COMSC-165 Lecture Topic 0
Course Orientation

Introduction
Advanced Programming, with C/C++
This is your second programming class
You will learn the C++ programming language
  prereq: Comsc-110
We will...
  1. Review the basics
  2. Learn and apply pointers
  3. Learn linked lists
  4. Apply recursion
About the instructor
  Ph.D. Purdue, Mech.Engr.
  programmer since 1969
class website http://cs.dvc.edu
this is NOT an online class
  attend lecture
  attend lab

Course Goals
Prepare for COMSC-200, 210, and 260
Prepare for xfer to UC/CSU COMSC program
Grading policy
  redos, late work, sequence
12 hours per week *(3 lec + 3 lab + 6)*
Learning process: read, quiz, lecture, lab, project
Syllabus and Course Outline (with schedule)
  academic honesty policy

Lecture Period
sign the sign-in sheet (look for "see me")
no electronics during lecture (except photos)
lectures recorded (MP3), files posted
command-line compiling used in lecture
  "lowest common denomiator"
  vendor-, system-, and compiler-independent

Lab Period
Microsoft Visual C++
  on PCs in ATC, ET, and L bldgs (*USB drive recommended*)
  Express 2013 for Web version available for free download
  IDE mode: static library, no precompiled headers
g++ Mac download ; PC MinGW ; PC Cygnus (see instructor)
Upload **.cpp** source files to the COMSC server
  workInProgress, Google Drive, OneDrive, Dropbox
free resources
  DreamSpark accounts
  onedrive.live.com

Textbooks
"C++: How To Program: Late Objects Version" by Deitel
  other editions may be suitable
  click HERE for TOC
we cover ch.1-8 (and part of 9)
  Comsc-200 covers ch.9 and beyond
other required reading:
  the lecture notes
  www links in the lecture notes

Lab Assignments
Highly structured assignments, 1 per week
Strict about coding practices
  debugging techniques
  alignment and indenting
about "redos"
  no grade until work is complete and correct
  no consideration of next lab until current is complete
  2 point penalty for not following instructions
  no point penalty for "learning opportunities"

just because your program works for you,
  it does *not* mean that
  you did it right!
programming conventions and lab 0

Review -- what you should know so far...
Computers are *dumb*
What is a "language"
  CPU -- voltage I/O
  machine language
  assembly language
  C language
Algorithms -- recipe, instructions, flow charts
Progr. Lang. is *simplified*, *strict* English
Coding: translating algorithm to code
Where C and C++ fit into the world view

Changes From Comsc-110
teaching C++11 standard
  Visual C++ 2012, MinGW 4.8, XCode 4.6, and above
using C `char`-based arrays used for strings (not C++ `string`s)

do *not* use `using namespace std;`
  use (e.g.) `using std::cout;`
distinguish between C libraries and C++ libraries
  C library names start with "c"
  C libraries do *not* use `using std::`'s

do *not* read `int`s or `double`s directly from console

```
// do NOT use this anymore
int x;
double y;
cin >> x >> y;

// use THIS instead: "string buffer method"
int x;
double y;
char buf[100];
cin >> buf; x = atoi(buf); // requires #include <cstdlib>
cin >> buf; y = atof(buf); // requires #include <cstdlib>
```

accommodate "round-off error"
  ...in calcs using `double`s

```
// do NOT use this anymore (for example)
double y = ...; // result of a calculation
if (y == 100)
```

optional: Burns "Intro To Programming Using C++ and Java"
basic C++ reference

Class Website Tour
Internet URL http://cs.dvc.edu/
student accounts: IDs and passwords
lab assignments and lecture topic notes
online quizzes -- strict time periods
contact instructor via form
some replies to google group

```
// use THIS instead (for example)
if (99.999 < y && y < 100.001)
```

g++ command line compiling with C++11

```
g++ -std=c++11 hello.cpp
```

XCode On OSX Mavericks (and Yosemite)
go to a Terminal session, and type the command: `g++`
If not installed, you'll see a prompt saying so
and inviting you to download and install it
accept the invitation, and in a few minutes...
...you'll have g++ (fully C++11 compatible, too).

Three program design consideration for COMSC 165

1. Pausing a the end of a program *(optional)*:

```
  cout << "Press ENTER to continue..." << endl;
  cin.get();
} // main -- return 0; not required
```

2. Programmer identification in *all* CPP files

```
// Lab LAB NUMBER HERE, LAB TITLE HERE
// Programmer: YOUR NAME HERE
// Editor(s) used: XP Notepad
// Compiler(s) used: VC++ 2010 Express
```

3. Programmer identification in program output

```
int main()
{
  // identifying output statements
  cout << "Lab 1, Problem Solving With C++, Part b\n";
  cout << "Programmer: Joe Student\n";
  cout << "Editor(s) used: XP Notepad\n";
  cout << "Compiler(s) used: VC++ 2010 Express\n";
  cout << "File: " << __FILE__ << endl;
  cout << "Complied: " << __DATE__ << " at " <<  __TIME__ << endl << endl;
  ...
}
```