

# Introduction to Software Development – CS 6010

## Lecture 4 – Strings

Master of Software Development (MSD) Program

Varun Shankar

Fall 2022

# Lecture 4 – Strings

- Topics
  - Strings
  - Code Review – Vending machine
  - Lab – Strings & Loops
  - Homework – Roman Numerals

# Quick Review on Variables

- Variables store data
  - ... and allow the programmer to easily access it.
- What information is necessary to create a variable?
  - Type, Name, Value
  - Syntax?
    - `float distance = 929000000.0; // Miles`

# Quick Review on Variables, cont

- What are the 5 types (datatypes) of variables we have discussed?
  - Basic Types
    - Numbers
      - integers, floats, doubles, signed, unsigned
    - Characters
    - Booleans

# Quick Review on Variables, cont

- What are the 5 types of variables we have discussed?
  - Advanced Types
    - Arrays
      - Ordered list of the *same* type of data
        - eg: [ 'H', 'e', 'l', 'l', 'o' ] or [ 1.4, 3.2, 7.0, 0.1, -2.4 ]
    - Structures/Objects/Classes
      - Containers of multiple different pieces of data that represent one thing.

# Common If Statement Mistakes

- `if( x == 5 || 6 || 7 )` // WRONG
  - `if( x == 5 || x == 6 || x == 7 )` // RIGHT
- `if( !count == 100 )` // WRONG
  - `if( count != 100 )` // RIGHT
- `if( !( count == 100 ) )` // works, but is not ideal. Use above version!

# Proper Indentation is Important!

```
int main() {  
    int value = 1;  
    for( x = 1; x < 10; x++ ){  
        if( x > 5 ){  
            cout << "2nd half\n";  
        }  
        else {  
            if( x < 3 ) {  
                cout << "small num\n";  
            }  
        }  
    } // <- What { does this match?  
}
```

Indent!

Correctly.

Please.

Or eventually lose points.

# Proper Indentation is Important!

```
int main() {  
    int value = 1;  
    for( x = 1; x < 10; x++ ) {  
        if( x > 5 ) {  
            cout << "2nd half\n";  
        }  
        else {  
            if( x < 3 ) {  
                cout << "small num\n";  
            }  
        }  
    } // <- Matches the { of the for loop  
}
```



# Strings are Objects

- `char answer = 'y';`
- What if we wanted *answer* to be “yes”?
- What do we call a group of more than one character?
  - A string (or an array of characters)
- The ability to create a string object is provided by the `<string>` library.
  - To use it, simple `#include <string>` at the top of your `.cpp` file
- A *string* is a compound data type, or more specifically an *object*.
- In C++, an *object* can contain multiple (sub) pieces of information, but also provides functionality on that data.
  - This functionality is provided through *methods* on the object.
- Generically, to use (invoke) a method of an object, you would say:  
`object.method( parameters )`
  - This will do something to those parameters, and possibly return an output

# The *string* object

- `#include <string>` // use the string library
- `using namespace std;` // or use `std::string`
- `string name = "Varun";`
- Basic operations on a string:
  - `length()` – Returns the number of characters in the string (a method).
  - `stringname[somePosition]` – Returns the character at that position in the string (not a method). **Positions in C++ start at 0**
  - `==, !=, <=, >=` - Compares two strings with each other.
  - `+` - Concatenates two strings together
  - `find( substr [, pos] )` – finds the location of *substring* (starting at *position*) (a method)
  - `substr( pos, length )` – returns a new string which starts at *position* and goes *length* characters (another method)
  - many others...

# Strings and Integers

- How to convert a string (eg: "10") to a number (10).
  - `std::stoi( myString )` // stoi == string to integer
  - `std::stof( myString )` // string to float
- Note: how to convert "ten" to a number (10).
  - Can't be done (well, not directly).
- How to convert a number to a string?
  - `int myNumber = 10`
  - `std::to_string( myNumber )` // Returns "10"

# String Examples

- `// Ask the user to enter a string, then display the following:`
- `// Number of letters in the string`
- `// The first and last letter of the string. Do this in multiple ways`
- `// All but the first and last letters of the string`
- `// And “ing” to the end of the string.`
- `// Display the original string concatenated with some other string`
- `// Determine if the two strings are the same, and if not, which one is alphabetically first?`
- `// Replace the first letter of original string with the first letter of the created string.`
- `// Print out each letter in the string with spaces separating them.`
- `// Convert a string to a number and back`

# Today's Assignment(s)

- Code Reviews
- Lab – Strings & Loops
- Homework – Roman Numerals