

OBJECT ORIENTED PROGRAMMING USING C++*

COP 3330
Piyush Kumar

Handout #1, – Course Information

Course Web Site. <http://www.compgeom.com/~piyush/teach/3330/>

Class Mailing List. Announcements for the course, homeworks, reading assignments, programming projects will be available on the course web site (<http://www.compgeom.com/~piyush/teach/3330/>). Make sure you check both the course web site and the blackboard at least once in two-three days throughout the semester.

Instructor. Piyush Kumar.

URL: <http://www.compgeom.com/~piyush>.

Office Hours: Tuesday, 4:30 to 5:30pm.

Alternatively, you may schedule an appointment, either by email or by phone.

Phone: 645-2355

Email: piyush@acm.org

Venue: Office Hours will be held at Love 161 (My Office)

Teaching Assistants. Ryan Walega, Yuhua Zhu.

Office Hours: Tuesday 4:30pm to 5:30pm (Yuhua) , Tuesday, 5:00 to 6:00pm(Ryan Walega).

Alternatively, you may schedule an appointment by email.

Email: rpw05@fsu.edu , zhu@cs.fsu.edu

Venue: Office Hours will be held in Majors Lab.

Lectures. Recitations on Friday, at 9:00am to 09:55am, 10:10am to 11:00am and 11:15am to 12:05pm all in room MCH 0202.

Exams. Finals exams will be held on Dec 14th (Thursday), from 3:00pm to 5:00pm in Love 0101. Midterm will be in class on Oct 12th. (Love 101)

Course rationale:

Writing reusable code is a challenging task in C++. You have already been introduced to the C++ programming language. C++ is a large language and writing good code in the language can be a daunting task. This course will help you learn object oriented programming in modern C++. It will also expose you to the use and design of the C++ standard library, whose wide-spread availability has changed the way C++ code is written. This course involves programming, understanding and creativity. I hope the course is an enjoyable learning experience.

Course Description. Countless programmers today use the object oriented paradigm in C++ to create and manage countless lines of solid code. So far, you have acquired proficiency in basic C++ programming. This course will expose you to the advanced features in C++ as well as help you design software using the object oriented paradigm of programming. This course is the next step towards becoming a good C++ coder for the real world. We plan to cover the following topics in this course (tentative).

*Preliminary version. I will distribute the final version in the first class.

- Basic tools Review: Make, Doxygen, SVN.
- Introduction to basic Coding Standards.
- Review of C++ Basics.
- Using Profiling tools: Valgrind, gprof, memcheck.
- The use of Libraries in C++:
 - IO : Streams, IO Class hierarchy, File IO, Stream manipulators, Stream errors, String Streams.
 - STL : Iterators, vector, string, bitset, algorithms, containers, Function objects, Function adaptors.
- Class Design and Data abstraction.
- Object Oriented Programming : Constructors, Operator Overloading, Friends, Type conversion, Inheritance, Dynamic Vs Static Binding, virtual functions, Abstract Classes, RTTI, Proxy Classes, Static Vs Dynamic Polymorphism.
- Exceptions.
- Recursion, using recursion with template metaprogramming.
- Templates and Generic Programming
- Using Data Structures: Hashing, Stacks, Vectors, Maps, Linked lists, Queues, Priority Queues, Graphs, BFS.

Learning Objectives. The objective of this course is to encourage you to learn and use advanced C++.

Prerequisites. See the prerequisites handout. Starting Fall 2006 prerequisites for COP3330 are COP3014, plus Pre or Co requisite of COP3344. Students that have had COP3502 & CGS3408 for majors from fall 2002 through spring 2006 will have met the prerequisites. Grade of C- or better in prerequisite courses is required. Finally, it is useful to have experience with Makefiles, Doxygen, and Python. (You should be able to code in C++.) Almost all of the homeworks will ask you to write code. The students should be familiar with the following devices and be able to use them to solve programming problems (from CGS 3408 or equivalent course):

