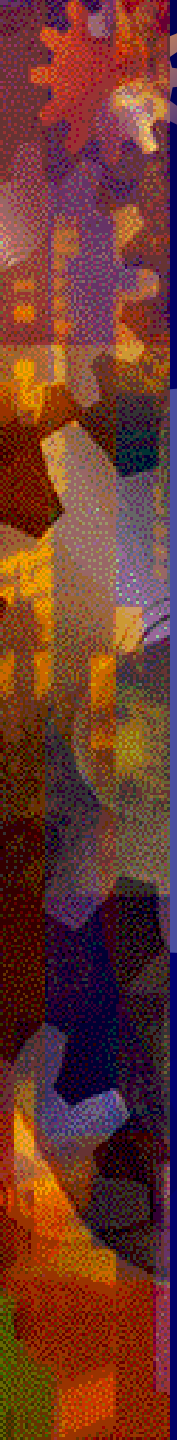

COP3330 Object Oriented Programming in C++

Syllabus

Instructor

Xin Yuan (xyuan@cs.fsu.edu)

Office: 168 LOV

Office hours: T, H 12:45pm – 1:45pm, or by appointment

Class website:

http://www.cs.fsu.edu/~xyuan/cop3330

Teaching Assistant

Md Shafayat Rahman

Office: MCH 106B

Office Hour: Monday 11:30am – 1:00pm

Contact: rahman@cs.fsu.edu

Responsibility: Recitation 1, grading

Teaching Assistant

Zhou Tong

Office: MCH 105F

Office Hour: Friday 9:00am – 11:59am

Contact: tong@cs.fsu.edu

Responsibility: Recitations 2 and 5, grading

Teaching Assistant

Zachary Yannes

Office: Lov 170

Office Hour: Wednesday 2:00am – 3:30am

Contact: yannes@cs.fsu.edu

Responsibility: Recitations 3 and 4, grading

Course Objectives

- Understand object oriented programming and advanced C++ concepts
  - Be able to explain the difference between object oriented programming and procedural programming.
  - Be able to program using more advanced C++ features such as composition of objects, operator overloads, dynamic memory allocation, inheritance and polymorphism, file I/O, exception handling, etc.
  - Be able to build C++ classes using appropriate encapsulation and design principles.
- Improve your problem solving skills
  - Be able to apply object oriented or non-object oriented techniques to solve bigger computing problems (than the ones in COP3014).
- Ultimate goal: to make you a good programmer.

Implication

- To make you a good programmer
  - With proper training, anyone can be a good programmer.
    - This is the course to get the training.
  - Programming skill is not something one processes naturally.
    - It takes practice to develop the skill.
    - To become a good programmer, you have to write a certain amount of code, make the code work, and fix a certain number of bugs.
      - Watching other people making your code work is not going to help your program skills and is a missed opportunity.
      - **Running for help at the first sight of a problem is the biggest obstacle for one to become a good programmer.**
    - No pain, no gain!!!

Prerequisites

- C- in COP 3014
  - Understand procedural programming using C/C++
    - Varaibles and arrays
    - Various control flows
      - Expression and assignment
      - Sequence, conditions, loops, subroutines
    - Basic IO mechanisms.

Course Material

- Lecture notes (posted at the class website)
- Textbooks:
  - Walter Savitch, *Absolute C++*, 5th Edition

Class Grading

- Midterm (25%)
- Final (35%)
  - Covers the whole course
- Programming projects (30%)
- Homework/quizz/bug_fixing_log/ attendance/etc (10%)

Programming projects

- Tentative 8-10 programming projects (30%)
  - Some on problem solving
  - Some on Object oriented programming
  - Some mixed
  - Almost 1 project every week
- Complete project within due date
  - 10% for up to one day.
- Two lowest grade projects will be dropped; others are equally weighted for the project grade.
- Key code segments in the projects will appear in the exam.

