

COURSE SYLLABUS FOR ECS 30-A — SPRING QUARTER 2015
“INTRODUCTION TO PROGRAMMING AND PROBLEM SOLVING”

Instructor: Dr. Bernd Hamann, <http://graphics.cs.ucdavis.edu/~hamann/>
Classes: Tuesday/Thursday, 1:40pm–3:00pm, 1100 Social Sciences
Office hours: Thursday, 10:45am–11:30am, 3035 Kemper Hall
(questions regarding lecture material only)
Teaching assistants: Divya Banesh (dbanesh@ucdavis.edu), Alexander Couvrette (ajcouvrette@ucdavis.edu),
Siddhika Cowlagi (sncowlagi@ucdavis.edu), Lin Zheng (lzheng1984@gmail.com),
?? (??@ucdavis.edu), ?? (??@ucdavis.edu)
Discussion sections: 30-A01, Monday, 9:00am–9:50am, 205 Olson
30-A02, Monday, 3:10pm–4:00pm, 1150 Hart
30-A03, Thursday, 9:00am–9:50am, 1 Wellman
30-A04, Monday, 12:10pm–1:00pm, 158 Olson
30-A05, Friday, 12:10pm–1:00pm, 1 Wellman
30-A06, Tuesday, 4:40pm–5:30pm, 223 Olson
TA office hours: Announced in discussion sections and on UC Davis Smartsite
(questions regarding lecture material, written homework, programming
projects, C programming language, UNIX operating system, exams, grade)
Course materials: Available on UC Davis Smartsite, <https://smartsite.ucdavis.edu>
Questions/answers: Posted and updated on UC Davis Smartsite, <https://smartsite.ucdavis.edu>
Prerequisites: Math 16A or 21A (may be taken concurrently); experience with
basic programming concepts (variables, loops, conditional statements)

Required books

- Eric S. Roberts, “The Art and Science of C,” Addison-Wesley, 1995, ISBN 0-201-54322-2
- Graham Glass and King Ables, “UNIX for Programmers and Users,” third edition,
Pearson Education/Pearson Prentice Hall, 2003, ISBN 0-13-046553-4

Objectives

Upon completion of this course, students should be competent programmers who can solve problems of reasonable size on a computer using a high-level programming language and basic operating system tools. Students will use the C programming language and the UNIX operating system.

Assignments and exams

- Three (3) written homework assignments and five (5) programming projects
- Midterm: 5/14/2015, 1:40pm–3pm, final: 6/10/2015, 3:30pm–5:30am; 1100 Soc Sci; open-book

Grading criteria

- A: $\geq 90\%$; B: $\geq 80\%$; C: $\geq 70\%$; D: $\geq 60\%$; F: $< 60\%$
- The programming projects contribute 35% to the overall grade.
- The written assignments contribute 15% to the overall grade.
- The midterm contributes 20% and the final exam 30% to the overall grade.

Due to the limited resources that are available to this class for grading written homework assignments and programming projects, it might be unavoidable that only a subset of the work turned in by students can be graded. In the case that it is not possible to grade all parts of a specific written homework assignment or programming project, an instructor-selected subset of a written homework assignment or programming project will be graded. The selected subset would be the same for all students. In this case, the grade of a particular written homework assignment or programming project, a percentage between 0% and 100%, might sometimes be based only on the correctness of the selected subset of the material turned in by students.

A programming project will be graded based on (i) program design/style; (ii) correctness; and (iii) documentation/comments. These criteria carry the same weight. **Written assignments will not be accepted after 4:30pm on the due date. Programming projects can be late by at most four (4) days. For each late day, ten (10) percent of the sum of points will be deducted.** This policy will be applied to late programming projects: (i) There are five (5) programming projects. (ii) Each programming project consists, in general, of multiple individual programs/modules, called, for example, (a), (b), (c), etc. (iii) A student **must turn in** all individual programs/modules that are part of a specific programming project and that he/she wishes to have graded **on the same day**, and within the period of maximally allowed number of late days, i.e., four (4) days; programs/modules that are turned in later (for the same specific programming project) — either additional or revised programs/modules — will not be considered for grading. The teaching assistants will let you know in the discussion sections how to submit your written homework assignments and programming projects.

Academic honesty

Students may and are encouraged to discuss the course material with each other. They may also discuss the specifications of written homework assignments and programming projects. **Collaborations must stop prior to writing down any material that is to be submitted as one's own work. Submitted work must not be copied in any way. Each student must write submitted work independently. Failure to do so represents academic misconduct.** See *UC Davis Code of Academic Conduct*, available at <http://sja.ucdavis.edu/cac.html>.

Tentative course outline

Week 1:	Ch. 1: Introduction (pp. 1–16), Ch. 2: Learning by Example (pp. 21–51)
Week 2:	Ch. 3: Problem Solving (pp. 59–89), Ch. 4: Statement Forms (pp. 99–130)
Week 3:	Ch. 5: Functions (pp. 137–176), Ch. 6: Algorithms (pp. 185–212)
Week 4:	Ch. 7: Libraries and Interfaces: A Simple Graphics Library (pp. 221–249)
Week 5:	Ch. 8: Random Number Lib. (pp. 259–290), Ch. 9: Strings/Chars. (pp. 301–331)
Week 6:	Ch. 10: Mod. Development (pp. 339–365), Ch. 11: Arrays (pp. 375–413)
Week 7:	Ch. 12: Searching and Sorting (pp. 425–446), Ch. 13: Pointers (pp. 453–482)
Week 8:	Ch. 14: [Strings Revisited (pp. 491–516)], Ch. 15: Files (pp. 523–547)
Week 9:	Ch. 16: Records (pp. 557–589), Ch. 17: Looking Ahead (pp. 601–637)
Week 10:	Ch. 17: Looking Ahead (pp. 601–637), continued

Assigned written homework and programming projects

The written homework assignments and programming projects for this course are taken from the “Review Questions” and “Programming Exercises” sections in the textbook by Eric S. Roberts.

Written assignment #1:	Ch. 2: 22; Ch. 3: 12; Ch. 4: 21; Ch. 5: 5, 15; Ch. 6: 6 (due: 4-22-2015)
Written assignment #2:	Ch. 7: 14, 15; Ch. 8: 8, 14; Ch. 9: 5, 8, 17; Ch. 10: 4, 10; Ch. 11: 4, 20 (due: 5-6-2015)
Written assignment #3:	Ch. 12: 4, 7; Ch. 13: 6, 11; Ch. 15: 1, 12; Ch. 16: 8, 14, Ch. 17: 1, 7, 13 (due: 6-5-2015)
Programming project #1:	Ch. 2: 1, 3, 4; Ch. 3: 5, 9 (due: 4-16-2015)
Programming project #2:	Ch. 4: 4, 8; Ch. 5: 2, 6, 12; Ch. 6: 2, 7 (due: 4-29-2015)
Programming project #3:	Ch. 7: 2, 7; Ch. 8: 8; Ch. 9: 8; Ch. 10: 10 (due: 5-13-2015)
Programming project #4:	Ch. 11: 9; Ch. 12: 12; Ch. 13: 5; (due: 5-27-2015)
Programming project #5:	Ch. 15: 9; Ch. 16: 7; Ch. 17: 1, 4 (due: 6-5-2015)