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.

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.

```
#include <iostream>

int main(){
    int myArray[10];
    myArray[5] = 2;
    std::cout << myArray[5] << "\n";
    return 0;
}
```

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?

```cpp
#include <iostream>

int main()
{
  int myArray[10];
  myArray[15] = 2;
  std::cout << myArray[15] << \n;
  return 0;
}
```

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?

```cpp
// Run to find the solution of char[1] vs char.
// Note that decltype(a) is the same thing as saying
// declare a new variable as the same
// type of variable a which is a char and vice versa.

#include <iostream>
#include <string>

// A way to test if types are the same
// Create new types and try to initialize them
to the previously created types.
int main()
{
  char a;
  char b[1];
  // Will these lines throw an error?
decatype(b) c = a;
decatype(a) d = b;
  return 0;
}
```

Listing 5.3: decltype example