CS 170: High-Level Programming II (Introduction to C++ Programming)
Winter/Spring 2020 Course Syllabus

Contact Information

Instructor:	Matthew Mead
Office Hours:	By arrangement (I'm usually here M-F all day and can usually meet anytime I'm not in class.)
Office Phone Number:	895-4441
Email address:	mmead@digipen.edu
Home page (DigiPen):	http://azrael.digipen.edu/~mmead/www/Courses/2020/winter/cs170/index.html

Course Description

This course is a continuation of High Level Programming I (CS 120). It introduces the C++ language with particular emphasis on its object-oriented features. Some of the topics to be covered include stylistic and usage differences between C and C++, namespaces, function and operator overloading, classes, inheritance, exceptions, class and function templates, STL lists and vectors. After successfully completing this course, the student will be prepared for the next programming course in the sequence (CS 225 Advanced C/C++) and will also be able to begin using the C++ language in the second-year game course. CS120 is a prerequisite for this course.

Objectives and Outcomes

After successfully completing this course, the student should be able to read, write, and understand much of the introductory C++ programming language. Specifically, students will be able to:

1. Understand the differences between imperative programming (CS120) and object-oriented programming.
2. Understand the concept of data abstraction and inheritance.
3. Understand the concept of interface versus implementation.
4. Understand the challenges of building large-scale programs and how object-oriented programming facilitates it.
5. Understand the built-in libraries and how to use them effectively in problem solving.
6. Apply the course concepts to implement data structures and programs to solve various problems.

The successful student will be prepared for the next programming course in the sequence (CS 225 Advanced C/C++) and will be able to use the C++ language in the third semester game course.

Section	Day and Time	Room
A	M, 9:00 am - 10:20 am F, 11:00 am - 1:20 pm	Michelangelo Michelangelo
B	M, 12:00 pm - 1:20 pm W, 12:00 pm - 2:20 pm	Michelangelo Michelangelo

Textbooks and References

Required

- *C++ primer plus; 6th ed; Stephen Prata; ISBN: 9780321776402*. Note that the *Fifth* Edition can be used as well, although some of the page and assignment numberings may differ and there isn't any C++ 11 coverage.

Additional references (Optional)

- *The C++ Programming Language, Fourth Edition, by Bjarne Stroustrup. Published by Addison-Wesley Publishing Company. Copyright © 2013. (ISBN-10: 0321563840, ISBN-13: 978-0321563842).*
- *The World Wide Web*. Quite possibly the greatest asset to learning since the teacher and the textbook.

Grading

Grades will be derived from homework assignments and exams. The detailed weightings and letter grades are as such:

Midterm exam	20%
Homework	35% (assignments: 20%, labs: 15%)
Final exam	45%

You must receive an average score of 60% on both the midterm and final exams combined to pass this course, regardless of your homework/lab scores.

x%	Grade
$x \geq 93$	A
$90 \leq x < 93$	A-
$87 \leq x < 90$	B+
$83 \leq x < 87$	B
$80 \leq x < 83$	B-
$77 \leq x < 80$	C+
$73 \leq x < 77$	C
$70 \leq x < 73$	C-
$60 \leq x < 70$	D
$x < 60$	F

Attendance is mandatory. There are no makeup exams. Also, for every lecture that is missed, you will lose one point from your final grade (e.g. a 90 becomes an 89). The only exceptions are if you notify me prior to your absence (or shortly thereafter if you were unable to notify me due to illness) with a valid reason. (Sleeping, studying for another class, working on your game, etc., are not valid reasons for an absence.) For freshmen courses, such as CS170, there is a school-wide Digipen Freshmen Attendance Policy that must be followed. Read it on the website. If you do have to miss a class (for any reason), it is your responsibility to contact other classmates to find out what was covered.

Disabled Student Services
Students with physical, psychological or learning disabilities that affect their ability to perform major life activities associated with this class may be eligible for reasonable accommodations under the Americans with Disabilities Act. If you have a documented disability please contact the Disability Support Services office to arrange for accommodations for this class.

Tentative Schedule (This is a very coarse guideline for the semester and is subject to change.)
Below is a list of topics that will be covered this semester. Depending on time, I may add additional topics or skip some of the ones listed. The right-hand column corresponds to the pages in the text book that covers the topic. The readings should be done before the lecture is given. There will be several graded programming assignments throughout the semester. There will also be labs each week. The programming will require between 2 and 5 hours per week, depending on your grasp of the subject matter and your performance in CS120.

