
CGS 3406

Course Syllabus

Spring Term 2010

Course Instructor : Prof. A. Ford Tyson

Office Location: 105D Love Building

Phone: 850-644-0973

Office Hours : see listing on course Blackboard site under Staff Information

Email Address: aftyson@cs.fsu.edu

Teaching Assistants : names and contact information are posted on course Blackboard site

Office Location: 114 MCH

Office Hours and Email Addresses: see listing on Blackboard site under Staff Information

ATTENDANCE AND CLASS SCHEDULE

This class is offered as two lectures per week and one recitation section per week. Attendance at all lectures and recitations is required to do well in this course.

COURSE FOCUS AND OBJECTIVES

This course is intended for non-majors in computer science or related areas and focuses on the fundamental concepts of computer programming using the C++ language. This course may *not* be used as a programming pre-requisite for COP 3330.

By the end of the semester, a student who has completed this course with a passing grade should be able to:

- Demonstrate a basic understanding of fundamental computer science concepts, including software and hardware.
- Solve computing problems using a top-down approach in a well-structured design in the procedural paradigm. Understand introductory basics of an object-oriented approach.
- Design, implement, test and debug a C++ program to solve a given problem.
- Demonstrate knowledge and use of control structures including sequence, selection, iteration and functions.
- Make use of data types and structures in C++ including integer and floating point types, arrays (one-dimensional, two-dimensional, strings) and structs; pointer basics, arrays of structs and structs containing arrays. Have a basic understanding of the C++ class and class operations. Be able to write a program based on a class data structure, utilize class objects, and provide appropriate class member functions.

- Utilize fundamental algorithms studied to perform tasks such as finding max and min in a data set, counting, summing, tracking a previous value, searching and sorting, reading until EOF, etc.
- Consider, compare and evaluate code segments or algorithms for relative efficiency in a basic fashion.
- Time permitting, be able to understand and recognize the basics of recursion.
- Time permitting, be able to understand the basics of setting up and using linked lists.

COURSE MATHEMATICS PRE-REQUISITE

All students taking CGS 3406 are required to have *previously* taken and passed (with a C- or higher final grade) MAC 1105. If you have not taken and passed this course or its equivalent, you will be automatically dropped by the CS department staff.

REQUIRED TEXTBOOKS

Starting Out with C++ Brief Version: From Control Structures through Objects ISBN 978-0136022534

Tony Gaddis, 6th Edition, Pearson Addison Wesley Publishers, 2009

Practical Debugging in C++ ISBN 0-536-43844-7

Ann Ford Tyson, Toby Teorey, Gary Tyson, 2nd Edition, Pearson Custom Publishing, 2007

BLACKBOARD WEB SITE

Course materials are made available on the class Blackboard web site, all of which is considered required reading. You can access the class Blackboard site at <https://campus.fsu.edu/webapps/login>. In the "Syllabus and Other Fundamental Information" section you will find this *handout* as well as other important information. Read this syllabus entirely and carefully as it contains vital details which will affect your grade in this course.

Other web site materials will include

- additional required and reference handouts
- current assignment write-ups and related information
- links to turn in your programming projects
- some solutions to current semester programming projects, as they become available
- prior semester exams and answer keys for practice
- important course announcements

EMAIL

The best way to contact the teaching staff outside of class and office hours is via email. We will send class email using the Blackboard site for this course. Email sent to the class email group is considered required reading. The class email group is to be used by the teaching staff for official course matters only. For example, if we need to send information about exam room locations, we will send email to everyone in the class and we will assume that everyone in the class does receive and read it. *Please note that email will be sent to your FSU email address only. When you send email to the teaching staff, only email sent from an FSU email account is guaranteed to be read by the teaching staff.* Email sent from

sites such as yahoo, hotmail etc. will often be filtered out by automatic spam filters and deleted without being read.

OVERVIEW

In this course you will learn to write programs in the C++ programming language using good style, structure and design. No prior programming knowledge is assumed, however this is not a course in how to *use* computer software such as word processors and web browsers. This course is concerned with how to *design and write* computer software. Six programming projects will be assigned during the term, some of which will require a large time commitment on your part. Many students find the work load in this course to be heavy, especially during the last half of the course. Assignments will begin with simple problems, and the difficulty level and length will increase as the term progresses. The first project will not require you to write a program yourself, but will consist of a simple exercise which will familiarize you with our computer system and the C++ compiler we use.