- Looping: for, while, and do statements
- Branching: if, else, else if, and switch statements
- Arrays, pointers, and character strings
- Functions: Declaring (prototyping), defining (implementing), and using (calling) functions
- Recursion: direct and indirect recursive function calls
- Classes or structures: class or struct

This material corresponds approximately to Chapters 1-7 in the textbook. The students should also be familiar with the following Unix services (from COP 3502 or equivalent):

- Basic shell commands such as mkdir, cd, ls, and rm
- Use of on-line manual pages through man
- Use of Email services through applications such as Elm or Pine
- Creating and editing text files using Emacs or Vi
- Compiling programs using command line compilers, such as gcc or g++

Textbooks. I will assume that each of you own a copy of *C++ Primer* (Softcover) by S.B.Lippman, J. Lajoie and B.E.Moo. (**Fourth Edition**) ISBN: 0201721481. You should not buy earlier editions of this book. You should also have access to [Gaddis] book:

1. *Tony Gaddis*. Object oriented programming in C++. (2nd edition). Addison-Wesley, 2005.

The text book should be available at FSU bookstore or Bill's on Copeland or at Bill's on Tennessee.

Software Setup: All students are required to have an account on `linprog4.cs.fsu.edu`. The following softwares are installed on this machine for your use:

- gcc 3.4.3 (with g++): Default C++ compiler.
- gcc 4.1.1 (with g++) (`/usr/local/bin/g++4`)
- Intel C++ compiler 9.1 for linux (`/opt/intel/cc/9.1.038/bin/icpc`)
- doxygen
- gdb
- Undo-DB: (`undodb-gdb`) <http://undo-software.com/>
- python 2.3.4
- C++ Boost library (`/usr/include/boost`)
- make / autoconf / automake / scon
- Editors: xemacs, vim
- svn

You should compile and test your code on all the three compilers available on the machine before submission.

Course Policies

1. **Homeworks:** The best way to learn the material is by programming. Unless you learn how to solve problems and program solutions to problems, I *promise* that you will get burned on the exams and thus for your final grade.
2. Your solutions will be submitted using the svn server automatically. You are required to make frequent updates to your svn server as you go ahead with your programming assignment. A project that is checked in at once will be disqualified for evaluation. We want to see how you programmed and what changes you made to your project as time passed by.
3. Your solutions should be well documented in the directory structure that will be shown in the first class (using doxygen). One of the best way to make your solutions clear is to *include pictures* and *examples* in your doxygen documentation.
4. Homework assignments will be due 11:00am on the due date (1.5 hours before *beginning of class*). **Late assignments** *will not be accepted* because the solutions will be available. A script will download all your submissions automatically at that time.
5. It is extremely important that you *start homework assignments early*. The homeworks are very challenging, and if you get behind in your work, you may find it too difficult to catch up. Out of all the graded projects, I will drop the min score before calculating the total homework score towards the final grade. Since I drop the lowest score, missing one homework due to an illness should not be a problem.
6. A short surprise quiz will be given in class (without any announcement). There will be one or two quizzes in the entire semester out of which I will drop the minimum score (if there are two) and count only the top score towards the 5% weight of quizzes.
7. **Grading Criteria:** The grade for COP 3330 will be assigned based on the following approximate percentages.

	Approximate Percentage	Variable
Class Participation (including quizzes)	10%	
Programming Projects	40%	$h \in [0, 50]$
Midterm	20%	$f \in [0, 50]$
Final	30%	

To Pass: $h \geq 25$ and $f \geq 25$.

Final Grades: Your final grades (letter grades) will depend on your $(h + f) \in [0, 100]$ score. I reserve the right to modify these numbers uniformly by 5% each. I reserve the right to de-emphasize the homework grades if

there is evidence of students who copy instead of doing the homework themselves. There is no pre-established scale or curve. I will sort all the $(h + f)$ scores for those who pass and assign letter grades to different non-overlapping intervals (The highest level being A and decreasing thereof). I will at my discretion, use clustering to generate the intervals or the following intervals (Whichever yields you a *better* grade). In prior semesters, grades determined using the class curve yielded approximately 25% students an A (or A-), 25% received a B+/B/B-, 25% received a C+/C/C-, while the remainder received a grade of D+/D/D- or F. The grade scale will be changed (up or down) if the performance of students taking this course differs from those of prior semesters.

