = 0; j < 5; ++j)
        {
            arr[i][j] = i + j;
            std::cout << arr[i][j] << " ";
        }
        std::cout << std::endl;
    }

    return 0;
}
```

```
TERMINAL  PROBLEMS  OUTPUT  DEBUG CONSOLE

derek@DESKTOP-3L8T6AU:/mnt/c/Users/Derek Jacobs/Desktop/CSC/TA/211$ g++ test.cpp && ./a.out
0 1 2 3 4
1 2 3 4 5
2 3 4 5 6
```

Back to argv…

- It’s really just a 2-D array
  - But we can’t use [][] when declaring

```
6  int main(int argc, char *argv[])
7  {
8      std::cout << argv[1][3] << std::endl;
9
10     return 0;
11 }
```

```
TERMINAL  PROBLEMS  OUTPUT  DEBUG CONSOLE
derek@DESKTOP-3L8T6AU:/mnt/c/Users/Derek Jacobs/Desktop/CSC/TA/211$ g++ test.cpp && ./a.out 01234567
3
```

Exercise 3 (10 Minutes)

Write a program that does the following:

1) Creates a 2-D Array with a length 3 “Backbone”and length 20 subarrays
2) Takes in 3 Names on standard input (cin)
    a) Stores those names in your array
3) Prints out “Hello ${name}” for each element in your array, where ${name} is each name that was input using a for loop

Solution

```cpp
int main(int argc, char *argv[])
{
    char arr[3][20];

    for(int i = 0; i < 3; ++i)
    {
        std::cin >> arr[i];
    }

    for(int j = 0; j < 3; ++j) std::cout << "Hello " << arr[j] << std::endl;

    return 0;
```

TERMINAL PROBLEMS OUTPUT DEBUG CONSOLE

```
derek@DESKTOP-3L8T6AU:/mnt/c/Users/Derek Jacobs/Desktop/CSC/TA/211$ g++ test.cpp && echo "Derek Mike David" | ./a.out
Hello Derek
Hello Mike
Hello David
```

Final Remarks/Points of Emphasis

- You can create an array of **any** type
  - An array cannot contain multiple types
- Arrays are static by default
- We can create dynamic arrays
- We can create arrays of any dimension
- Argv is just a 2-D array

Submission

For credit, please send an email to:

Doesn’t have to be code, could just be some sort of verification you were here.

See you next week!