

Winter 2005

CEG 220-01: Introduction to C Programming for Engineers

Robert Helt

Wright State University - Main Campus

Follow this and additional works at: https://corescholar.libraries.wright.edu/cecs_syllabi



Part of the [Computer Engineering Commons](#), and the [Computer Sciences Commons](#)

Repository Citation

Helt, R. (2005). CEG 220-01: Introduction to C Programming for Engineers. .
https://corescholar.libraries.wright.edu/cecs_syllabi/1234

This Syllabus is brought to you for free and open access by the College of Engineering & Computer Science at CORE Scholar. It has been accepted for inclusion in Computer Science & Engineering Syllabi by an authorized administrator of CORE Scholar. For more information, please contact corescholar@www.libraries.wright.edu, library-corescholar@wright.edu.

Syllabus
CEG 220 Introduction to C Programming for Engineers
Section 1 - Winter 2006

T Th 6:05 – 7:45 p.m. in Russ Engineering Center Room RC152A

Description: This course provides a general introduction to computers as a problem-solving tool using the C programming language. Emphasis is on algorithms and techniques useful to engineers. Topics include data representation, debugging, and program verification. 4 credit hours. Prerequisite: MTH 229 (Calculus I) or EGR 101 (Engineering Mathematics).

Instructor: Robert Helt, Russ Engineering Center Room RC160 (Student Lounge Area) E-mail: rhelt@wright.edu Office hours: 5:00 to 6:00 p.m. M-W or immediately following class. Other hours arranged and confirmed by email.

Textbooks:

C Programming: A modern Approach, K. N. King, W. W. Norton and Company, 1996.

Software: Dev-Cpp 4.9 C Compilers available on the lab computers as well as a CD from the instructor. Alternate C compilers are Visual Studio.net, Visual Studio 6.0, and the UNIX GNU C compiler. Other C compilers must be approved by the instructor.

Grading:

Programming Projects: Programming projects are assigned on Tuesday. Each project is due the following Tuesday, one week from when it was assigned. Each programming assignment is worth 10 points. Programming assignments will be graded as *Satisfactory* or *Unsatisfactory!* For a project to be satisfactory, 1) the source code file must contain the required header information, 2) the source code must meet style and documentation guidelines, 3) the program must compile and run without warnings or errors, 4) and the program must achieve all the objectives of the project. If a project is graded as satisfactory, six to ten points will be awarded, depending on how well the project meets the specifications and grading criteria. Each time a programming project is turned in and graded as unsatisfactory, one point will be lost. Unsatisfactory projects must be resubmitted not later than **midnight** two days after it is returned. If any portion of the assignment is turned in late, one-half point will be deducted for each day it is late. Any unsatisfactory assignment that is finally graded satisfactory will not be worth less than 6 points. The programming projects will comprise 25% of the grade. **All six projects must be turned in by the last day of classes for the quarter and completed with a grade of Satisfactory to pass the course!**

Examinations: There will be two mid-course **50-minute Exams** that will comprise 40% of the final grade. There also will be a **Final Exam** at the end of the course worth 35% of the final grade. All exams will be closed book, closed notes, no electronic devices in view.

Grades: **A:** 100-90, **B:** 89-80, **C:** 79-70, **D:** 69-60, **F:** 59-0, **X:** all programming projects not completed with a grade of *Satisfactory*.

Policy: All **programming projects** will be turned in electronically by email and are due before **midnight** on the dates specified. See the **CEG220 Section 1 Home Page** for detailed requirements and instructions for turning in programming projects. No make-up exams will be given unless a serious illness or a bona fide emergency can be verified. Exceptions to the policy for turning in work late and giving make-up exams may be made in unusual circumstances when the student provides documentation in writing from an appropriate source. **All work must be your own; copying or sharing program code will constitute a breach of academic integrity and could result in failure of the course for all individuals involved.** Sharing

programming ideas and general computer skills with others outside of class is encouraged, especially through the **CEG220 News Group @ "wright.ceg.220"**.

Course Materials: All lecture slides, project information, handouts, and sample programs are available at the **CEG220 Section 1 Home Page** at <http://www.cs.wright.edu/~rhelt/ceg220/ceg220.html>.

Schedule: Lecture dates for topics may vary slightly. Project due dates and exam dates are firm.

Week	Topics - Readings in the King Text - Project Dues Dates - Exams
1	C Fundamentals and Formatted Input/Output - Chapters 1, 2, and 3
2	Expressions, Math Functions, Basic Types, Character Functions – Chapters 4, 23.3, 7.1-7.5, and 23.4
3	Selection Statements, Loops, and Declarations - Chapters 5, 6, and 18 - Project No. 1 due 17 Jan
4	File Operations - Chapters 22 - Project No. 2 due 24 Jan - Exam#1 on 26 Jan
5	Functions - Chapter 9
6	Functions, Program Organization, and Arrays - Chapters 9, 10, and 8 - Project No. 3 due 7 Feb
7	Arrays and Strings - Chapters 8, 13, 23.5 - Project No. 4 due 14 Feb - Exam#2 on 16 Feb
8	Recursion and Pointers - Chapters 9.6, 11, and 12
9	Structures – Chapter 16 - Project No. 5 due 28 Feb
10	Structures - Chapter 16 and Course Review - Project No. 6 due 7 Mar
	Final Exam – Thursday, 16 Mar, 8:00 – 10:00 p.m. in Room RC152A