# Introduction to C Language Programming, EL ENG X24

2 Semester Credits in EECS

## Course Description

This course provides an introduction to the C Programming Language. You will learn through a combination of reading, doing written exercises, and doing programming exercises.

## Prerequisites

While no previous programming experience is required, you should have access to a computer hosting a C/C++ compiler or access to the Internet to allow you to download and install a C/C++ compiler. Familiarity with fundamental programming language concepts such as variables, data types, and compiling source code is helpful.

## Learning Outcomes

After successfully completing this course, you will be able to:

* Create, compile and execute a C Language program
* Explain the "behind the scenes" activities of your computer and operating system
* Perform basic input and output
* Use various types of control statements including if, if-else, if-else-if, switch, for, while, and do-while
* Create, implement and call your own functions
* Understand the usefulness and importance of pointers
* Work with files and directories
* Develop and manipulate data structures
* Create multi-module programs

## Course Materials and Technical Requirements

### Required Materials

* Prata, S. (2014). C Primer Plus, 6th ed. Pearson Ed.

### Recommended Materials

* *C: A Reference Manual* by Samuel P. Harbison and Guy L. Steele (Prentice Hall).

### Notation

Note that in the material following C code included in the text will look like this: Sample C code

### Use of Compilers for the Programming Assignments

Coding, testing, and debugging the assigned programs require the normal set of tools that programmers use:

* Source code editor
* Compiler
* Linker
* Debugger (not required, but often very helpful)

Most of the C compilers that you will find on the market today will come bundled with all these tools. Almost any compiler for sale in a store and many of the compilers available for download will satisfy the requirements for doing the homework in this course. The main characteristic that you should check for is that the compiler implements the complete C language. There are some shareware and demo or instructional compilers that omit key features, such as floating point. Trying to use an incomplete implementation of C could make it impossible for you to complete all the homework programs.

### ANSI vs. pre-ANSI Compliers

The ANSI Standard for C, which was adopted in 1989, was the first formal standard for the C language. Besides establishing a standard, it also defined some key extensions to the language, which constitute a real improvement. Given a choice between a pre-ANSI and an ANSI C compiler, the ANSI one is generally preferred. However, there are many systems, particularly UNIX systems, that still only have pre-ANSI compilers. There are also many application programs in use today that were developed in a pre-ANSI environment and are being maintained using ANSI compilers. This is possible because ANSI compilers can compile pre-ANSI programs. Because of this, the required text discusses both the ANSI way and the pre-ANSI way to do things in those instances when they are different.

### C vs. C++ Compliers

Most of the compilers you might consider buying for use in this course will be advertised as C++ compilers. This apparent incongruity is not actually a cause for alarm. Many of these compilers can operate in either C or C++ mode. The documentation that comes with the compiler should tell you how to select C mode. However, even if you operate your compiler in C++ mode, your C code will probably work anyway. C++ is almost entirely a superset of C. This means that most coding that works in C will also work in C++ and will usually give the same results.

### Graphical User Interface Environments

The sample programs in the textbook and the programming assignments are mainly designed for a user environment with simple text input/output such as MS-DOS or a UNIX shell. These programs can be run in a graphical user interface (GUI) environment such as Microsoft Windows or the Macintosh, but this is generally less convenient. Thus, if you have a choice of programming environments, a nongraphical environment will usually be easier to work in. However, it is certainly possible to do the work in a GUI environment.

### Technical Requirements

This course is built on a Learning Management system (LMS) called Canvas and you will need to meet these [computer specifications to participate within this online platform.](http://guides.instructure.com/s/2204/m/4214/l/82542-what-are-the-basic-computer-specifications-for-canvas" \o "Confirm your computer meets Canvas specifications)

#### Optional

Canvas allows you to record audio or video files of yourself and upload them in the course. Although doing so is not required for any of the activities, using these features will enhance your engagement in the course. If you would like to use these features, you will need to have a webcam and a microphone installed on your computer.

