

CS 1 - Programming and Problem Solving CS 23021 (Spring 2008) - Sections 003 & 004

Tuesday & Thursday 11:00 am - 12:15 pm
Room 121 MSB, Call No. 12319 & 12320

Instructor:

Dr. L. Gwenn Volkert

Email : volkert@cs.kent.edu

Office: 216 MSB

Phone: (330) 672-9037

Mailbox: 233 MSB

Web: <http://www.cs.kent.edu/~volkert>

Office Hrs: T/Th 12:15 pm - 2:00 pm, and by appointment

Class Web: <http://www.cs.kent.edu/~volkert/cs23021/S08>

Course Description:

This is a programming course that also introduces the object oriented programming paradigm. Object Oriented Programming is the most important and dominant programming approach today. Object Oriented Programming is quite different than functional or procedural programming, and it is difficult to learn on your own. This course is concerned primarily with beginning programmers who have never programmed in the C++ language. The course will focus on programming correctly in C++ by teaching structured and object oriented programming techniques, and proper program design. Students will design, develop, write (translate), compile, execute and debug C++ programs throughout the course. Hands-on programming will be a key part of the course. Please note that this course, CS 23021, is a prerequisite for CS 33001, Data Structures. A grade of C (2.0 honor points) or better in CS 23021 is required to take CS 33001. Note that a grade of C- (1.7 honor points) will NOT meet this requirement. Please see <http://www.cs.kent.edu/programs/ugrad/planner.html> for additional details.

Course Objectives:

- Introduce a disciplined approach to problem solving and algorithm development.
- Program development, including design, coding (translating), debugging, and testing.
- Basic language statements - syntax, semantics, usage of the C++ language
- Functions - syntax, semantics, usage, functional abstraction
- Familiarity with strings and vectors
- Types - syntax, semantics, usage
- Structures
- Introduce the basics of classes (limited)
- Pointers and dynamic memory for single objects (if time permits)
- Correct translation of algorithm into program
- Basic tool usage - editor (emacs or vi), compiler (g++), shell (bash), search tools (grep)

Prerequisites

CS10051 Introduction to Computer Science or CS10061 Intro to Computer programming or permission from the instructor.

Textbook

- Problem Solving With C++, The Object of Programming, 6th edition, Walter Savitch, Addison Wesley, ISBN: 0-321-41269-9, 2006.

Lectures

Students are expected to attend each lecture. I will not take roll, yet attendance and active participation during a lecture will help you learn the material and succeed in class. Please note your attendance will be noticed as I often direct questions to individual students through out the lecture.

Class Participation

20 points are given for participation. You are expected to answer questions I ask in class. The questions usually deal with the material we covered in the previous class. If you do not attend the class I consider that you do not answer questions I ask you. Rather than participate in class you may select to do a harder last project (which will earn you the extra 20 points.) If you select this option you have to *inform me by e-mail within the first two weeks of classes*. Once you choose this option, you cannot go back to class participation option. Even though I provide this alternative, I encourage you to select class participation since I believe this is the best way to learn the material.

Quizzes

There will be approximately 7 quizzes held during the class. The date of the quiz is announced about a week in advance (there will be no surprise quizzes.) A quiz is held during the first 10 minutes of the class. Late students will not be given extra time to complete the quiz. A quiz usually contains 10 multiple-choice questions. Each question is worth 1 point. I will not count your worst score towards your final grade (missing quiz is equivalent to scoring 0.)

Exams

There will be one midterm exam (held during class) and a final exam (held during finals week). All exams are closed book, closed notes, and must be individual work. It is expected that you take each exam at the scheduled time, unless you make *prior* arrangements with me, or have a *documented* illness (in which case I expect you to contact me as soon as possible). You will be tested on the material I covered in class. The textbook alone may not be sufficient for adequate preparation for the exams.

Labs

The labs are designed to augment the lecture material with practical experience using command line development tools and to reinforce programming concepts. Attending and completing the accompanying programming lab is required. **You will not receive credit for the labs that you miss.** A university-approved excuse will allow you to make up a lab.

Programming Projects

There will be approximately 6 programming projects. The programming projects involve reading, modifying and writing C++ code. You will submit your projects electronically. The projects will also be graded electronically. Details on individual programming assignments and the detailed requirements for them will be given when they are assigned. The general program requirements listed on the course website apply to all programs.

You will be provided with an account on departmental undergraduate Unix server. You are, however, free to do your work on any other Unix machine you have access to.

Late Policies

- quizzes no late quizzes accepted, no make-up quizzes;
- exams no late exams, no make-up exams;
- projects late projects accepted. 10% of the grade is subtracted for each day the project is late. For penalty calculation Saturday and Sunday are counted as one day.

