

CPS171 Syllabus for Winter 2003 (Introduction to Programming with C++)

Tuesday 07:00PM - 08:55PM TI 126

Thursday 07:00PM - 08:55PM OE 152

Instructor: Victor R. Volkman

Office: none

Work Phone: 734-994-7300

E-Mail: volkman@wccnet.org

Web site: <http://www.HAL9K.com>

Mailing list: <http://groups.yahoo.com/group/cps171/>

Mail boxes: TI 118

Office Hours: Tutoring available in Learning Support Services

Open Lab: TI 108

Student email accounts can be obtained in the open lab.

Credits: 4 hours

Prerequisites: Computer literacy and Math 169 – see below for remarks about CPS 120.

Proficiency in word processing skills is recommended. Students are strongly encouraged to become proficient in keyboarding at the level accomplished in BOS 101A.

CATALOG COURSE DESCRIPTION

This course is an introduction to programming using the C++ language. Students should have basic experience using a computer but no prior programming is required. (Experienced programmers should consider CPS290). Students learn about problem solving strategies, top-down program development and programming style. Topics include sequential, decision and iterative control structures, functions, basic data structures and an introduction to classes. Students write and execute approximately eight C++ programs.

REQUIRED TEXT

C++ Programming: From Problem Analysis to Program design

Author: D.S.Malik

Publisher: Course Technology

Copyright: 2002

Optional: Microsoft Visual C++ .NET - either the standard edition (approx. \$50) or the professional edition (approx. \$100) can be used. You will need this software if you wish to do homework on your home computer. Before buying this software, make sure that you look at the minimum specifications to run it – it takes a lot of memory and hard drive space. If you intend to use the WCC open lab, you do not need to buy this.

Optional Book: Practical Debugging in C++ by Ann R. Ford and Toby J. Teorey. Published by Prentice Hall

Books and software can be obtained at:

The WCC Bookstore in the Student Center Building **or at**

Campus Book and Supply, 1078 N. Huron River Drive, Ypsilanti (485-2369)

COURSE OBJECTIVES

Objective #1. The student will identify the hardware components of a computer and will describe how they act together to form a complete system including the scientific principles on which they are based.

Objective #2. The student will edit, compile, execute and get hard copy of a simple program.

Objective #3. The student will use sound software engineering techniques in C++.

Objective #4. The student will use good documentation, formatting and naming conventions to insure program readability.

Objective #5. The student will identify and describe the ethics and legal issues concerning computer use, including privacy of information and copyright of software.

Objective #6. The student will describe the effects of technology on the individual as well as on society and the environment as a whole.

Objective #7. The student will write a program using the C++ arithmetic operators, input/output methods and appropriate manipulators for formatting.

Objective #8. The student will write a program using appropriate selection statements such as if-else and switch.

Objective #9. The student will write a program using appropriate looping statements such as while, for and do-while.

Objective #10. The student will write a program using functions with parameters passed by value and by reference.

Objective #11 The student will use both one dimensional and multi-dimensional arrays.

Objective #12. The student will describe different sorting and searching algorithms.

Objective #13. The student will use character data and string processing.

Objective #14. The student will use structs in a program.

Objective #15 The student will use classes with data, member functions and constructors

GRADING CRITERIA

Machine Problems will account for 40% of the grade, labs and other exercises will be 10%, and the remaining 50% will be based on the 3 tests and the Final. **YOU MUST GET A SCORE OF 50% OR BETTER ON THE FINAL TO PASS THE COURSE.** Late assignments will be penalized.

Grades will be assigned using the following numeric scale (no rounding up will be done):

A = 92 - 100%, A- = 90 - 91.9%

B+ = 88 - 89.9% B = 82 - 87.9% B- = 80 - 81.9%

C+ = 78 - 79.9% C = 72 - 77.9% C- = 70 - 71.9%

D = 60 - 69.9%

Fluency in programming cannot be attained by simply reading and studying; you must practice the skills by designing, writing and debugging computer programs on your own. You may get help by coming to my office hours or by emailing me – but there may be some days when I am out of town or do not check my email so do not expect immediate help this way. My concept of helping you with machine problems is that I will help you learn to debug programs, but I will not debug them for you. Therefore, when you seek help, you should already have some idea about the nature of the program bugs. A hard copy of your code can also speed up the process for me.

ACADEMIC INTEGRITY

It is always suspect to have machine problem scores significantly higher than your test scores! This usually indicates that you are getting too much help in writing the machine problems. The basic rule is that you may not give or receive assistance for any work that you are submitting as your own.

Some examples of cheating:

1. having someone else write your program (in whole or in part),
2. copying a program someone else wrote (in whole or in part),
3. collaborating with someone else to the extent that the programs are identifiably similar (in whole or in part).

What is not cheating? (a few examples)

1. talking to someone in general about topics and concepts involved
2. asking someone for help with a specific error message from the compiler
3. getting help with the specifics of C++ syntax
4. using information from the program writeup e.g. copying text describing the problem for your comments.

You should also be familiar with the contents of the Student Rights and Responsibilities booklet available at the registration area in the Student Center Building.

Designing programs is often time consuming. Most students find that they spend 10-15 hours per week on this course. If you are having difficulty designing a program please see me for additional help. It is most helpful if I can monitor any trouble areas that seem common to several students so that additional class time might be spent clearing the confusion. You should expect to make several revisions/runs of a program assignment over several days before it is working to your (and my) satisfaction. Do not wait to the last day to start designing your machine problems!

If you decide within the first few weeks of the course that CPS 171 is too difficult for you, please consider changing to CPS 120 Introduction to Computer Science. This course will give you an overview of Computer Science with a small amount of programming and may make it easier for you to do CPS 171 in a future term. There is a 10 week section of CPS 120 that starts on Feb 18, 2003 and you can register for it without paying more tuition if you use the magic words "level change" on your drop and add cards when you adjust your registration.

Class attendance is taken during each class period. It is expected that you will come prepared to contribute to the classroom activities. We will sometimes do group activities, such as design programs, together in class and active participation by each individual helps make the class more interesting for each person. If you must miss a test for any reason, you **must** notify me in advance, otherwise no makeup will be available

Please feel free to inform me of any special circumstance or need that you have so that appropriate steps can be taken.