**CS 368-1: Lecture 1**
**Learning a New Programming Language : C++ for Java Programmers**

Take a copy of the syllabus, and be sure to sign the attendance sheet

**A. Preview**

| **Previously….** | **Today** | **Next Time** |
|---|---|---|
| • CS 302<br>• CS 367 (recommended)<br>• programming confidence | • handout – course info<br>• course intro/logistics<br>• historical overview<br>• simpleProg.cpp<br>• bool is equiv to 0 or 1<br>• constants similar to Java<br>• enumerated types<br>• chained assignments | • structures<br>• arrays<br>• vectors<br>• cardExample.cpp<br>• editing with vim<br>• compile and run with g++ |

**B. Course Information**
**Waiting List information: This course is full. If you cannot wait to take this course next semester, please have your advisor email me to explain the details.**

Course content will come from lecture, textbook, and course website http://pages.cs.wisc.edu/~cs368-1/

To receive 1 credit (a grade of CR)
• attend at least 12 of the 15 lectures (must sign attendance sheet at each lecture)
• achieve a cumulative score of 70% on the assignments
• for each missed lecture past 3 add 5% on to the cumulative score requirement
miss 0-3, need 70% miss 4, need 75% miss 5, need 80% etc…..

Assignments
• 5 programs, 1 written homework
• p1, p2, p3, p4 = 20% each
• p5, hw = 10% each
• collaboration policies: homework must be done individually, pair programming is allowed on programs

You are encouraged to bring a laptop to class.

**C. Historical Overview**

C(1972) : Designed by Dennis Ritchie, Bell Labs(Microsoft, Google of the time), early 1970s.
-was working converting Unix from Assembly Language to a more portable language

Came from B, which came from BCPL, (created by Martin Richards, Cambridge )
Meant to be a portable, English-like language with access to low level operations.
Machine Language ------Assembly Language ----------C ---- High Level Languages

Provided powerful mechanisms that could be manipulated and abused
C "treats you like a consenting adult".... do whatever you want, "Dangerous if not used properly"
as opposed to Pascal (created for education), Ada (created by Dept of Defense)

A program is a series of functions, with a main method
int **main (int argc, char **argv)** **argc is argument count, argv is argument vector (list)**

C++ (1980): Bjarne Stroustrup ……. C++ (increment C)
Originally called "C with Classes"
- upward compatible with C - as fast as C (little compiler checking to avoid misuse)
One of the lowest of the High Level Languages
Machine -------Assembly ---------- C --- C++ ----Other High Level (Java, Python, etc)

C++ was standardized in 1998 and again in 2011 (C++98, C++11)
Features of the C++ language are put in there to help programmers solve real problems
The language does not try to force programmers to do things a certain way

Java (1995) : Designed by Sun Microsystems Designed to be a cleaned up version of C++
Simplify choices, give the programmer only one way to do a certain thing
Less syntax (less to learn)
Took out some features of C++ that programmers regularly abused
Has a Compiler (system specific) and a Virtual Machine(universal)
The Compiler turns Java into Java ByteCode, which is interpreted at runtime by the Virtual Machine.
Java ByteCode can be run on any platform.

The Compiler detects more errors, Virtual Machine throws exceptions
programming errors (bugs) are avoided
system security is enhances (others can't mod code to access privileged memory)

Fewer Programmer choices, more strict rules of what you can and cannot do

Java assumes that compilers are fast, computers have lots of memory,

**D. Different philosophies & goals:**

| Java (came later) | C++ (came first) |
|---|---|
| tries to force programmers to write good code<br>doesn't worry about speed | allow programmers who know what they are doing to quickly write code that runs efficiently |
| Java has built in memory management "garbage collection"..... | programmers manage memory themselves |
| ArrayIndexOutOfBoundsException | Allows programmers to access any memory through an array variable (or pointer) |
| Every variable must be assigned a value before it is used, some have pre-assigned default values.<br>( 0 or null) | A variable that is not assigned a value could contain anything. |
| API is huge (lots of predefined classes) | No real API, but uses C libraries |

**E. Simple C++ program**

```
// PREPROCESSOR DIRECTIVES
#include <iostream>                          // preprocessor inserts the code in this file
                                             //  < > for system files, “ “ for user-defined

using namespace std;                         // import java.lang.*;


 // MAIN FUNCTION                            // public static void main (String [] args)
int main( ) {

 cout << "Enter a number and a letter: "<< flush;   //”what follows is inserted into cout”
                                                    // cout means character out

 int a;
 char b;
 cin >> a >> b;
 cout << endl;

 cout << "The number you entered was: " << a << endl;
 cout << "The letter you entered was: " << b << endl;

 return 0;                                   // main returns an int
                                             // common practice to return 0 if OK result
                                             // return 1 if there was an error
}
```

Let’s try to run this program in a browser compiler.