Late work will be accepted as stated above. I may waive the late policy conditions only in case of a *documented* illness or some extraordinary circumstance. In either case you have to contact me immediately. With respect to projects, my decision to grant you a waiver is partially influenced by the degree of completion of the work assigned. For example, if the project is assigned for 2 weeks, by 10th day I expect you to complete 65-70% of the work.

In general, you will have adequate time to complete each assignment. However, you should begin working on each assignment early so that you will have plenty of time for debugging which may take significantly longer than the initial code writing. Waiting to start coding until the night before the due date for a project is a bad idea.

Academic Integrity

Student-teacher relationships are built on trust. Students must trust that teachers have made appropriate decisions about the structure and content of the courses they teach, and teachers must trust that the assignments that students turn in are their own. Acts that violate this trust undermine the educational process. Academic dishonesty in any form will be penalized up to assigning grade F in the course.

Cooperation on Homework Assignments and Programming Projects

For both homework assignments and programming projects, I strongly believe that discussion with your peers is an excellent way to learn. If you don't understand something, discussing it with someone who does can be far more productive than beating your head against the wall.

Having advocated discussion, then, I must be clear what is allowed, and what is not. In general, students are allowed to cooperate as follows: you are allowed to discuss with other students the assignment, and general methods for solving the assignment. However, you are not allowed to work with someone else to actually *solve* the assignment, or to *write code* (even pseudo-code) for a program, and you are certainly not allowed to *copy* anyone else's solution; doing any of these things will be considered cheating, and will constitute grounds for failing the course. Note that there is a fine line between discussion and cheating. If you are unsure what is allowed and what isn't, feel free to discuss the distinction with me, but if something feels uncomfortable, it's probably not allowed.

Finally, you should be careful not to give others access to your code. This means that you shouldn't keep your program in a publicly accessible directory, you shouldn't leave your terminal unattended, and you shouldn't forget to pick up your printouts.

Grading

Your final course grade will be calculated as follows:

- | | |
|--|---|
| • quizzes (approximately 7) | 10 points each, worst score dropped |
| • class participation | 20 points |
| • programming projects (approximately 5) | 20 points for the first project
50 points for each subsequent project (250 pts tot.) |
| • midterm exam | 150 points |
| • final exam | 150 points |
| • lab points | 200 points |

TOTAL POINTS: 850 points

The sum of the possible scores on all assignments is considered 100% and your final course grade will be determined as follows – A = 93–100%, B = 83–86.99%, etc. **There will be no curve at the end of the course**, although individual exams may occasionally (although rarely) be curved. Note that this means that **your score will not be rounded up: if you get 69.99% you will get a D+ not a C-**. Thus you should always be able to determine how well you are doing in the course.

You will provide me with a pseudonym by the end of the second week of class. Your grades will be posted on the course's webpage under your pseudonym in a directory that is only accessible with a proper userid and password.

Letter grades will be assigned according to the following percentages. Plus/Minus grades will be given.

93% <= A <=100%
90% <= A- < 93%
87% <= B+ < 90%
83% <= B < 87%
80% <= B- < 83%
77% <= C+ < 80%
73% <= C < 77%
70% <= C- < 73%
67% <= D+ < 70%
60% <= D < 67%
0% <= F < 59%

Students with Disabilities

University policy 3342-3-18 requires that students with disabilities be provided reasonable accommodations to ensure their equal access to course content. If you have a documented disability and require accommodations, please contact the instructor at the beginning of the semester to make arrangements for necessary classroom adjustments. Please note, you must first verify your eligibility for these through Student Disability Services (contact 330-672-3391 or visit www.kent.edu/sds for more information on registration procedures).

Student Expectations

- This is a computer-programming course. Computer-programming is best learned by actually writing lots of computer programs. You will need to spend a lot of time designing, writing and debugging programs.
- Start work on a programming assignment as soon as possible after it is given. This will allow you to discover things that aren't clear to you and ask questions about them. It is hard to write a program quickly at the last minute, it is highly recommended to avoid this situation. Also, unforeseen circumstances often occur.
- Attendance is necessary and expected. It is up to the student to make up any missed material. If a class is missed it is best to get notes from a fellow student (who has taken good notes). This will be the best record of what transpired during the class meeting.
- You have a printer quota of 50 pages. Additional pages may be purchased at 3 cents per page. At least 24 hours must be allowed for the quota to be increased.
- Please turn off any phones, beepers, or other noise-making device before class begins.
- The schedule and procedures for this course are subject to change. Changes will be announced in class and posted on the course website, it is the student's responsibility to learn and adjust to changes.
- If you have any problems, including understanding the material that we cover in class or using the computer, please email me and/or bring your questions/problems to office hours.