Programming projects

- The projects are designed for you to do on your own.
  - You are the only one who is responsible for your own code.
  - Asking for help on your code is a missed opportunity and should only be used as a last resort.
    - If you need to ask anyone to look at your code, ask the TA or myself in person.
    - Asking your friends to look at your code is a form of cheating.
- Two policies to encourage you to make YOUR program work ON YOUR OWN as much as possible.
  - Programs with any compiling error will receive a 0 grade.
    - For an experienced programmer, software design and coding takes about 50% of the development process. The other 50% is in debugging, testing, and making the software work.
    - If your code has compiling errors, you did not spend enough time on the project.

Programming projects

- Second policy: 6 lifelines for each person after the first project.
  - The lifelines can be used for the TA or myself to identify bugs (compiler or run-time) **in your code.**
  - Any question that can be answered without directly looking at your program do not count as a lifeline.
    - If you are able to isolate the problem in one or a few lines of code, you can hand-write the code segment and ask questions without being counted as the lifeline.
  - You can only use your lifelines in person.
    - You should NOT email you code to anyone for bug fixing.
  - Unused lifelines will be converted to extra points for the course at the end of the semester.

Letter grades

- A : 92-100%
- A-: 90-92%
- B+: 88-90%
- B: 82-88%
- B-: 80-82%
- C+: 78-80%
- C: 72-78%
- C-: 70-72%
- D+: 68-70%
- D: 62-68%
- D-: 60-62%
- F: 0-59%

To get C- or above, you must have a C- for the exam and the combined grade. If the exam grade misses the threshold, the Highest letter grade is a D+.

Computer Accounts

- Computer science account
  - Various tools
    - SSH, E-mail, text editor, g++, make
- FSU account
  - Receiving class emails
- Please communicate with the instructor and the TA using a fsu account (cs or garnet). Emails from outside fsu accounts (yahoo, hotmail, gmail, etc) will be ignored.

Tentative schedule

- Week 1: Structures and Classes (Chapter 6)
- Week 2: Constructors and other tools (Chapter 7)
- Week 3: Operator overloading, friends and references (Chapter 8)
- Week 4: Arrays and classes (Chapter 10)
- Week 5: Pointers and dynamic classes (chapter 10),
- Week 6: Midterm
- Week 7: String classes (Chapter 9)
- Week 8: Recursion (Chapter 13)
- Week 9: Inheritance (Chapter 14)
- Week 10: Polymorphism, virtual function (Chapter 15)
- Week 11: Templates (Chapter 16)
- Week 12: Final exam

Academic honor policy

- Read the student handbook
- All violations will be processed by the university
- Step 1 penalty: 0 grade for the particular homework/project/exam AND 1 letter grade downgrade for the final course letter grade (e.g. B->C).
  - Expect to fail the class if caught cheating once.

Academic honor policy

- We will use all our resources to maximize the possibility for catching cheater.
- We will do our best to make sure that those who cheat fail the class at least.
  - There will be a large percentage of exam points directly related to the programming projects.

Your Responsibilities

- Understand lecture and reading materials
- Attend office hours for extra help, as needed
- Uphold academic honesty
- Turn in your assignments on time
- Check class Web page and your email account and regularly

Dos and *Don'ts*

- Do share debugging experiences
- Do share knowledge of tools
- Do acknowledge help from others
- Do acknowledge sources of information from books and web pages

Dos and *Don'ts*

- Don't go for help at the first sight of a problem.
  - Get help in coding as the last resort.
- Don't work on other people's code.

- Don't cheat
- Don't copy code from others
- Don't *paraphrase* code from others either
  - E.g., changing variable names & indentations
- Don't leak your code to any place
  - There is no difference in terms of penalty between copying and being copied.
- All of the above honor code violations will be resolved through the Office of the Dean and the Faculties.
  - Zero for the particular assignment/exam AND one letter grade deduction for the level 1 agreement (first violation).

Course Policies

- Attendance mandatory
- There are no make-up exams for missed exams unless one (1) has a good excuse AND (2) notices the instructor before the exam.
- Students with disabilities
  - Report to Student Disability Resource Center
  - Bring me a letter within the first week of class