Lecture 01 - Getting Started

CS 1342 - SMU

Mark Fontenot, PhD

May 21, 2019

Agenda

1. Introduction to C++
2. C++ Compared to Java
3. Basic console input and output
4. Control Structures
5. Writing your first program

Introduction to C++

Comparing C++ to Java

- C++ is compiled to machine code.
    - Compiling and Linking results in a *machine executable*.
- Java is compiled to Java Byte Code.
    - Byte code is then interpreted through the JVM
- C++ allows much more direct access to memory than Java.

Comparing C++ to Java (more)

- The basic unit of code is the **function**
- Major control structures are the same:
  - Sequence
  - Decision
  - Repitition
- Roughly similar primitive data types
  - int, char, float, double, long, … and some others

The Simplest C++ Program

```cpp
int main()
{

    return 0;
}
```

- Contains just one method/function
- **main** serves the same purpose as in Java… it is the entry point for the project.
- Returns an integer to whatever called it (usually the IDE or the shell)
- Notice: it isn't in a class like it would be in Java.

Basic Console Output

- **cout** is an object that is responsible for *output* to the *console*.
- In C++, there are operators that help with i/o.
    - In Java, you had to use `System.out.println(...)` and friends

```
int main()
{
    cout << "Hello World";
    return 0;
}
```

- << is called the **stream insertion operator**
- Anything in double quotes is considered a **string literal**.

Including Header Files

- `#include <iostream>`
- `include`s are similar to Java's import
- gives you access to things implemented in the C++ standard library

```cpp
#include <iostream>
int main ()
{
   cout << "Hello " << "World!";
   return 0;
}
```

Output Stream Manipulators

- allow you to modify the output

```
cout << "Hello " << endl << "World!" << endl;
cout << "CS 1342 is" << endl;
cout << "THE BEST!" << endl;
```

- others include `setw`, `setprecision`, etc.

Primitive Variables

- very similar to Java

```
int x;
x = 10;
int a, b, c;
a = b = c = 25;
double pi = 3.1415; //float is OK too
```

Basic Console Input

- `cin` == console input
- >> == stream extraction operator
- allows you to read from the keyboard

```
int x;
cout << "Enter a number: ";
cin >> x;
int a, b, c;
cout << "Enter 3 numbers: ";
cin >> a >> b >> c; //don't separate by commas... use >>
```

- some rules and nuance to be aware of

Console Input Rules

- If reading into an integer var:
    - skip preceding whitespace and read digits until first non-digit character
- If reading into a FP var:
    - skip preceding whitespace and read digits and possibly a decimal point folowed by more digits. Stop at first non-digit or non-decimal point char.
- If reading into a single char variable:
    - skip preceding whitespace and read a single character from the input.

Checkpoint

```
int a;
char b;
float c;
cin >> a >> b >> c;
cout << a << endl << b << endl << c;
```

- What would be printed if the user entered:
  - `123b14.2` ?
  - `12.345.999` ?

Control Structures

Questioning Yourself (Conditionals)

- most common is the `if` statement

```
int x = 10, y = 20; //notice the double var
                    //declaration and init.
if (x < y)
{
    cout << "X is less than Y" << endl;
}
```

- `if...else`

```
if (...)
{
}
else
{
}
```

Conditionals Continued

- `if...else if...else`

```
if (...)
{
}
else if (...) //can be multiple of these
{
}
else
{
}
```

Conditional Tests

```
if (... && ...)

if (... || ...)
```

Repeating Yourself (Loops)

```
for(int x = 0; x < 10; x++)
{
    //do some cool stuff here
}

while (x < 10)
{
    //do some cool stuff here
    x++; //probably
}
```