We will use the *Microsoft Visual C++ Express Edition 2008 compiler* exclusively, available on computers at campus computing sites and for free download from the internet. All projects turned in must be completed using ANSI standard C++ with this compiler, and must compile and run successfully using this specific compiler, on a Windows machine. Note that the only supported platform in this course is Windows and the only supported compiler is this one.

TOPICS

We will start discussing programming fundamentals and the C++ language almost immediately. Some time will be spent addressing basic computer concepts and the use of the C++ compiler, but for the most part these topics will be covered in the early assigned readings and in your recitation section. The following topics will be covered, in this approximate order:

- Introduction to computers (hardware and software) and programming
- Computing facilities we use: FSU computing sites, the C++ compiler
- Introduction to C++, problem solving and algorithms; declarations
- Simple data types: integer and floating point, char, bool
- Input/output, including file input
- Assignment, operators, expressions, standard mathematical functions
- Conditional statements and logical expressions; if, switch statements
- Looping structures; while, for, do while statements
- Functions, parameter passing, scope, storage classes, macros
- Number representation and common calculation errors
- Arrays: 1-dimensional and multi-dimensional
- Sorting (selection sort) and Searching (linear search, binary search)
- Pointers and an introduction to dynamic memory allocation
- Strings (both C-style strings and the C++ string class)
- Structs
- Abstract Data Types; using data structures built from structs and arrays
- Classes, class data and function members, objects, fundamental operations and concepts using classes
- Time permitting: introductions to recursion and dynamic data structures
- Throughout the course we will discuss good programming style, design, documentation, and

efficiency using a procedural paradigm

PROGRAM ASSIGNMENTS

A sincere word of advice: we have found through many years of teaching experience that the most common reason for not doing well on the projects is not starting them early enough. You will be given plenty of time to complete each project. However, if you wait until the last minute or the last weekend before the due date to start, you may not be able to finish. Plan to do some work on the project every day. Also try to have it finished a few days ahead of the due date - many unexpected problems arise during programming, especially in the debugging phase. In addition, the computing sites can become very crowded and make it difficult to get a computer to use. Blackboard can become overloaded when projects are due and it can be a very slow process to submit your work. You must PLAN for these things to happen and allow ample time to meet course due deadlines. Meeting course deadlines is *your* responsibility.

There are many sources of help for you while programming for this course. First, recitation sections and office hours. You can send questions to the teaching staff via email at this address: 3406help@cs.fsu.edu. Keep in mind that many types of questions cannot be answered without *seeing* your work. Also keep in mind that if everyone puts off working on their programs until the last minute, we will probably not be able to help everyone adequately in office hours; the teaching staff just can't handle all students in the course coming in all at once in the day or two before a program is due.

Some computing sites may have staff consultants who are available to help you at the site. They are good sources of help with email problems, using the computers etc. However they are not there to help you with your program design or with writing your program.

Whenever you want help with your program in office hours, you must be able to access your program file via CS department file space (your "Z" drive) or a USB flash drive, and bring a *CURRENT* print-out of the program, input, and the printed results of a run.

DEADLINES & FOLLOWING COURSE INSTRUCTIONS

A large part of your grade in this course will be determined by whether or not you can meet deadlines and follow course instructions correctly. Both are very important in the real world, and they are critical skills in this course. For example, be sure that you read the handout entitled "Submitting Your Program Assignments Electronically Using Blackboard" carefully and follows its instructions exactly.

DISKS & BACKUP COPIES OF YOUR PROGRAMMING ASSIGNMENTS

Always make multiple backup copies (on a disk or stored in the university's file space, and printed) of your work!!! This is a *course requirement*, and also common sense. The teaching staff may under certain circumstances have to ask you to produce your backup file copies.

PROGRAM & HOMEWORK DEADLINES

Due Times: Each programming and homework assignment will be given a due date. Your work must be turned in by 11:59 pm on the due date to be considered on time.

Grace Period: Work turned in by 11:59 pm by 48 hours after the due date will be accepted but will automatically lose 20% of the possible points. Work not turned in by this time will *not* be accepted and will automatically receive a grade of zero.

For example, if an assignment is due on a Monday, you must turn it in by 11:59 pm on that Monday for it to be considered on time. If it is turned in by 11:59 pm on the subsequent Wednesday, it will be accepted, considered late, and will incur a 20% point penalty. After this Wednesday deadline, the assignment will not be accepted, and you will receive a grade of zero for it.