## Learning Activities

This course consists of six modules of study outlined at the end of this syllabus.  Each module includes a combination of:

* a reading assignment from the required text;
* reading my commentary, which provide explanation and examples on each topic;
* short answer questions;
* a programming assignment ; and
* a class discussion on the Discussion Assignments forum.

You must also take a proctored final exam.

### Reading Assignments

Reading Assignments include both readings from the text book as well as the instructor's commentary. The instructor's commentary on the assigned material is included online in each module. The commentary provides guidance through the assigned material and helps students focus on key concepts. Commentaries may also provide links to other related Internet resources.

After reading the relevant information for each lesson and considering the short answer questions, you may still have questions. Please visit the Student Lounge discussion forum to see if any other students have similar concerns. You may wish to post your questions there. Please also feel free to contact me directly using our course email.

### Short Answer Questions

In each module you will answer a series of short answer questions. These questions are related to the assigned reading.

### Programming Assignments

Additionally for every module, there are programming exercises that are related to the reading material. The programs will be graded primarily on whether they fulfill the requirements of the exercise, but I will also include readability, documentation, and consistency in coding style as factors in grading. Comments should be included where necessary to make the code readable.

To help you with these programming exercises, I will refer you to sample programs in the textbook--called listings--that deal with similar problems. Study these indicated samples carefully.

Please complete the programming exercises and submit your answers into the drop box at the end of this module by attaching your .c program files that you compiled and executed to the drop box. Please adhere to the following instructions when submitting your programming assignments.

1. Each .c program should be a separate attachment into the Module's Drop Box. The Module 1 Programming Assignment Drop Box, for example, should have two files attached, each with a .c file extension.
2. Do not zip or compress the files before attaching.
3. The file names of attached programs should match the file name identified in the header comments of each program.
4. If a module consists of multiple programming exercises, each exercise should have it's own .c source code file that was compiled and executed on your compiler. Attach the actual .c source code file that you compiled and executed for the assignment.

Check returned evaluated assignments carefully, note all comments, and make the suggested corrections.

### Discussions

### The Discussion Assignments forum provides a vehicle for classroom interactivity based on questions created specifically for each module. All posts are "public," that is, available for all class participants to read and comment on.

### When a module has a discussion assignment, please respond to the questions in a manner that reflects critical thinking. Please feel free, as well, to positively critique and offer leads and suggestions to comments and questions that other students have made.

### Each of us brings a unique perspective to this class based upon our life experiences and previous studies. But because of the continuous, open-enrollment nature of this course, at times it might be hard to sustain an actual conversation in the discussion threads with your fellow students. So instead, let's look to the discussion threads as a place to connect our multiple perspectives and construct an evolving knowledge base. The process will enrich your own studies, and the words you leave behind will help students who join the class even after you've gone on to new pursuits.

### Final Exam

The final exam is a 3 hour, proctored exam. You **may not** bring books or notes to the exam but you may bring blank notebook paper. The exam will be proctored, and you must schedule your appointment with a proctor ahead of time. More information about scheduling and taking your final exam is available on the [UC Berkeley Extension website](http://extension.berkeley.edu/static/studentservices/onlineinfo/?utm_source=internal&utm_medium=bullet&utm_campaign=online_exams#exams). You must pass the final exam in order to pass the course.

## Communication and Office Hours

You can always get in touch with me during the course. You can access course email by clicking on the Inbox link on the Corner Help toolbar (see also [Canvas Overview Video](http://guides.instructure.com/s/2204/m/4210/l/141852-canvas-overview-video" \t "_blank" \o "Canvas Overview Video)). You can expect me to respond to email within 48 hours of receiving messages unless I have notified the class otherwise (e.g., because of vacation or other reasons).

Please note: all course communication between students/instructor must occur within the course.

## Grading and Course Policies

Final grades will be assigned according to the following percentages:

* Discussion Assignment, 10%
* Short Answer Questions, 20%
* Programming Assignments, 50%
* Final Proctored Exam, 20%

You must pass the final exam with at least 70% to pass the course.

