CS 195   **INTRODUCTION TO COMPUTER SCIENCE**   **FALL 2001**

Syllabus & Course Policies

INSTRUCTOR: Dr. Robert (Bobby) England   EMAIL: englandr@rhodes.edu
OFFICE: OH 419   PHONE: 843-3725

OFFICE HOURS: Monday – Friday, 10:15am - 11:15am
          Available lots of other times --- please make an appointment!

TEXT: *Computer Science: A Structured Approach Using C++*, by Forouzan and Gilberg.
          Supplemental material may be distributed in class as needed.

MAIN TOPICS: [not necessarily in this order:]
          Computer program design and problem solving.
          Using a C++ software development environment.
          Intro to fundamental programming constructs and practice (using C++):
                    1.   Data types
                    2.   Decisions
                    3.   Loops
                    4.   Functions / methods
                    5.   Arrays
                    6.   Classes / object-oriented programming

OBJECTIVES:

The main objective of this course is to introduce you to the basic techniques and foundational concepts of computer programming and computer science. Algorithmic design and problem-solving strategies will be emphasized as well as basic C++ syntax and data structures.

PREREQUISITE:
          None. (Yea!)

COREQUISITE:
          CS 195L (Boo.)

DROPS:
          The last day to withdraw from the class is Friday, 26 Oct 2001.

GRADING:
          Program assignments: 40%
          2 in-class tests: 40%
          Final exam: 20%   [Fri, 7 Dec, 5:30-8:00pm]

ATTENDANCE:

Students are responsible for all material assigned or covered in class. The instructor's own lecture notes will not be made available for copying or review, so be sure to get notes from a colleague in the class if you should (unavoidably) miss a class meeting.

Regular class attendance is not formally required but it is strongly encouraged. As a bonus to those who attend regularly, each student who misses no more than 2.0 class meetings will be allowed to omit selected questions on the final exam worth a total value of approximately 15 points out of 100. These 2.0 allowed absences are to cover emergencies and do not have to be explicitly excused --- no excuses for other absences will be accepted with regard to this bonus. Each tardy (arrival after the start of class) counts as 0.5 absence. After class, you should alert me to your presence if you were tardy; otherwise, you will probably be counted absent. No makeup tests will be given unless arrangements are made in advance.

PROGRAM ASSIGNMENTS:

All programs assigned in this course must be written in C++. Each program assignment will each be awarded a letter grade A through X:

A: (100 pts)
‘A’ programs are carefully designed, efficiently implemented, well documented, and produce clearly formatted, correct output.

A- : (94 pts)
This is an ‘A’ program with one or two of the minor (?) problems described for grade ‘B’.

B: (88 pts)
A ‘B’ program typically could easily have been an ‘A’ program, but it may have minor/careless problems such as poor, inadequate, or incomplete documentation; several literal values where symbolic constants would have been appropriate; wrong file names (these will be specified per program assignment); incomplete hard copies; sloppy source code format; minor efficiency problems; minor (?) memory leaks; etc. (This is not an exhaustive list.) You would be wise to consider ‘B’ the default grade for a working program --- this might encourage you to review and polish your first working draft of an assignment to produce a more professional quality final version of your program.

C: (75 pts)
A ‘C’ program has more serious problems: incorrect output for important special cases (the "empty" case, the "maxed-out" case, etc.), failure to carefully follow design and implementation requirements spelled out in the assignment description, very poor or inefficient design or implementation, near complete absence of documentation, etc.

D: (60 pts)
A ‘D’ program compiles, links, and runs, but it produces clearly incorrect output for typical cases. Or, it may deviate greatly from the design or implementation requirements stated in the assignment description.

F: (35 pts)
Typically, an ‘F’ program produces no correct output, or it may not even compile. It may "look like a program" when printed as a hard copy, but there remains much work to be done for it to be a correct, working program. Still, as a last resort, an ‘F’ program is better than no program turned in at all.

X: (0 pts)
A grade of ‘X’ will be recorded for each program not turned in. Each ‘X’ has the additional

side effect of lowering the final grade for the course by one letter.

A fully documented sample program that you can use as a model for source code format will be distributed with or before the first programming assignment. The first line of each program source code file submitted for credit must be a comment that states the name of the source code file. Each student is responsible for keeping a back-up copy on disk of all source code turned in for an assignment. Failure to do so could result in loss of credit for an assignment.

Programming assignments must be turned in on the due date to receive full credit. Programs will be accepted late, but with a penalty of one letter grade per day.

Recommendation: *keep this list* of policies handy as a partial checklist of requirements to review before turning in each completed assignment!

TESTS:

Both of the regular tests and the exam will be closed book, closed notes. Typical test format is a list of multiple choice questions, one code writing problem, and one code trace problem, though there may be slight variations to this format.

All programs and tests must be the student's *own* work. Copying all or part of an assignment, or downloading code from the Internet and submitting it as your own, or having someone else write code for your assignment, or having someone else debug your assignment, or *allowing* someone else to copy from you, or coding or debugging someone else's assignment --- these are all included in the definition of reportable Honor Code violations for this course. If you have any doubts about whether or not a program development practice on an assignment is legal, clear it with the instructor before proceeding.

The instructor reserves the right to alter this syllabus as necessary.