Percent	Letter	Percent	Letter	Percent	Letter	Percent	Letter
94-100	A+	84-87	B+	74-77	C+	64-67	D+
90-93	A	80-83	B	70-73	C	60-63	D
88-89	A-	78-79	B-	68-69	C-	57-59	D-
0-56	F						

- Scribing will be worth *approximately* 4 or 5 percentage points of extra credit (This option is only for people who know \LaTeX and XFig or are willing to put the effort to learn it).

Class Participation: Part of the class participation will be scored using one or two surprise quizzes given in class.

- Project Assessment:** Projects will be assessed a grade in two stages. First an objective assessment will be done to test compilation and correctness of the running program. Then a member of the instructional staff will add subjective assessment based on the test results and source code.

- You may earn up to approximately 66 percentage points for a correctly functioning project meeting all requirements. (The exact percentage will depend on the particular assignment.)
- Your remaining project score will be based on the subjective assessment.
- You must understand your project work. If you are asked to explain your work, and if you cannot do so, you may be assigned a grade of zero.

- Missed exam Policy:** A missed exam will be recorded as a grade of zero. We will follow the university rules regarding missed final exams (see http://registrar.fsu.edu/dir_class/fall/exam_schedule.htm), for all the exams, including the final exam.

- Grade of 'I' Policy:** The grade of 'I' will be assigned only under the following exceptional circumstances:

- The final exam is missed with an accepted excuse for the absence. In this case, the final exam must be made up during the first two weeks of the following semester.
- Due to an extended illness or other extraordinary circumstance, with appropriate documentation, the student is unable to participate in class for an extended period. In this case, arrangements must be made to make up the missed portion of the course prior to the end of the next semester.

- Academic Honor Code:** The basic course rule is that you may not give or receive 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. Because a primary goal of the course is to teach professionalism, any academic dishonesty will be viewed as evidence that this goal has not been achieved, and will be grounded for receiving a grade of F (You must read the FSU Academic Honor Code in the Student Handbook and abide by it). Copying/Modifying other people's programs/code will be treated the same as copying in an exam.

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.

- Every student must write his/her own code and homework. Showing your code or homework to members of other teams, giving it to them, or making it accessible to them (e.g., by making the files world-readable) is academic dishonesty.
- You are responsible for ensuring that your code/documentation/results/homeworks are adequately protected and not accessible to others. Change permissions of your working directory to 0700 (`chmod 0700 {directory}`).

- 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.

Once again: There is no excuse for cheating in any circumstances. See me before you even *contemplate* cheating.

13. **Accommodation for Disabilities:** If you have a physical, psychological, medical or learning disability that may impact on your ability to carry out assigned course work, I would urge that you contact the staff in the Student Disability Center and bring a letter to the instructor indicating the need for accommodation. The Student Disability Resource Center will review your concerns and determine, with you, what accommodations are necessary and appropriate. All information and documentation of disability is confidential. They can be contacted at (850) 644-9566.
14. **First Day Attendance Policy:** Official university policy is that any student not attending the first class meeting will be automatically dropped from the class. For distance students, this policy is interpreted as posting to the discussion forum "First Day Attendance" no later than the first day of the semester.
15. **Attendance Policy:** The university requires attendance in all classes, and it is also important to your learning. The attendance record may be provided to deans who request it. If your grade is just a little below the cutoff for a higher grade, your attendance will be one of the factors that we consider, in deciding whether to "bump" you up to the higher grade. Missing three or fewer lectures will be considered good attendance. In rare cases, such as medical needs or jury duty, absences may be excused with appropriate documentation. You should let me know in advance, when possible, and submit the documentation I seek. You should make up for any materials missed due to absences.
16. **Syllabus Change Policy:** The syllabus is guide to the course and subject to change with advanced notice.