### Grading Information

Final grades follow the UC Berkeley grading system:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Letter Grade** | A | A- | B+ | B | B- | C+ | C | C- | D+ | D | D- | F |
| **Percentage** | 100-94 | 93-90 | 89-86 | 85-83 | 82-80 | 79-76 | 75-73 | 72-70 | 69-66 | 65-63 | 62-60 | < 60 |

Table 2: UC Berkeley Grading Systems

To [view your final grade and request official transcripts](http://extension.berkeley.edu/), login to your student account and go to "My Enrollment History."

When I grade your writing assignments, I'll be looking at content, organization, and mechanics. Please keep the following criteria in mind:

|  | **Poor** | **Needs Improvement** | **Meets Expectations** | **Exceptional** |
| --- | --- | --- | --- | --- |
| **Content** | Poor writing style with little or no specific details, no evidence of having studied the material, and/or off topic. | Adequately written; some points elaborated but with minimal use of concepts from the material. | Well written, most points elaborated with clear and detailed information that supports thoughts and ideas and uses concepts from the material. | Well written, fully elaborates points. Clear and detailed information supports thoughts and ideas and shows full acquisition of concepts from the material. |
| **Organization and Mechanics** | Little or no structure present. Grammatical errors interfere with comprehension. | Organization present but awkward. Some grammatical errors present. | Good organization with few statements out of place. Minor grammatical errors. | Clearly organized and remains focused. Few or no grammatical errors. |

Table 3: Criteria for Writing Assignments

In the Discussion Assignments forum, I'll also be looking for evidence of participation:

|  | **Poor** | **Needs Improvement** | **Meets Expectations** | **Exceptional** |
| --- | --- | --- | --- | --- |
| **Participation** | Minimal posts in number or length. Posts show little or no reflection on the topics or previous posts. | Posts address the topic but consist mostly of a rote repetition of the study materials. Little or no reflection on previous posts. | Posts address the topic with reflection. Many responses build on previous posts. | Posts show a genuine interest in contributing to the overall life of the forum. |

Table 4: Evidence of Participation for the Discussion Assignments Forum

Late Work Policy   
  
It is your job to complete class readings, assignments, and writings on time. All assignment due dates are listed in the course Calendar, and you are expected to work on only that week's assignments. You may not lag behind or work ahead. Instructors have up to 1 week to grade and return assignments. Any late work must be approved ahead of time by the instructor.

### DSP Accommodations

If you are a student with special needs and haven't already contacted the [Disabled Student Services](http://extension.berkeley.edu/static/studentservices/career/#disabled) (DSS), please contact the office right away. Be sure to review our detailed DSP accommodations instructions.

## Academic Integrity, Research, and Proper Citation

As an online student, you are encouraged to reach out to your fellow students in the online classroom to discuss materials and ask each other questions, but there are limits to this collaboration. Reviewing lecture and reading materials and studying for exams can be enjoyable and enriching things to do with fellow students. This is recommended. However, unless otherwise instructed, homework assignments are to be completed independently, and materials submitted as homework should be the result of your own independent work.

As a UC Berkeley student you are bound by the [Academic Integrity, Research and Proper Citation policies](http://extension.berkeley.edu/upload/academic_integrity.pdf) outlined in the [UC Berkeley Extension Code of Student Conduct Policy Statement](http://extension.berkeley.edu/upload/studentconduct.pdf) dated July 11, 2013 that clearly defines what constitutes cheating, as well as plagiarism and other forms of academic misconduct.

You must review all sections of the Academic Integrity Pledge and Course Policies Module within your Canvas course and complete the following item prior to gaining access to course content:

Take the Pledge to Academic Integrity

## Course Evaluation and Course End Date

### Course Evaluation

UC Berkeley Extension is committed to improving our online courses and instruction. Please take a few minutes to participate in our course evaluation in Module Nine, to share your opinions about this course. We are interested in your online learning experience, and your candid feedback will help us plan for the future and make improvements. Please complete the evaluation before your course End Date. The evaluation does not request any personal information, and your responses will remain strictly confidential.

