

Activity 5

Comp 11 - Summer Session — Strings and Arrays

With a partner(or two), discuss the following code sample and answer the questions below. The instructor and teaching assistants will let you discuss and then be around to answer questions. ¹

5.1 Description

Arrays are an abstraction for storing pieces of contiguous data.

As we saw in lecture, `std::string` is actually an array of characters.

```
1 #include <iostream>
2
3 int main() {
4
5     int myArray[10];
6     myArray[5] = 2;
7
8     std::cout << myArray[5] << "\n";
9
10    return 0;
11 }
```

Listing 5.1: Arrays Template

5.2 Questions

1. Am I allowed to store different types of data within an array? That is, can I create one array and store, `int`, `double`, `std::string`, and other types within this same array?

¹Activities do not need to be returned to instructors, they are for your benefit.

2. Write a program that generates 100 random numbers in a loop and stores them in an integer array called *lotteryNumbers*.
3. What will happen in the program below?

```

1 #include <iostream>
2
3 int main(){
4
5     int myArray[10];
6     myArray[15] = 2;
7
8     std::cout << myArray[15] << "\n";
9
10    return 0;
11 }

```

Listing 5.2: Array Index Error

4. Write a program that generates 100 random numbers in a loop and stores them in an integer array called *lotteryNumbers*.
5. What is one purpose of the '0' character?
6. Is a *char[1]* different from a *char*?
7. Why might we prefer to use the *string* type over the character array to store strings?

```

1 // Run to find the solution of char[1] vs char.
2 // Note that decltype(a) is the same thing as saying
3 // declare a new variable as the same
4 // type of variable a which is a char and vice versa.
5
6 #include <iostream>
7 #include <string>
8
9 // A way to test if types are the same
10 // Create new types and try to initialize them
11 // to the previously created types.
12 int main(){
13
14     char a;
15     char b[1];
16
17     // Will these lines throw an error?
18     decltype(b) c = a;
19     decltype(a) d = b;
20     return 0;
21 }

```

Listing 5.3: decltype example