

# COSC 1030

## Introduction

(adapted from Dr. Kim Buckner)

Nels Frazier

University of Wyoming

Aug 28, 2023

# Course Labs

- If you want to work at home outside lab sessions you will need to
  - install Visual Studio Community edition (only on Windows machines) or
  - use the IT Remote Labs.
  - Use an 'open' IT lab like in the Library or the IT building. See [microlab.uwyo.edu/labs/](https://microlab.uwyo.edu/labs/) for a listing of labs.
  - Use EN4072, the Computer Science student lab.

## (more ...)

- There will be a TA for the labs, Paul Gyreyiri.
- You will have to do most of the work on your own
- You CAN discuss labs/programming assignments with me, with Paul, or with your classmates but DO YOU OWN WORK. Anything else is academic dishonesty.

- Course concepts
  - Learn C++.
  - Use an IDE.
  - Use command prompt.
  - Design, write, compile, execute, debug, and **test** programs.

- Why do we concentrate on it at UW?
- History.
  - Announced in 1979.
  - Creator is Barne Stroustrup.
  - Developed as “C with Classes”.
  - 1983 – renamed C++.
- The STL.
  - This is the Standard Template Library.
  - Has support for algorithms, containers, functions, and iterators.

# (more ...)

- C++ Resources.
  - MSDN (online)
  - Course text
  - [cplusplus.com](http://cplusplus.com)
  - [cppreference.com](http://cppreference.com)
- The UW library.
  - search on C++
  - 600+ entries, many ebooks that you can access
  - might want to look at “C++ Primer Plus”
  - or “C++ Primer”

(more ...)

- Instructor and TA
  - we have office hours,
  - come in if my door is open, and
  - ask questions early.

# Tutoring

- STEP is providing COSC 1030 tutoring at the Library. See the syllabus for more detail.



# Course management

- Temporarily, resources will be linked to [www.cs.uwyo.edu/~nfrazier1/cosc1030/](http://www.cs.uwyo.edu/~nfrazier1/cosc1030/)
- We will be using WyoCourses as our LMS. (As soon as I get requisite access!)
- This is reasonably easy.
- You need to login and set your personal preferences.
- Go to: [wyocourses.uwyo.edu](http://wyocourses.uwyo.edu)
- Also available from WyoWeb.

# Syllabus

- The most current version will be available on the WyoCourses site. (Soon!)
- Please read the syllabus, it contains information on course management, grading, expectations, and some recommendations.

# What will happen

- We will discuss C++ programming concepts during the lectures.
- I will try to make sure that there is plenty of time for questions and answers.
- I expect you to put in some effort.

# (more ...)

- That effort should include asking questions, programming on your own, trying things, reading.
- I strongly urge you to attend class, material discussed in class may not be in the text or on the lecture notes.

# Quizzes and Exams

- Starting next week, we will have a brief quiz every Tuesday (not following a Thursday exam) at the end of class.
- There will be three one-hour exams. The schedule is on the WyoCourses. It may change but I doubt it.
- There is a final exam, it is comprehensive (the entire course).

(more ...)

- The Quizzes are 5% of the total grade, Exams 30%, Final Exam 10%.
- That means 45% of the grade is exams of some sort.

# Labs

- Weekly labs, attendance (at the start of lab) is mandatory or no credit for that week's lab.
- Labs count 25% of the final grade.
- There is an in-lab assignment that must be turned in within 24 hours to receive lab credit.

## (more ...)

- There is a TA for each lab.
  - There to help.
  - May give a brief intro at the beginning of lab.
  - NOT there to give you the code.
  - If you do not ask questions, the TA cannot help
  - TA will leave when there are no more students, or the end of the session whichever comes first.
- The TA is NOT there for you to shower abuse upon.



# Programming Assignments

- These are really homework.
- Programming assignments are made weekly, posted on Tuesday.
- Due Friday by Midnight.
- They can be late, at a **20%** per 24 hours penalty. Max of 2 days late.
- Programming Assignments are 30% of the final grade.