## F. Program Notes

**Include:**
Class and function libraries are typically compiled separately and linked with client code to create an executable program. The client code must import the class and/or function declarations so the compiler can determine if classes and functions are used correctly. Declarations are typically imported using the preprocessor directive #include which literally copies the specified file into the program being compiled.

The C++ standard specifies that standard include files are specified without a .h suffix, (eg), <iostream> and <string>. For the most part, these header files import declarations that are in the *std* namespace (see Sec. A.2.3). Using a file with the .h suffix, for example <iostream.h>, imports the file in the global namespace. This means that the directives below on the left are the same as that on the right.

```
#include <iostream>              #include <iostream.h>
 using namespace std;
```

Although most systems support both <iostream> and <iostream.h>, the namespace- version is what's called for in the C++ standard. In addition, some files do not have equivalents with a .h suffix—the primary example is <string>.

**Namespaces:**

Large programs may use classes and functions created by hundreds of software developers. In large programs it is likely that two classes or functions with the same name will be created, causing a conflict since names must be unique within a program. The namespace mechanism permits functions and classes that logically related to be grouped together. Just as member functions are specified by qualifying the function name with the class name, as in Dice::Roll or string::substr, functions and classes that are part of a namespace must specify the namespace. Examples are shown in Prog. A.1 for a user-defined namespace *Math,* and the standard namespace *std.* Note that using namespace std is not part of the program.

```
#include <iostream>          #include <iostream.h>
 using namespace std;
```

Although most systems support both <iostream> and <iostream.h>, the namespace- version is what's called for in the C++ standard. In addition, some files do not have equivalents with a .h suffix—the primary example is <string>.

- (from Computer Science Tapestry, Owen Astrachan, 1999).

**G. Constants / Booleans / Enumerations**

1. (Symbolic) Constants are declared differently in C++, but act in the same was as they do in Java.

```
const int MAXSIXE = 100;
const double MINIMUM_WAGE = 7.5
const char BEE = 'b';
```

2. Variables of type bool:

| | |
|---|---|
| • C++ has a boolean data type, named bool, C does not<br><br>• 0 or 1 can be used where a boolean value is expected | `#include <iostream>`<br>`using namespace std;`<br><br>`int main() {`<br>`  bool livesOnCampus = false;`<br><br>`  if (livesOnCampus == 1){`<br>`  cout << "one" << endl;`<br>`  }`<br>`  else{`<br>`  cout << "two" << endl;`<br>`  }`<br>`  return 0;`<br>`}` |

3. Enumerated types:

| | |
|---|---|
| Enumerations<br>General form :<br>enum type-name { list-of-values };<br><br>allows you to create all the legal values for a new type<br>enum CoinValue {heads, tails};<br><br>chained assignment operators<br><br>Another Example :<br>enum Suits { clubs, diamonds, hearts, spades }; | `#include <iostream>`<br>`using namespace std;`<br><br>`int main() {`<br>`  enum direction {up, down, left, right};`<br>`  direction othermove;`<br>`  direction nextMove = othermove = up;`<br>`  if (nextMove == 0)`<br>`  cout << "one" << endl;`<br>`  else`<br>`  cout << "two" << endl;`<br>`  cout << "next move is " << nextMove << endl;`<br>`  // nextMove++;  compiler  error !`<br><br>`  if (nextMove == down)`<br>`  cout << "three" << endl;`<br>`  else`<br>`  cout << "four" << endl;`<br><br>`  return 0;`<br>`}` |

| | |
|---|---|

**On your own:**

1. Try running the sample program in part E. with the following user inputs:
87b
-12h
9 b
43.2a
9 b abc
y76

• What do you think will happen?
• What actually happens?

2. Review how to use the CS lab accounts that you used in CS302. If you forgot, plan to attend a tutorial session given this and/or next week.

3. Read Chapter 0 and Chapter 1 in the Text.

4. If you never learned about Hexadecimal Numbers, learn how to count in Hex.

5. Make your own enumerated types. Try them out in a browser-based compiler.