### Course End Date

Your access to the online classroom will expire on the course End Date, which is indicated in the initial e-mail you received when you enrolled.

## Canvas Tech Support and UC Berkeley Extension Student Services

### Canvas Tech Support

The learning management system (LMS) used in this course is Canvas, which has convenient mobile apps for phones and tablets. Part of the orientation materials in your course will help you make sure that your computer is at par with Canvas specifications. Anytime you are in Canvas you can report problems, get support, and search Canvas user guides from the Help link on the top menu bar. Other options:

* Canvas Support 24/7 Hotline:  855-308-2758
* Email: [msupport@instructure.com](mailto:msupport@instructure.com)

### UC Berkeley Extension FAQs and Student Services

Start at the [Student Services webpage](http://extension.berkeley.edu/static/studentservices/" \t "_blank" \o "UC Berkeley Extensions Student Services) to find help with issues such as the following:

* Course registration
* Exam proctoring services
* Refunds, withdrawals, and transfers
* Grade options
* Requests for transcripts or official receipts

If you need further help, UC Berkeley’s [Extension Contact page](http://extension.berkeley.edu/static/online/" \l "contact" \t "_blank" \o "UC Berkeley Extension Contact page lists names, emails and phone numbers for Program Coordinators) lists Program Coordinator’s email and phone numbers for each academic department.

## Course Outline

You'll find complete instructions for your assignments within the course modules.

Module 1: Introduction to C

Read C Primer Plus Chapters 1-3

Read Module 1 Commentary

1.1 Compiler Basics

1.2 Creating, compiling and executing a program

1.3 Comments

1.4 C Program Structure

1.5 Variables

1.6 printf()

1.7 Selecting Variable Names and Data Types

Module 1 Short Answers

Module 1 Programming Assignment

Module 1 Discussion

Module 2: Input/Output and Operators

Read C Primer Plus Chapters 4-5

Read Module 2 Commentary

2.1: Character Strings and Arrays

2.2 Operators

2.3 Operator Precedence

2.4 printf()

2.5 scanf()

2.6 while Loop

2.7 type def

2.8 size\_t

Module 2 Short Answers

Module 2 Programming Assignment

Module 2 Discussion

Module 3: Loops, Decisions and Control

Read C Primer Plus Chapter 6-7

Read Module 3 Commentary

3.1 Character Input and Output

3.2 Conditionals

3.3 for Loop

3.4 Logical Control and Pitfalls

3.5 Conditional Operator

3.6 switch statement

3.7 Cast Operator

Module 3 Short Answers

Module 3 Programming Assignment

Module 3 Discussion

Module 4: Character Input/Output and Functions

Read C Primer Plus Chapters 8-9

Read Module 4 Commentary

4.1 Input and Output Concepts

4.2 Redirection and Testing

4.3 Mixed Number and Character Input Problems

4.4 Functions and Structured Programming

4.5 Creating Functions

4.6 Variable Scope

4.7 Pointers

Module 4 Short Answers

Module 4 Programming Assignment

Module 4 Discussion

Module 5: Arrays, Character Strings and Pointers

Read C Primer Plus Chapters 10-11

Read Module 5 Commentary

5.1 Arrays

5.2 Passing Arrays to Functions

5.3 Pointers Continued

5.4 Multidimensional Arrays

5.5 Compiler Treatment of String Constants

Module 5 Short Answers

Module 5 Programming Assignment

Module 5 Discussion

Final Project Proposal

Module 6: File Input/Output and Structures

Read C Primer Plus Chapters 12-14

Read Module 6 Commentary

6.1 Storage Classes, Scope and Extent

6.2 Dynamic Memory Allocation

6.3 Structures

6.4 Unions

6.5 Working with Files

6.6 const Variables

6.7 Multi-Module Programs

Module 6 Short Answers

Module 6 Programming Assignment

Module 6 Discussion

**PROCTORED FINAL EXAM**