CIS 29 - Syllabus

Advanced C++ Programming
4.5 Unit(s)

Instructor: Grant Larkin
Email: larkingrant@fhda.edu

Hours:
Lecture - 6:00 to 6:50
Lab - 7:00 to 7:50
Office Hours – 4:30 to 5:45 W

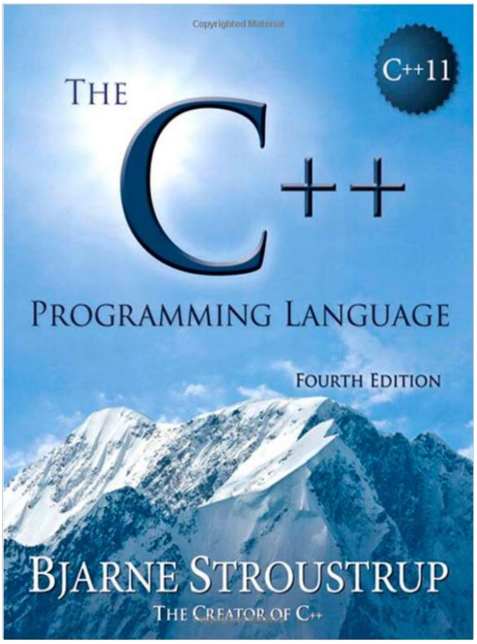
Requisites:
Prerequisite: CIS 22B or 27 or equivalent.
Advisory: MATH 212 or equivalent.

Description:
Learn Advanced C++ language features and techniques to design and program classical computer and not-so-classical algorithms. Topics include: Templates, Exceptions, Lambda expressions, STL Containers/Algorithms, Threading, the new C++11/14 features, basic UML and programming style.

Student Learning Outcome Statements (SLO)
- Student Learning Outcome: Create C++ programs using standard classes, advanced operators, multiple inheritance, and exception handling.
- Student Learning Outcome: Create and use libraries with the C++ language.
- Student Learning Outcome: Create and use templates, including the Standard Template Library, in C++ programs.

Course Objectives:
A. Apply standard classes
B. Apply operator overloading like new and delete
C. Demonstrate classes using multiple inheritance and polymorphism
D. Create exception handling procedures
E. Create and use libraries in compilation
F. Create and use templates
G. Create programs using the Standard Template Library
H. Create programs which use good programming style and object oriented design techniques.

Text Book:
Stroustrup, Bjarne; Pearson Education, "C++ Programming Language", Fourth Edition, 2013.
ISBN-10: 0321563840, ISBN-13: 978-0321563842



Grades:
Labs - 50%
UML - 10%
Tests - 40%

Compilers:
• Preferred C++ compiler is Visual Studio C++. There is a Windows and Macbook version available from Microsoft Imagine. You will receive an Account with Microsoft Imagine 2 weeks after the start of the quarter. While waiting for your account you can download Visual Studio Express from Microsoft.
• Eclipse has a C++ version available that depends on Cygwin or Mingw. Installing this version is not straight forward. These are platform independent versions. However, I don't have installation instructions for all platforms.
• Code::Blocks is used by many courses on this campus. It's open source and multi-platform. However, it is not a good C++ compiler, generates sub-standard code, and doesn't have good debugging tools.
• Xcode for Apple platforms supports C++, Objective-C, and Swift. However, it's code generation is average and is a difficult tool for debugging C++ programs.

UML Diagram Editor:
This course doesn't require that student have UML diagram experience. To assist with the process of learn how to create effective classes, UML editors are a good tool. One of the best is a free, open-source tool with a simple user interface: http://umlet.com/. A demonstration will be given on the first class session.

Labs:

Please follow the lab specifications carefully. Combine all your source code into ONE C++ file and upload the file to Canvas, and include the .H files in the C++ file, as this makes grading the assignment easier. Use Advanced C++ features in preference to standard C++ features. Labs also include a prerequisite UML Class diagram, and the diagram is due 5 days before lab assignment is due. The purpose of this is to compose the design of the program before coding the solution.

Late Labs:
Late labs may be turned in after the due date, with a 30% point deduction. Late UML Class diagrams are not accepted.

Tests:
The midterm may or may not be given, depending on the instructor's discretion. A final is required by the CIS Department; however, an optional final project can be substituted. The tests are open-book, open-notes, and restricted Internet. The tests are posted on Canvas.

Academic Integrity:
This course expects student to do their own work. While this course encourages groups to get together to discuss topics related to the courses, it discourages group members from copying each other's work. Also, there are many available websites that offer help with assignments. Students should not request help from these resources.