

# Syllabus

## Course Description

### EECS 22L Software Engineering Project in C Language (3)

Hands-on experience with the ANSI-C programming language. Medium-sized programming projects, teamwork. Software specification, documentation, implementation, testing. Definition of data structures and application programming interface. Creation of program modules, linking with external libraries. Rule-based compilation, version control.

Prerequisites: EECS 22 (Design Units: 3)

---

## Course Outcome/Performance Criteria

Course objectives represent what the course strives to accomplish.

Course outcomes represent what will be measured to determine if the course met its objectives.

Course outcomes relate to a subset of the program outcomes.

### Course Outcomes:

- Students will be able to specify and document a software program for a contemporary application (EAC c, e, g, j)
- Students will be able to design and implement dynamic data structures using user-defined data types (EAC e, k)
- Students will be able to implement, test, and debug a software program (EAC b, k)
- Students will be able to utilize software engineering tools such as version control and scripting (EAC k)
- Students will be able to write original source code to solve an engineering problem (EAC c, e, j, k)
- Students will be able to collaborate and communicate effectively in a team (EAC d, g)

### Related Program Outcomes:

- (EAC b) an ability to design and conduct experiments, as well as to analyze and interpret data
  - (EAC c) an ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability
  - (EAC d) an ability to function on multidisciplinary teams
  - (EAC e) an ability to identify, formulate, and solve engineering problems
  - (EAC j) a knowledge of contemporary issues
  - (EAC k) an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice
- 

## Course Resources

### • Course web site

online at

<https://canvas.eee.uci.edu/courses/14040>

(<https://canvas.eee.uci.edu/courses/14040/>).

### • Reference book:

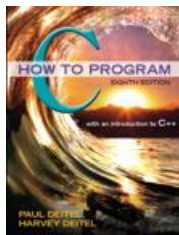
B. W. Kernighan, D. M. Ritchie:  
The C Programming Language,  
Second Edition,  
Prentice Hall, 1988.  
ISBN 0-13-110362-8



### • Supplemental text book:

Paul Deitel, Harvey Deitel:  
C: How to Program,

Eighth Edition,  
 Prentice Hall, 2013.  
 ISBN-10: 0133976890  
 ISBN-13: 9780133976892



## Course Contents

| Week | Lecture topics                                    | Project tasks |                                     |
|------|---|---------------|-------------------------------------|
| 1    | Introduction to software engineering              | Project 1     | Application specification           |
| 2    | Software architecture, design flow, documentation |               | Software architecture specification |
| 3    | Introduction to version control, GUI programming  |               | Documentation, implementation       |
| 4    | Software development, testing, documentation      |               | Implementation, testing, debugging  |
| 5    | Software packaging, installation, deployment      |               | Delivery, installation, deployment  |
| 6    | Project planning, organization, maintenance       | Project 2     | Application specification           |
| 7    | Data structure and API design                     |               | Software architecture specification |
| 8    | System programming, shell scripting, Linux tools  |               | Documentation, implementation       |
| 9    | Outlook to object-oriented programming in C++     |               | Implementation, testing, debugging  |
| 10   | Course wrap up                                    |               | Delivery, installation, deployment  |

- Note that contents are *tentative* and may change.
- Please see the [Schedule \(https://canvas.eee.uci.edu/courses/14040/pages/schedule\)](https://canvas.eee.uci.edu/courses/14040/pages/schedule) page for up-to-date scheduling information.

## Course Policies

### • Attendance Policy:

The class meets for 1 hour of lecture, 1 hour of discussion, and 3 hours of laboratory (on average each week) for 10 weeks. Attendance at lecture, discussion, and laboratory sections is **mandatory** (sign-in sheets). It is each student's responsibility to make up for any missed instruction and any project contributions to the team. Make-up projects and/or exams will only be arranged for absence due to medical (or similar) reasons. Proper documentation is required.

### • Laboratory Policy:

Discussions and labs assigned to this course are listed in the Schedule of Classes (see course codes [18020-18034 \(http://websoc.reg.uci.edu/perl/WebSoc?YearTerm=2017-0&CourseCodes=18020-18034\)](http://websoc.reg.uci.edu/perl/WebSoc?YearTerm=2017-0&CourseCodes=18020-18034)).

