

Fall 2005

# CEG 220: Introduction to C Programming for Engineers I

Robert Helt

*Wright State University - Main Campus*

Follow this and additional works at: [https://corescholar.libraries.wright.edu/cecs\\_syllabi](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: Introduction to C Programming for Engineers I. .  
[https://corescholar.libraries.wright.edu/cecs\\_syllabi/40](https://corescholar.libraries.wright.edu/cecs_syllabi/40)

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](mailto:corescholar@www.libraries.wright.edu), [library-corescholar@wright.edu](mailto:library-corescholar@wright.edu).

# Syllabus

## CEG 220 Introduction to C Programming for Engineers

### Section 2 - Fall 2005

MW 6:05 – 7:20 p.m. in Russ Engineering Center Room RC148

**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](mailto:rhelt@wright.edu) Office hours: 4:30 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:** LLC-win32 or Devcpp 4.9 C Compilers available on CD from the instructor. Alternate C compiler is 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 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 2 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 2 Home Page** at <http://www.cs.wright.edu/~rhelt/ceg220/ceg220.html>.

**Schedule of Events:**

<b>Week</b>	<b>Topics - Readings in the King Text - Project Dues Dates - Exams</b>
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 - <b>Project No. 1 due 20 Sep</b>
4	File Operations - Chapters 22 - <b>Project No. 2 due 27 Sept - Exam#1 on 29 Sep</b>
5	Functions - Chapter 9
6	Functions, Program Organization, and Arrays - Chapters 9, 10, and 8 - <b>Project No. 3 due 11 Oct</b>
7	Arrays and Strings - Chapters 8, 13, 23.5 - <b>Project No. 4 due 18 Oct - Exam#2 on 20 Oct</b>
8	Recursion and Pointers - Chapters 9.6, 11, and 12
9	Structures – Chapter 16 - <b>Project No. 5 due 1 Nov</b>
10	Structures - Chapter 16 and Course Review - <b>Project No. 6 due 8 Nov</b>
	<b>Final Exam – Wednesday, 16 Nov, 8:00 – 10:00 p.m. in RC148</b>

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