Programming Project 6 must be turned in by the absolute due deadline; it cannot be turned in late because it is due at the end of the term and the teaching staff needs adequate time to grade it before final course letter grades are due.

Extension Policy: The above grace period is designed to cover normal expected and unexpected delays. Deadline extension requests will only be considered if you contact the course instructor in person during office hours. Such requests will normally only be considered if they are made before the assignment is due, and will only be granted for medical or personal emergencies. In all cases, be prepared to substantiate any extension request with written proof consisting of original and verifiable documents, for example, a letter from your medical doctor, a hospital emergency room receipt, or written funeral home documentation.

Extensions are NEVER granted via email.

Extensions are not granted for reasons such as:

- You could not get to a computer because the university computing sites were crowded
- Blackboard access became slow on the due date
- You (or your dog, roommate, etc.) erased your files by mistake, or you lost your work somehow
- You had other course work or job commitments which interfered with your work in this course

There will be no extensions whatsoever on the last assignment, Programming Project 6, except in extremely unusual circumstances such as emergency hospitalization.

You can avoid all problems by starting the assignments early and working on them incrementally, as discussed in class. If you are having trouble understanding the material or working on your assignments, we encourage you to come to office hours for help right away.

PROGRAMMING PROJECT GRADING & SUBMISSION

The projects will be graded in part for running correctly (doing all the required tasks and giving correct results), and this is worth 50% of the project grade. The other 50% of the grade is based on your program style, documentation, design and efficiency.

If a program you turn in does not compile without errors under the *required* compiler and platform, for whatever reason, a maximum penalty of 50 points may be taken off. If your program does not compile successfully, it indicates that you have not tested it properly before turning it in. Be careful when you turn in your program - even a single missing punctuation mark can cause your program to fail the compilation process. During the grading process, if a trivial syntax error is found, the graders may, at their discretion, take off -5 for each trivial error, up to three errors maximum. If more errors are found, or more serious errors are found, the penalty will be -50 points.

All programming assignments must be submitted by the deadline time in electronic form according to class requirements. The detailed requirements of these electronic submissions are provided in the handout on the class web site entitled "Submitting Your Program Assignments Electronically Using

Blackboard," which is required reading. Be sure you read and understand the *entire* submission handout and follow the instructions carefully.

Note that no programming assignments will be accepted via email under any circumstances.

ACADEMIC INTEGRITY

The basic course rule is that you may not give or receive substantial assistance for any work you are submitting as your own. In all cases in which we have reason to believe that cheating has occurred, we will submit relevant materials to appropriate university authorities for evaluation. If a violation of university academic standards has occurred, a zero will be given on the project or exam in question and other sanctions may be determined as well.

Students are expected to uphold the Academic Honor Code published in the Florida State University Bulletin and the Student Handbook. The Academic Honor System of the Florida State University is based on the premise that each student has the responsibility (1) to uphold the highest standards of academic integrity in the student's own work, (2) to refuse to tolerate violations of academic integrity in the university community and (3) to foster a high sense of integrity and social responsibility on the part of the university community. You can find the honor code on line at http://registrar.fsu.edu/student_handbook/.

Keep this in mind: If you are having trouble finishing an assignment, it is far better to do your own work and receive a low score than to go through an academic integrity investigation and suffer any penalties which may be involved, which can be very severe.

What is cheating on a programming project? (a few examples)

- having someone else write your program, in whole or in part
- copying a program someone else wrote, in whole or in part
- collaborating with someone else to the extent that the programs are identifiably extremely similar, in whole or in part

In all of the above, it is not relevant whether the "someone else" is a friend, a tutor, a complete stranger, a textbook or an internet web site.

In this course, all programming projects are to be done ON YOUR OWN unless otherwise stated in writing by the instructor on the assignment write-up itself.

What is not cheating? (a few examples)

- talking to someone in general about topics and concepts involved
- asking someone for help with a specific bug or error message in your program
- getting help with the specifics of C++ syntax
- utilizing information given to you by the teaching staff of the course, for example copying a paragraph describing the program from the assignment write-up we provide to you
- copying parts of code from a required textbook used this semester in this course; you must cite as a reference the textbook and page(s) used in your program comments

Generally speaking, *talking* about course work is ok; *sharing, using, looking at or reading ANY form of printed, written, electronic or hand-drawn material* is a violation of academic integrity policies.

Obtaining inappropriate material from the internet is also considered a violation.

GRADING AND EXAMS

Final grades will be based on the total points earned on six programming projects and two exams.

