ISC-3313 - Introduction to Scientific Computing
Programming language: C++
Spring 2012

Course Description

This course introduces students to the science of computation. We will cover algorithms for standard problems in computational science, as well as the basics of an object-oriented programming language. The programming language will be C++, a multi-platform language that is the preferred language of many professional developers. Examples will be biased toward biological applications to provide students from a computational/mathematics background interesting and, perhaps unfamiliar, real-world applications and those from a biological background with a familiar application domain.

Instructor

Dennis E. Slice
Associate Professor
Department of Scientific Computing

Office: 150-U Dirac Science Library
Phone: (850) 645 1324
Email: dslice at fsu dot edu
Web (Blackboard): http://campus.fsu.edu

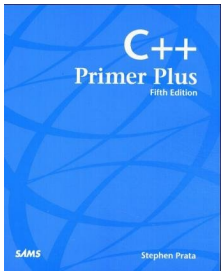
When and Where

11:15a – 12:05p
Monday, Wednesday, and Friday
DSL 0152
Office Hours: 1:00p – 3:00p Wednesdays, other times by appointment or serendipity (not before class).

Prerequisites

MAC 2311, MAC 2312.

Text



Any general C++ reference will do. I will be using...

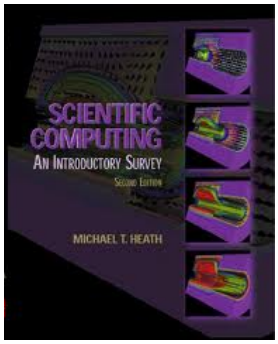
C++ Primer Plus (5th Edition) by Stephen Prata
ISBN-13: 978-0672326974

And I will be referring to...

Scientific Computing: An Introductory Survey (2nd Edition) by Michael T. Heath

ISBN-13: 978-0072399103

...but it is neither required nor (at this time) recommended.



Course Objectives

After successful completion of this course students will be able to:

1. Identify the components of scientific computing.
2. Identify standard problems in scientific computing.
3. Implement basic algorithms for standard problems in computational science using the programming language C++.
4. Write, debug, and verify computer codes.
5. Output results of computer simulations on a meaningful manner.

Computer Competency Requirement

In order to fulfill FSU's Computer Competency Requirement, the student must earn a "C-" or better in the course, and in order to receive a "C-" or better in the course, the student must earn at least a "C-" on the computer competency component of the course. If the student does not earn a "C-" or better on the computer competency component of the course, the student will not earn an overall grade of "C-" or better in the course, no matter how well the student performs in the remaining portion of the course.

Grading Policy

The student's grade for the course will be based upon classwork, homework, a midterm and a final student project. This work is weighted as follows:

- Classwork/Homework – 50% (10 x top 5 assignment scores, including project proposal)
- Midterm Exam - 20%
- Final Project – 30%
- Bonus Quizzes (up to 10) – 5%

Homework/Project Submission

Each homework assignment or project must be submitted as PDF documents and/or C++ source and header files in a single tar file via email to the instructor.

Projects will be graded on:

1. minimum functionality
2. robustness
3. code organization
4. documentation (including user's manual and code comments)
5. complete, correct, concise answers to questions

University Attendance Policy

Excused absences include documented illness, deaths in the family and other documented crises, call to active military duty or jury duty, religious holy days, and official University activities. These absences will be accommodated in a way that does not arbitrarily penalize students who have a valid excuse. Consideration will also be given to students whose dependent children experience serious illness.

Academic Honor Policy

The Florida State University Academic Honor Policy outlines the University's expectations for the integrity of students' academic work, the procedures for resolving alleged violations of those expectations, and the rights and responsibilities of students and faculty members throughout the process. Students are responsible for reading the Academic Honor Policy and for living up to their pledge to ". . . be honest and truthful and . . . [to] strive for personal and institutional integrity at Florida State University." (Florida State University Academic Honor Policy, found at http://dof.fsu.edu/honorpolicy.htm.)

Americans With Disabilities Act

Students with disabilities needing academic accommodation should:

(1) register with and provide documentation to the Student Disability Resource Center; and

(2) bring a letter to the instructor indicating the need for accommodation and what type. This should be done during the first week of class.

This syllabus and other class materials are available in alternative format upon request.

For more information about services available to FSU students with disabilities, contact the:

Student Disability Resource Center
874 Traditions Way
108 Student Services Building
Florida State University
Tallahassee, FL 32306-4167
(850) 644-9566 (voice)
(850) 644-8504 (TDD)
sdrc@admin.fsu.edu
http://www.disabilitycenter.fsu.edu/

Syllabus Change Policy

Except for changes that substantially affect implementation of the evaluation (grading) statement, this syllabus is a guide for the course and is subject to change with advance notice.

Content

I Components of Scientific Computing

1. Introductions
2. UNIX basics

II C++ basics

1. Editing
2. Minimal program structure
3. g++
4. Simple data types
5. Visualization

III A simple example – who doesn't like pi?

1. Netbeans IDE: an integrated development environment
2. Algorithm development
3. Program testing and documentation
4. Visualization and analysis of results

IV Solving a non-linear equations

1. Description of problem and some simple algorithms
2. Iterative methods, required accuracy of result
3. Implementation of the Bisection method
4. Program testing and documentation

V Object-oriented programming (OOP) – non-linear eq. cont.

1. Encapsulation
2. Inheritance
3. Polymorphism
4. Abstract classes and datatypes

VI Operations on vectors and matrices

1. Development of general functionality that is usable in many places

2. Vector and Matrix operations
3. Vector norms
4. Concurrency and parallel processing of such calculations

VII Polynomial interpolation of data

1. Description of problems and (biological) applications
2. Algorithms: Lagrangian interpolation in detail
3. Implementation to fit a set of data
4. Piecewise interpolation
5. Implementation and visualization of of piecewise interpolation

VIII Solving ordinary differential equations systems

1. Description of problem: Lotka-Volterra Predator-Prey system
2. Algorithms
3. How to use functions from other libraries
4. How to assess correctness of program
5. Visualization of results

IX Markov chain Monte Carlo Integration

1. Description of method
2. Example application
3. Implementation
4. Testing and visualization of results

X Final project