

Wright State University

CORE Scholar

Computer Science & Engineering Syllabi

College of Engineering & Computer Science

Fall 2008

CEG 221: Advanced 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. (2008). CEG 221: Advanced C Programming for Engineers. .
https://corescholar.libraries.wright.edu/cecs_syllabi/102

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 library-corescholar@wright.edu.

Syllabus

CEG 221 Advanced C Programming for Engineers

Section 1 – Fall 2008

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

Description: This course introduces advanced constructs, algorithms, and data structures in the C programming language. Emphasis is on problem solving and techniques useful to engineers. Topics include functions, array, pointers, structures as well as sorting algorithms, linked lists, complex numbers, stacks, queues, hash tables, and binary trees. 4 credit hours. Prerequisite: CEG220 (Introduction to C Programming for Engineers).

Instructor: Robert Helt, Russ Engineering Center Room RC160 (Student Lounge Area) E-mail: rhelt@wright.edu Office Hours: 4:30 to 6:00 p.m. M and W by appointment or immediately following class. For other hours, request an appointment via email.

Textbooks:

C: The Complete Reference, Fourth Edition, Herbert Schildt, Osborne/McGraw-Hill, 2000.

Software: The Dev-Cpp 4.9 C Compiler is installed on the lab computers, and the software is available on the CEG221 Home Page to download for home computers. 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 Monday. Each project is due the following Monday, one week from when it was assigned. Programming assignments will first 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 and produce the correct output, and 4) the project report must cover all the required areas. If a project is graded as *Satisfactory*, up 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 *Unsatisfactory*, two points will be deducted from the possible score. An *Unsatisfactory* project must be resubmitted within two days after it is returned to avoid further loss of points due to late penalties. If a project is turned in late or missing required files, one point will be deducted for each day it is late or incomplete. The programming projects will comprise 25% of the grade.

Examinations: There will be three **One-Hour Exams** that will comprise 75% 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:** Course requirements not met.

Policy: All **programming projects** will be turned in electronically by email and are due before **midnight** on the dates specified. See the **CEG221 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, excused absence, 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 course failure for all individuals involved.** Sharing programming concepts, best practices, and general computer skills with others outside of class is encouraged, especially through the Course Home Page on WINGS at <http://luminis1.wright.edu/cp/home/loginf>. See the handout for accessing the CEG221 Course Home Page on WINGS.

Schedule: Topics and lecture dates may vary. Programming project due dates and exam dates are firm.

Week	Topics - Readings in Schildt Text - Project Dues Dates - Exams
1	Review of C Including Control Statements, Data Files, - Chapters 1, 2 (to p. 48), 3, 8-10, 13-14 (to p.368)
2	User-defined Functions, Array, and Pointers, an strings - Chapters 6, 4-5, 17, 14 (pp. 368-382)
3	Structures, Unions, Enumerations, and typedef Statement - Chapter 7 - Project No. 1 due on 22 Sep
4	Sorting Algorithms - Chapter 21 - Project No. 2 due on 29 Sep - Exam No. 1 on 2 Oct
5	Sorting Algorithms and Complex Numbers - Chapters 21, 11
6	Linked Lists, Stacks, and Queues - Chapter 22 - Project No. 3 due on 13 Oct
7	Double and Circular Linked Lists - Chapter 22 - Project No. 4 due on 20 - Exam No. 2 on 23 Oct
8	Bitwise Operations, Searches, and Hash Tables – Chapters 2 (pp. 48 – 53), 23
9	Hash Tables - Chapter 23 and Binary Trees – Chapters 22, 23 – Project No. 5 due on 3 Nov
10	Binary Trees - Chapter 22, 23, and intro to C++ – Project No. 6 due on 10 Nov - Exam No. 3 on 13 Nov