Welcome Back

What you learned in CSCI 261 (or equivalent):
- Variables
- Types
- Arrays
- Expressions
- Conditionals
- Loops
- Functions
- Classes & Objects
- Pointers

You remember all of this, right? 😊

What This Semester Is About

Learning objectives:
- Know basic data structures
  - What they are and how (and when) to use them
  - Know how to design and program them
- Understand and use (most of) C++ features
- Understand and use object-oriented methods
- Understand performance of algorithms

Hello, Let’s Review

Here’s a simple C++ program:

```cpp
#include <iostream>
using namespace std;

int main()
{
    string hello = "Hello, world!";
    cout << hello << endl;
    return 0;
}
```

Hello, Let’s Review

```cpp
#include <iostream>
using namespace std;

int main()
{
    string hello = "Hello, world!";
    cout << hello << endl;
    return 0;
}
```
Variables

Example:
int x;

Use in expressions:
\[ x + 10 \]

Set via assignment operator:
\[ x = 4; \]

Declare and initialize:
\[
\begin{align*}
\text{int } & \quad x; \\
& \quad x = 42;
\end{align*}
\]

Or
\[
\text{int } x = 42;
\]

Types

- Basic types
  - Integer types:
    - int: 42
    - char: 'k'
  - Floating point types:
    - double: 3.14159, 4.5e3
  - Boolean type:
    - bool: true, false
- Pointers
- Arrays
- Class/struct types

Expressions & Operators

Anything with a value is an expression:
- Variables
- Indexed array variables
- Literals
- Arithmetic/logical expressions:
  - 4 + 7 / 3.0
  - (x * 2 + 1) % y
  - a || b && c

Which operators act first? Use parentheses or know precedence rules.

Loops

What if we want to print "Hello, world!" three times?

```
int i;
for (i = 1; i <= 3; i++) {
    cout << i << " Hello, world!" << endl;
}
```

Output:
1 Hello, world!
2 Hello, world!
3 Hello, world!

Another Loop

```
int i = 3;
while (i > 0) {
    cout << i << " Hello, world!" << endl;
    i--;
}
```

Output:
3 Hello, world!
2 Hello, world!
1 Hello, world!

Conditionals

```
if (true-false-expression) {
    true-block
} else {
    false-block
}
```

Also should know use of:
break
continue
Hello, if?

Let's modify Hello to respond to an input:

```cpp
char answer;
cout << "Say (H)ello or (G)oodbye?" << endl;
cin >> answer;
if (answer == 'H') {
    cout << "Hello, world!" << endl;
} else {
    cout << "Goodbye, world!" << endl;
}
```

What if the user enters “h” instead of “H”?

Arrays

```cpp
int numbers[3];
numbers[1] = 14;
numbers[2] = -3;
numbers[3] = 7093;
```

Oops? What’s wrong here?

Loops on Arrays

```cpp
for (int i = 0; i < 3; i++) {
    cout << numbers[i] << endl;
}
for (int i = 2; i >= 0; i--) {
    cout << numbers[i] << endl;
}
```

Note array initialization – only when array is declared!

Arrays

```cpp
int numbers[3];
numbers[0] = 14;
numbers[1] = -3;
numbers[2] = 7093;
```

Let’s print out the numbers in the array.
What about in reverse order?

Something New

C++ 11 added a new type of for loop:

```cpp
int numbers[] = {14, -3, 7093};
for (int x: numbers) {
    cout << x << endl;
}
```

This denotes that x is a variable of type int which will take on each value in numbers in turn.

As we’ll see, this ranged-based for loop can be used with many container data structures as well.

Functions

We've seen one function:

```cpp
int main() { ... }
```

Here’s another:

```cpp
int print_it(string msg) {
    cout << msg << endl;
    return msg.length();
}
```

A member function of the string class.
Hello Functions!

A silly program.
#include <iostream>
#include <string>
#include <cmath>
using namespace std;

int print_it(string msg) {
    cout << msg << endl;
    return msg.length();
}

int main() {
    int n;
    double nroot;
    n = print_it("Hello, world!");
    nroot = sqrt(n);
    cout << "The square root of the number of characters printed is";
    cout << nroot << endl;
    return 0;
}

About Strings

In C/C++, the literal "Hello" is called a string. It is of type char[] (a char array).

Confusingly, C++ defines a new type, string.

A string is mostly interchangeable with a string. But, you can do more with string objects:

#include <string>

- string foo = "Hello"; // note assignment of string to string
- string bar = "World"; // actually implicit constructor call
- string hello = foo + ", " + bar + "!";
- if (foo == bar) {...} // test for equality works with string

More About Strings

Know/learn the string interface!
- See Help page of course website for C++ documentation websites
- Some functions you should know at minimum:
  - length operator[]
  - size operator+
  - find operator++
  - substr relational operators

Pointers

...will wait for another day.