Tentative Schedule (topics and ordering subject to change)

Wk	***Topic***	***Required reading (6th and 5th editions)***
1	Intro to CS170, C vs. C++, namespaces	Ch. 1, 2, 9 (*6th* 482-497, *5th* 424-437)
2	Formatted output	Ch. 17 (*6th* 1061-1097, *5th* 951-982)
3	Functions (overloaded, default parameters, references)	Ch. 8 (*6th* 381-418, *5th* 337-370)
4	Classes and objects	Ch. 10
5	Operator overloading	Ch. 11
6	Constructors, destructors, assignment	Ch. 12
7	Inheritance, polymorphism, base classes, virtual methods	Ch. 13
8	Aggregation (containment) **Midterm**	Ch. 14 (*6th* 785-797, *5th* 701-712)
9	Function templates	Ch. 8 (*6th* 419-424, 5*th* 370-375)
10	Class templates	Ch. 14 (*6th* 830-843, 5*th* 744-756)
11	**Spring Break**	
12	Introduction to the STL, the standard string class, vectors, lists, iterators	Ch. 16 (*6th* 951-968 and 978-996, 5*th* 857-894)
13	File I/O	Ch. 17 (*6th* 1114-1133, 5*th* 1003-1029)
14	Exceptions	Ch. 15 (*6th* 896-921, 5*th* 805-829)
15	**TBD**	
16	**Finals**	

Workload
During the semester there will be two major exams (midterm and final). There will be several programming assignments and labs to work on outside of class. These are not large and you will be given enough time to complete them (although they generally take no more than a few hours to complete). In addition to attending the lectures, you should plan to spend at least 6 hours per week reading, studying, and programming for this class.

Submitting Homework
Programming assignments will (obviously) use the C++ programming language. More specifically, all programs must adhere to Standard C++, which is what this course is about. Assignments will be graded using the GNU C++ compiler version 8.x, but the assignments may also need to be compiled and tested with other compilers (e.g. Microsoft) and will be specified in the assignment. You are encouraged to build and run your programs with many compilers, since this is the only way to help ensure that your code is legal and robust. Any submissions that do not compile cleanly will not be accepted and will result in a grade of 0. Additional detailed instructions will be provided with each assignment. This usually includes a handout as well as supplemental material available on the web site.

Each homework assignment that is to be submitted will specify the time/date that it is due. Late assignments will not be accepted. There is more than enough time in your schedule to complete all homework assignments on time. Of course, if you wait until the due date/time is near to begin your homework, you may not finish it on time. Time management is your responsibility.

Code Documentation and Conventions
Unless otherwise specified, all programs must be documented using Doxygen tags. There is an extensive help document included with Doxygen which shows examples of each tag. (I will be covering the use of Doxygen during the first weeks of the course.) In addition to the "normal" documentation (file and function header comments), you must also put comments next to each header file that you include. The comments must list the functions, types, variables, symbols, etc. that you are using from that library. The purpose of this is for you to demonstrate to the graders and myself that you know why you are including these libraries in your project. Here are a few examples (note the formatting and alignment of comments):

```
#include <stdio.h>    /* NULL, printf              */
#include <stdlib.h>   /* atoi                      */
#include <algorithm>  /* sort, transform, for_each */
#include <functional> /* bind2nd, ptr_fun          */
```

As a rule, absolutely **no implementation by the student is allowed in any header file for any reason**. Go back and re-read that last sentence. Also, we will still be following the CS120 Style Guide that is posted on the website.

Academic Honesty
All homework/lab assignments and exams must represent your own, individual work. It is permissible to discuss assignments (not solutions) with other students in the class, but the solutions must be recognizably your own. Cheating of any kind (copying someone else's work, allowing others to copy your work, collaborating, etc.) will not be tolerated and will be dealt with SEVERELY (at the discretion of the instructor, which may likely include removal from the class with a grade of F.) Please keep in mind that discussing solutions to exams, homework, etc. with students that haven't taken the exam or turned in the assignment is also prohibited. Ultimately, you are only wasting your time (and money) because if you can't master the fundamentals covered in this course, you have little hope of succeeding in other courses or as a programmer in the Real World.

From The "It-shouldn't-need-to-be-said-but..." Department
During class, all electronic devices must be turned **OFF**. This includes cell phones, game consoles, digital cameras, computers, or any other devices. If you absolutely must have a cell phone on for an expected emergency situation, you must put your phone on vibrate and clear it with me **BEFORE** class begins so I know to expect that you may have to leave class early.

In addition to showing up for class on time, other student responsibilities include proper behavior during class, learning the material, completing assignments correctly, submitting assignments properly and on time, studying for the exams, and participating in class by asking or answering questions during the lectures.