You are required to take both the first and second exams at the scheduled times. If a serious medical or personal emergency is not involved in your missing an exam, you will receive a zero for that exam. If a serious medical or personal emergency results in a missed exam, you must come in person to the course instructor's office hours and substantiate any related request for special consideration with written proof consisting of original and verifiable documents, for example, a letter from your medical doctor, a hospital emergency room receipt, or written funeral home documentation.

If you anticipate an exam in another course or a religious holiday which conflicts with our exam time, you must notify the instructor at least two weeks before the exam date in person during office hours, or the request will not be considered. The exam dates are given at the beginning of the term so that you can avoid scheduling job interviews or other commitments on exam days, hence job interviews etc. are not considered valid reasons for missing an exam.

ATTENDANCE and PARTICIPATION

Attendance will not be formally recorded this semester, with the exception of the first day of classes. However, your level of class participation (actually attending, doing in-class exercises, asking and answering questions) will be taken into account in final course grading, especially if your point total falls on a borderline between letter grades.

Note that your lecture instructor and your recitation instructor pay close attention to who is attending and who is not, and this will make a difference in your course grade and in other course issues. For example, let's say that you have not been attending, and then you ask for special help outside of class and office hours, or for other special consideration - what will your instructor think? That you are not "holding up your end" in this class. We will all go out of our way to help people who are attending class and we can see are working hard, but if we can see you are not, well, that's different. Imagine yourself as a teacher, how would you respond to someone who comes to class every time, knows the readings, and is clearly working conscientiously? What about to someone who rarely comes to class and has missed so much information they don't seem to have any idea what's going on?

It is against university policy and academic principles to register for two classes which meet at the same time, whether in whole or in part. The teaching staff of this course will make no special arrangements whatsoever for anyone who somehow registers for overlapping course meetings.

All students are required to attend the recitation section that they are officially registered for.

GRADING QUESTIONS

All grading questions involving all course work must first be discussed with your recitation instructor. If the questions involve a re-grading request, you must attach a cover page to the printed assignment or exam, where the latter describes your concerns and is specific about just what you are contesting and why you think it deserves re-grading. Your recitation instructor will then consider your request, make changes when deserved, note these on the cover page, and then record a grade change (if it differs). If you remain unsatisfied, then bring your printed assignment and the cover sheet (as already annotated by your recitation instructor) to the course lecture instructor in person during scheduled office hours. Note that if

you want any part of an assignment re-graded, the entire assignment will be re-graded. All requests to review the grade for a particular item must be made within 7 days after the date that your assignment grade or an exam key is provided to you. Special deadlines will be used for project 6 (and possibly project 5) and these will be announced when needed.

TENTATIVE POINT DISTRIBUTION

It is not likely that this will change, but circumstances might occur which would make changes necessary, at the discretion of the instructor. Points are allocated and weighted as follows:

- Six Programs : 64% of total points (P1 2%, P2 10%, P3 thru P6 13% each)
- Two Exams : 36% of total points (exam 1 18%, exam 2 18%)
- Total 100%

There are no letter grades for individual projects or exams. The final course letter grade is based on the weighted total of points earned. Final grades are typically based on a straight scale (that is, 90 through 100 is an A-, A or A+; 80 through 89 a B-, B or B+, and so on). Depending on overall class performance (the class averages and class achievement relative to previous semesters) these cutoff points might be adjusted, but class grading will never be "worse than" the straight scale. That is, a 90 will always be *at least* an A-. A curve will be used if appropriate, and if necessary, the mean or median class score may be used as the bottom cutoff for a B-.

Your level of class participation (attending, asking and answering questions) will be taken into account in final course grading, especially if your point total falls on a borderline between letter grades. At the instructor's discretion, your final grade may go up or down by as much as 5% based on class participation.

Please note that doing poorly in a course is not a reason for an incomplete (grade of "I"). Read the FSU Bulletin for a discussion of the policy on incompletes. If you are having problems in the course, come talk to the course or recitation instructor as soon as you possibly can.

COURSE LECTURE TOPIC SEQUENCE and ASSIGNED READINGS

This following table lists course lecture topics and corresponding assigned readings. This information is tentative, may be updated and is subject to change as the semester progresses. Check the course web site and email often for updates.

LEC refers to posted lecture files by Prof. Tyson. Note that lectures include example programs which are posted on Blackboard following each lecture file.