If open and available, students may use the laboratory rooms at other times as well. However, those students enrolled in a particular lab section have precedence in using the labs.

- No food or drinks are permitted in the lab at any time.
- Keep the lab clean and neat.
- No littering or abusing lab equipment.
- Shoes and shirts required at all times.

### • Computing server:

Linux is the standard computing platform used for this course. A central computer infrastructure is provided and students may remote login to the server at any time (24/7). All processes run on the EECS Linux servers `crystalcove.eecs.uci.edu` and `zuma.eecs.uci.edu`. While other platforms may be used to develop and test the programs, all course work will be checked

and graded on one of the servers.

SSH is needed to login to the server. Please refer to the [Resources](https://canvas.eee.uci.edu/courses/14040/pages/resources) (<https://canvas.eee.uci.edu/courses/14040/pages/resources>) page for a list of suitable SSH clients.

- **Projects:**

Course projects are listed on the [Projects](https://canvas.eee.uci.edu/courses/14040/pages/projects) (<https://canvas.eee.uci.edu/courses/14040/pages/projects>) course web page.

Projects will be assigned to student teams (see below) and include weekly deliverables.

Deliverables are generally due on **Monday at 12pm (noon)**, unless indicated otherwise on the project web page.

**The deadline for deliverable submission is hard.** Deliverables turned in after the due date/time will not be graded and will receive no credit. The clock on the server will be used as the reference clock.

All deliverables are to be submitted electronically by a designated team member while being logged into the team account on the server. Detailed submission instructions will be provided with each project description.

- **Teams:**

The programming projects in this course will be performed by student teams. Depending on course enrollment, teams of 5 to 10 students will be formed at the beginning of each project. Team work is an essential aspect of this class and every student needs to contribute to the team effort. While tasks may be assigned in a team to individual members, all members eventually share the responsibility for project deliverables. A team account will be provided on the server for each team to share data among the team members. Since teams will compete in the projects, sharing of data across teams is not permitted.

- **Exams:**

The course includes one midterm and one final examination, scheduled as listed on the [Schedule](https://canvas.eee.uci.edu/courses/14040/pages/schedule) (<https://canvas.eee.uci.edu/courses/14040/pages/schedule>) page. Exams will be short oral presentations of individual students at the computer terminal. Students need to explain their original contribution to the team, their original deliverable (source code and/or documentation), and answer questions. The individual contribution may be additionally assessed by TA feedback and/or peer review. The course instructor will assign an individual score for the exams after all exams are completed. An official photo ID is required for inspection.

Final examinations are administered during examination week at the time and place announced on the course schedule page.

- **Grading Policy:**

The final grade for the course will be calculated as a weighted sum of the points scored in participation, team deliverables, midterm and final examination, as follows:

- 5% Participation (individual effort)
- 15% Midterm examination (individual effort)
- 30% Final examination (individual effort)
- 50% Programming projects (team effort)

Extra credit may be earned with projects if indicated in the project description.

- **Academic Honesty:**

Academic honesty is a requirement for passing this class. Any student who compromises the academic integrity of this course is subject to a failing grade. **The work you contribute to your team must be your own.** Academic dishonesty includes, but is not limited to copying answers from another student, allowing another student to copy your answers, communicating exam answers to other students during an exam, attempting to use notes or other aids during an exam, or tampering with an exam after it has been corrected and then returning it for more credit. If you do so, you will be in violation of the UCI Policies on Academic Honesty (see [Academic Integrity \(https://aisc.uci.edu/\)](https://aisc.uci.edu/)). It is your responsibility to read and understand these policies. Note that any instance of academic dishonesty will be reported to the Academic Integrity Administrative Office for disciplinary action and is cause for a failing grade in the course.

While the projects will be graded based on team effort, exams are graded individually (see above). Thus, working together

in the team is necessary (!) for the projects, but not allowed during the examinations.

**Do not copy code across teams!**