

# OBJECT ORIENTED PROGRAMMING USING C++\*

COP 3330  
Piyush Kumar

Handout #1, – Course Information

**Course Web Site.** <http://www.cs.fsu.edu/~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.cs.fsu.edu/~piyush/teach/3330/>) or Canvas. Make sure you check both the course web site and the Canvas at least once in two-three days throughout the semester.

**Instructor.** Piyush Kumar.

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

Office Hours: Tuesday, 4:50pm to 5:50pm.

In case you are unable to see me during my office hours, you should schedule an appointment with one of our TAs during their office hours.

Phone: 645-2355

Email: [piyush@cs.fsu.edu](mailto:piyush@cs.fsu.edu)

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

**Teaching Assistants.**

- Sections: 11, 14  
Ashleigh Davis  
Recitation Timing: (Venue: MCH 202)
  - Section 11 - Monday 1:25 pm to 2:15 pm
  - Section 14 - Monday 2:30 pm to 3:20 pmOffice Hours: Tuesday, Thursday 12:30 - 2pm, Major's Lab  
Email: [asd13@my.fsu.edu](mailto:asd13@my.fsu.edu)
- Sections: 12, 13, 15  
Katie Brodhead  
Recitation Timing: (Venue: MCH 202)
  - Section 12 - Tuesday 5:15 pm to 6:05 pm
  - Section 13 - Wednesday 12:20 pm - 1:10 pm
  - Section 15 - Tuesday 6:45 pm to 7:35 pmOffice Hours: Monday 12:20 pm-1:50 pm, Major's Lab  
Thursday 5:30 pm- 7:00 pm, Major's Lab  
Email: [kb14ac@my.fsu.edu](mailto:kb14ac@my.fsu.edu)

**Lectures.** Tuesday, Thursdays 3:35pm to 4:50pm, in CAR 101. All recitations will be held at MCH 202.

**Exams.**

---

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

	Date	Time	Venue
Final	May 3, 2018	8 pm - 10pm	Classroom
Midterm	Feb 22, 2018	3:35 pm - 4:50 pm	Classroom
2+ Surprise Quizzes	NA	During Class or Recitations	Classroom/MCH 202

### 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).

- Basic tools Review: Make, Doxygen, hg.
- 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.

- A C- or better in COP 3014, or an equivalent introductory programming course.

Students should be able to code in C++ and be familiar with the following:

- 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

This material corresponds approximately to Chapters 1-6 in the textbook (except auto and decl type). The students should also be familiar with the following Unix services:

- Basic shell commands such as mkdir, cd, ls, and rm
- Use of on-line manual pages through man
- Creating and editing text files using Emacs or Vi

- Compiling programs using command line compilers, such as gcc or g++
- Remote login to linprog.cs.fsu.edu with SSH client (you must already have an account)
- Using a web browser (you must create a bitbucket.org account using your @fsu.edu email)

**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. (**Fifth Edition**) ISBN: 0321714113. You should not buy earlier editions of this book.

The text book should be available at FSU bookstore or Bill's on Copeland or at Bill's on Tennessee (or you can order one from Amazon).

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

- gcc 4.8.5 (with g++): Default C++ compiler.
- Clang C++ compiler 3.4.2 for linux (`/bin/clang++`)
- doxygen
- gdb
- python 3.5.2
- C++ Boost library (`/usr/include/boost`)
- make / autoconf / automake / scon
- Editors: vim, xemacs
- hg/git

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

## Course Policies

1. **Projects:** 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. There will be 1 to 2 projects biweekly in this class (Except Finals and midterm week).
2. Projects are to be submitted via Canvas but before submitting your solutions, make sure you commit every significant change to a hg server (e.g. bitbucket.org). This will retain a backup of your project along with its commit history. This will also let us see how you programmed and what changes you made to your project as time passed by.
3. Projects should be well documented in the directory structure that will be shown in the first few classes (using doxygen). Submissions should include a Makefile such that running make will create the required executable. The corresponding project write up will have more information on this.
4. Projects will be due 12:30pm on the due date (3 hours before *beginning of class*). **Late assignments will not be accepted** because the solutions will be available.
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. Short surprise quizzes will be given in class (without any announcement). There will be more than four quizzes in the entire semester out of which I will drop the minimum score and count only the remaining scores towards the weight of quizzes. Quizzes might also happen during recitations.
7. Any attendance grades, quiz grades, or graded work from recitation will count towards your final grades. So, make sure you do not miss the recitations.
8. **ABET/SMALC Assessment:** This course is one of the Computer Science department courses designated for assessing certain student outcomes, required by SMALC/SACS and ABET for accreditation purposes. The specific outcome being measured in this course is:

- Students will be able to construct computer software solutions for simple programming problems

This assessment will be done with 2 different programming assignments in the course, designated as such because they will incorporate multiple aspects of programming skills learned to date. They will be scored in several areas on a scale of values including "Ineffective", "Effective", and "Highly Effective". Each student must achieve an overall score of "Effective" or higher (Earning 70% of available assignment points will count as "Effective") on at least ONE of the two designated assignments. The assignment writeup for each will designate it as a programming assignment used to measure the ABET Outcome Assessment.

9. **Grading Criteria:** The grade for COP 3330 will be assigned based on the following approximate percentages.

	Approximate Percentage	Variable
Attendance	5%	
Programming Projects	30%	
Quizzes	20%	$f \in [0, 55]$
Midterm	20%	$h \in [0, 45]$
Final	25%	

**To Pass:**  $h \geq 23$  and  $f \geq 27$ .

**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).

Percent	Letter	Percent	Letter	Percent	Letter	Percent	Letter
92.00-100	A	82.00-87.99	B	72.00-77.99	C	60.00-61.99	D-
90.00-91.99	A-	80.00-81.99	B-	69.00-71.99	C-	0.00-59.99	F
88.00-89.99	B+	78.00-79.99	C+	62.00-68.99	D		

10. **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 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.

11. **Missed exam Policy:** If you miss an exam with a justified emergency, we will take the average of the other exams/quizzes as the grade of your missed one.

12. **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.

13. **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.

- Plagiarism detection tools, such as Moss (A system for detecting software plagiarism), might be used in this course.

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

14. **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.
15. **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.
16. **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. Total number of lectures and recitations missed should be less than 3 during the semester. Beyond this 1% of your grade will be deducted for each absence (for upto 5 days). 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.
17. **Syllabus Change Policy:** The syllabus is guide to the course and subject to change with advanced notice.