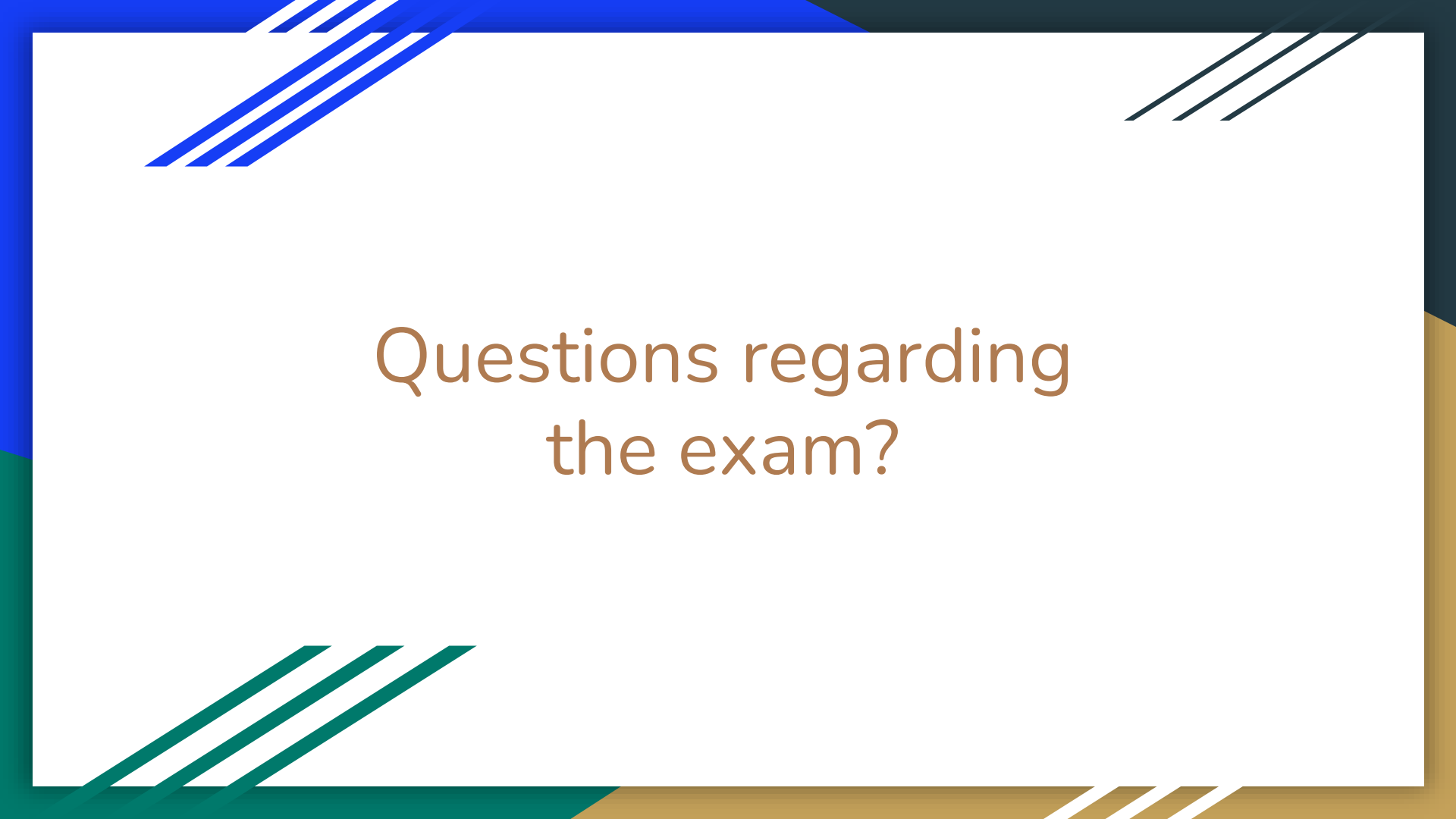


Discussion Section Week 6

More on Arrays...

Be sure to have an IDE open for any exercises



Questions regarding
the exam?

Review

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

1	<code>a[0]</code>
2	<code>a[1]</code>
4	<code>a[2]</code>
8	<code>a[3]</code>
16	<code>a[4]</code>

Solution

```
1  #include <iostream>
2
3  int main(void)
4  {
5      int arr[10];
6
7      for(int i = 0; i < 10; ++i) arr[i] = i;
8
9      for(auto j: arr) std::cout << " " << j;
10     std::cout << std::endl;
11
12     return 0;
13 }
```

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

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

- 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

```
1  #include <iostream>
2
3  int main(int argc, char **argv)
4  {
5      std::cout << argc << std::endl;
6
7      int i = 0;
8      while(argv[i])
9          std::cout << argv[i++] << std::endl;
10
11     return 0;
12 }
```

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

```
1  ✓ #include <iostream>
2    #include <cmath>
3
4  |  #define EXIT_FAIL 1
5
6  ✓ int main(int argc, char **argv)
7  |  {
8  |  |  if(argc != 3)
9  |  |  |  {
10 |  |  |  |  std::cerr << "2 arguments were not passed. Exiting..." << std::endl;
11 |  |  |  |  exit(EXIT_FAIL);
12 |  |  |  }
13 |  |
14 |  |  std::cout << pow(std::stoi(argv[1]), std::stoi(argv[2])) << std::endl;
15 |  |
16 |  |  return 0;
17 |  }
```

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

```
6 | int main(int argc, char **argv)
7 | {
8 |     int arr[3][5] = {0};
9 |     for (int i = 0; i < 3; ++i)
10 |     {
11 |         for (int j = 0; j < 5; ++j)
12 |         {
13 |             arr[i][j] = i + j;
14 |             std::cout << arr[i][j] << " ";
15 |         }
16 |         std::cout << std::endl;
17 |     }
18 |
19 |     return 0;
20 | }
```

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

```
5 |
6 | int main(int argc, char *argv[])
7 | {
8 |     char arr[3][20];
9 |
10 |    for(int i = 0; i < 3; ++i)
11 |    {
12 |        std::cin >> arr[i];
13 |    }
14 |
15 |    for(int j = 0; j < 3; ++j) std::cout << "Hello " << arr[j] << std::endl;
16 |
17 |    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