# Grading

	Exams 30%	Final 10%	Quiz 5%	Labs 25%	Progs 30%
Indiv.	73.9	71.2	79.4	85.1	47.8
Weight	22.17	7.12	3.97	21.275	14.34
	Total	→	68.875	⇒	<b>D</b>

# Office hours

- My office is EN4068.
- My office hours are on the WyoCourses site, and the syllabus.
- Open-door policy, if I'm in my office and the door is open, stop in!
- The TA's hours will be posted on the course site when we figure them out.

# The IDE

- We will be using Microsoft Visual Studio for the primary program development environment.
- This is an industry standard.
- There are a LOT of bells and whistles.
- The first Lab will involve getting started with Visual Studio
- Other IDE's and compilers exist – many ways to set up a development environment

## (more ...)

- For this course, Visual Studio will likely be the easiest
- You can get the Community edition FREE direct from Microsoft.
- Note that you MUST create a Microsoft account (also free) to use the software for more than 30 days.
- Hassle but just do it at the beginning.

## (more ...)

- When installing, MAKE SURE that you select the correct programming language to install.
- The default is probably something like F#. You need to make sure that you select C++, NOT C#.
- If you do not when you initially install, it is difficult (nearly impossible) to install later. Best to uninstall and reinstall.

# Lets look at a program.

- There are two things to think about
  - What is my goal with this program?
  - How do I accomplish that goal?
- The first, for this course, will be “do what the Instructor said to.”
- The second is the hard part.

# Syntax versus Semantics



# Syntax versus Semantics

- Syntax – the *correct* structure of a program. That really means: “Does it compile without errors OR warnings?”

# Syntax versus Semantics

- Syntax – the *correct* structure of a program. That really means: “Does it compile without errors OR warnings?”
- Semantics – what does the program **mean**.
- The semantics are what you define when you answer the second question: “How do I accomplish that goal?”

# Our basic program

- This will simply display a line of text.
- We will talk about the critical main() function and the include preprocessor directive.
- Will look at differences in the C++ code due to ANSI/ISO namespace use.

# First pass

```
// SimpleProg.cpp  
// Kim Buckner  
// COSC 1030 Section 14 Fa 2022  
// Jan 13, 2022  
// Display a line of text
```

```
#include <iostream>
```

```
int main()
```

```
{  
    std::cout << "Hello , World!\n";  
    return 0;  
}
```

# First pass

```
C:\> cd \Temp\SimpleProg\Debug
C:\Temp\SimpleProg\Debug> SimpleProg
Hello, World!
C:\Temp\SimpleProg\Debug>
```

# Second Pass

- This next one is the style from the text.
- Note how **cout** is accessed.
- This may be a somewhat more efficient way to access elements of a *namespace*.

# Second pass

```
// SimpleProg2.cpp  
// Do not lose points ,  
// use the correct header comments  
#include <iostream>  
using std::cout;  
  
int main()  
{  
    cout << "Hello again , World!\n";  
    return 0;  
}
```

# Third Pass

- This is a wholesale way to access elements of a *namespace*.
- In this one, all elements of the namespace are available.
- This can result in name clashes.
- **DO NOT** do this in any code you write for this class. Use the second version.



# Third pass

```
// SimpleProg3.cpp
// I won't even claim this one

#include <iostream>
using namespace std;

int main()
{
    cout << "Hello once more, World!\n";
    return 0;
}
```

# Final comments

- Learning to program involves
  - familiarization with the syntax of a programming language,
  - experience with the functionality of both built-in and standard library elements of the language,
  - and formulating concise data structures and algorithms for solving the problems or tasks at hand.
- These are our goals this semester.

# Final comments

- If you are wondering if C++ is relevant and/or useful, [here's an expert opinion](#) from Aug 20, 2023.