University of Montana

ScholarWorks at University of Montana

Syllabi

Course Syllabi

1-2015

CSCI 113.01: Programming with C++

Thomas Gallagher

University of Montana - Missoula, thomas.gallagher@umontana.edu

The University Of Montana - Missoula
Missoula College
Department of Applied Computing and Electronics
Course Syllabus

CSCI 113 Programming with C++
Credits: 3
Term: Spring 2015
Prerequisites: M 90 Introductory Algebra
Syllabus Last Revised: January 2015
Class Meetings: MW 11-12 GH 9 TR 11-12 HB04
Final Exam Meeting: Thursday, May 14 10:00-12:00

Faculty Contact:

Tom Gallagher - Thomas.Gallagher@umontana.edu
Office Hours: MTWR 8:30-9:30

Phone: 406.243.7814
Office Location: Griz House 8
(NE Building on COT East Campus)

Course Description:

Object oriented programming using C++. Implementation of structured programming concepts along with construction of classes to create data types for defining objects

Course Overview:

C is one of the most widely used programming languages. It was first developed by Dennis Ritchie in the late 1960s at AT&T Bell Labs. C is a procedural language widely utilized in programming hardware devices. C++ was first developed by Bjarne Stroustrup in 1979. C++ can be thought of as an enhanced version of C. C++ adds the use of classes in forming an object-oriented paradigm for programming. C++ was initially called C with Classes, but was renamed C++ in 1983. We will study the object-oriented principles of encapsulation, data abstraction, inheritance and polymorphism found as implemented in the C++ language.

There are numerous C++ compilers and integrated development environments available to programmers for nearly all computing platforms. In this class, we've chosen the Eclipse IDE as our development tool and the GNU C/C++ Compiler.

We'll also discuss the C language as a hardware programming tool. The Raspberry Pi (Linux OS) will be used as a platform for hardware-oriented programming.

Course Outcomes:

Upon completing this course, a student will be able to:

- Design and implement programs using C++;
- Explain and be able to use data types, variables, constants, assignment statements, and arithmetic and Boolean expressions in writing programs;
- Explain and be able to use fundamental programming constructs such as sequencing, decisions and iteration;
- Explain and be able to use fundamental object oriented principles such as classes, objects, methods, encapsulation, data hiding, inheritance and polymorphism;
- Explain and be able to use arrays and structures;
- Explain and be able to use exception handling

Required Textbook:

Gaddis T. (2011). *Starting out with C++: From control structures through objects*. Pearson. 7th ed. ISBN 978-0-13-257625-3

Software (compiler & IDE)

MinGW (Minimalism GNU for Windows) C/C++
Free download: http://sourceforge.net/projects/mingw/files/

Eclipse IDE
Free download: http://www.eclipse.org/downloads/

For installation instructions see:
https://www3.ntu.edu.sg/home/ehchua/programming/howto/EclipseCpp_HowTo.html

Other Materials

SD Card (prefer 8GB or greater)

Raspberry Pi Computing System. These will be provided for a series of classroom labs. They are an interesting development platform as they are a complete computing system at a retail price of $29.95. http://www.amazon.com/Raspberry-Pi-Model-512MB-Computer/dp/B00LPESRUK

Assessment

Grades will be weighted and graded as follows:

Assessment Area Weighting:		Grading Scale:	
Programming Assignments	40%	90-100%	A
Unit Quizzes	30%	80-89%	B
Homework	20%	70-79%	C
Final Project	10%	60-69%	D

Academic Integrity:

All students must practice academic honesty. Academic misconduct is subject to an academic penalty by the course instructor and /or a disciplinary sanction by The University. All students need to be familiar with the Student Conduct Code. The Code is available for review online at: http://life.umt.edu/vpsa/student_conduct.php

Using the Web to research materials and concepts is an integral part of learning in the twenty-first century. Studying with other students is a productive method of learning. A certain amount of collaborating on concepts with other students and using resources found on the Internet in an assignment is recommended. Copy and paste is not acceptable. It is expected that each student will input his/her assignment into the computer, and each student must be able to explain any assignment turned in. Collaboration on exams is strictly forbidden.

Collaboration on homework assignments is encouraged. Collaboration on programming assignments is prohibited.

Dropping and Adding Courses or Changing Sections, Grading or Credit Status

University Policy for dropping courses or requesting grading/credit status changes can be found in the catalog: http://www.umt.edu/catalog/acad/acadpolicy/default.html Students should become familiar with all academic policies

Disability Accommodations:

Eligible students with disabilities will receive appropriate accommodations in this course when requested in a timely way. Please contact me if you will be requesting an accommodation. Please be prepared to provide a letter from your DSS Coordinator. For more information, visit the Disability Services website at http://www.umt.edu/dss or call/text 406.243.2243.

Late Assignment Policy:

All quizzes and programming assignment are to be completed on the assigned date and time. Late assignments will be accepted without appropriate justification and only at the discretion of the instructor. Rescheduling of a quiz will be approved at the discretion of the instructor and only in extraordinary situations.

Learning Management System:

It is the responsibility of the student to access and familiarize herself/himself with the Learning Management System (LMS) for the course (Moodle). Access & training is available through UMOnline http://moodle.umt.edu

Changes to Syllabi:

Note: Instructor reserves the right to modify syllabi and assignments as needed based on faculty, student, and/or environmental circumstances. If changes are made to the syllabus, amended copies will be dated and made available to the class.

CSCI 113 Curriculum Outline:

Unit 1 Introduction, Variables, Mathematical Expressions, & I/O (Gaddis Ch. 1-3)

1 Introduction to Course
2 Introduction to Programming (Gaddis Ch. 1)
3 C++ Programming: Structure and Output (Gaddis 2.1-2.3)
4 Variables and Data Typing (Gaddis 2.4-2.13)
5 Arithmetic Operators (Gaddis 2.14-2.18)
6 Input, Type Casting, and Mathematical Expressions (Gaddis 3.1-3.6)
7 Formatting Output and Strings (Gaddis 3.7 – 3.11)

Unit 2 Decision Structures, Loops, and Files (Gaddis Ch. 4-5)

1 Relational Operators
2 *if/else* statement and Logical Operators
3 *switch* statement
4 Loops: *while* and *do-while* loop
5 Loops – *for* loop
6 Introduction to Files

Unit 3 Functions (Gaddis Ch. 6)

1 Introduction, Definition, Prototypes
2 Sending data, reference variables as value, and returning values
3 Variable Scope & Namespace
4 Reference variables as parameters
5 Function Overload and Stubs

Unit 4 Arrays and Search/Sort Algorithms (Gaddis Ch. 7 & 8)

1 Introduction to Arrays
2 Arrays & Functions
3 Two-dimensional Arrays
4 Search and Sort Algorithm(s)

Unit 5 C Programming & Bitwise Operations (Gupta 2005)

1 Introduction to C & I/O
2 Working with Decimal, Hexadecimal, and Bits
3 Constants and Macros
4 Bitwise Operators

Unit 6 C Programming for Hardware - TBA

Unit 7 Pointers and Structures (Gaddis Ch. 9 & 11)

1 Introduction to pointers, arrays, and pointer arithmetic variables, values, & addresses
2 Dynamic Memory and Pointers as Function Parameters
3 Structures: Abstract Data Types (ADTs) and Arrays
4 Structures: Functions & Pointers

Unit 8 C++ Classes (Gaddis Ch. 13)

1. Introduction to Classes
2. Class: Pointers, Constructors, Destructors, & Memberwise Assignments
3. Class Implementation, Header Files, and UML