GA refers to course textbook Starting Out with C++ Brief Version: From Control Structures through Objects, by Tony Gaddis and Barret Krupnow, 5th Edition, Addison Wesley Publishers, 2007

TTT refers to course textbook Practical Debugging in C++, by Ann Ford Tyson, Toby Teorey, and Gary Tyson, 2nd Edition, Pearson Custom Publishing, 2007

Other Abbreviations*H. = Course Handout**Prog. = Programming Assignment*

Course Lecture Topics	Lecture Files and Assigned Readings
Course Overview	LEC 01 H. Course Syllabus
Introduction to Computers and Programming	LEC 02 GA Chapter 1
Introduction to C++	H. How to Create a C++ Program Using MS Visual C++ Express 2008 GA Chapter 2
More Introductory C++	LEC 03
Operators and Expressions	LEC 04 H. Submitting Your Program Assignments Electronically Using Blackboard GA Chapter 3 TTT Chapter 1
Input and Output	LEC 05 LEC 06 TTT Chapter 2
Selection	LEC 07 GA Chapter 4 H. Style Guidelines for C++
Design	LEC 08 TTT Chapter 3
Iteration	LEC 09 GA Chapter 5
Iteration Continued	LEC 10 TTT Chapter 5
Testing and Debugging	LEC 11
Functions	LEC 12 GA Chapter 6
Functions Continued	LEC 13
Exam 1 Information	LEC 14
break, continue, more operators, number representation	LEC 15
Simple Data Types	LEC 16
Arrays	LEC 17 GA Chapter 7

	TTT Chapter 4
Strings	LEC 18 GA Chapter 9
Searching and Sorting Arrays	LEC 19
Structures	LEC 20 GA Chapter 10
Multi-Dimensional Arrays	LEC 21
Introduction to Classes	LEC 22 GA Chapter 12
More About Classes	LEC 23 GA Chapter 13
Exam 2 Information	LEC 24
Additional Topics as Time Permits	To Be Announced as needed

ASSIGNMENT DUE DATES and EXAM DATES

Assignment or Exam	Date Due
Programming Assignment 1	Wednesday 1/20
Programming Assignment 2	Wednesday 2/3
Programming Assignment 3	Wednesday 2/17
Exam 1	Thursday 2/25 during lecture period
Programming Assignment 4	Wednesday 3/24
Programming Assignment 5	Wednesday 4/7
Programming Assignment 6	Wednesday 4/21
Exam 2	Tuesday 4/27 10:00 am - 12:00 noon

SOME TIPS FOR DOING WELL IN THIS CLASS

Your goal, most likely, is to receive a good letter grade. Our goal is for everyone in the class to learn as

much about computer science and programming as possible. These goals actually are compatible. You will maximize your grade, and learn an awful lot at the same time, if you

- Attend all lectures and recitation sections; note that many tips and hints about projects are given during lectures and recitations, and if you miss them, it is *your* problem. If you do not attend and participate in class, you will not be able to earn as high a grade as someone who does. Even students who have taken programming classes before lose points on exams and projects due to non-attendance and missed material.
- Read the assigned readings (textbook, handouts, email, web pages)
- Hand in your work on time (even if a program does not work, turn in whatever you have done for possible partial credit)
- Start the projects and come for help as soon as you need it; don't wait until it's too late
- Follow the program style, documentation, and efficiency guidelines carefully
- Work through exercises in the textbook for exam practice
- Read the handout on the class web site entitled "Academic Success Tips" for useful advice for doing well in this and all of your university courses

MANY students who have taken this class previously have said that there is no reason not to do well, because everything you need to do well is provided to you. Your instructors work very hard to make sure that this is the case. If everyone utilized what is given and everyone got an A in this class, your instructors would be very, very happy, and you too!

We also hope that you will **HAVE SOME FUN** while learning a lot from this course!!!

AMERICANS WITH DISABILITIES ACT

Students with disabilities needing academic accommodation should: (1) register with and provide documentation to the Student Disability Resource Center (SDRC); (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. Contact and service information for the SDRC is available on line at <http://www.fsu.edu/~staffair/dean/StudentDisability/>.

Any requests for special exam arrangements due to a registered disability must be brought to the course instructor at least two weeks prior to the exam date, or they will not be considered. In addition students must follow all rules and procedures set forth by the SDRC.

NOTE ON FSU COMPUTER COMPETENCY

From FSU policy: "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."

One programming assignment in this course will be designated for assessment of the FSU computer competency requirement. The assignment designated will be announced to students in the programming assignment write-up.

Last Update: December 31, 2009 A. Ford Tyson
