

### Syllabus

#### Course Overview

This course uses the C programming language to introduce you to the fundamental concepts, design, and logic of programming languages.

You will learn the basic concepts of programming, such as variables, expressions, logic, flow control, functions, and data structures. C is a very influential programming language. Many other languages have adopted and adapted elements of C into their own languages.

#### Course Competencies

(Read Only)

To successfully complete this course, you will be expected to:

- 1 Define business problems that can be solved by using basic programming concepts and standards.
- 2 Describe fundamental programming constructs and processes.
- 3 Design a simple program that makes use of fundamental programming constructs.
- 4 Apply collaboration strategies in the software development process.
- 5 Communicate effectively.

#### Course Prerequisites

*There are no prerequisites for this course.*

## Syllabus >> Course Materials

### Required

The materials listed below are required to complete the learning activities in this course.

### Integrated Materials

Many of your required books are available via the VitalSource Bookshelf link in the courseroom, located in your Course Tools. Registered learners in a Resource Kit program can access these materials using the courseroom link on the Friday before the course start date. Some materials are available only in hard-copy format or by using an access code. For these materials, you will receive an email with further instructions for access. Visit the [Course Materials](#) page on Campus for more information.

#### Software

Free Software Foundation. (n.d.). Code::Blocks. Available from <http://www.codeblocks.org>

#### Book

Kochan, S. G. (2014). *Programming in C* (4th ed.). Indianapolis, IN: Developer's Library/Sams Pub. ISBN: 9780321776419.

Perry, G. (2014). *C programming: Absolute beginner's guide* (3rd ed.). Upper Saddle River, NJ: Pearson Education. ISBN: 9780789751980.

### Library

The following required readings are provided in the Capella University Library or linked directly in this course. To find specific readings by journal or book title, use [Journal and Book Locator](#). Refer to the [Journal and Book Locator library guide](#) to learn how to use this tool.

- Campbell, J. (2014). [Software development fundamentals: Introduction to math operators \[Video\]](#). Skillsoft Ireland.

### Suggested

The following materials are recommended to provide you with a better understanding of the topics in this course. These materials are not required to complete the course, but they are aligned to course activities and assessments and are highly recommended for your use.

### Optional

The following optional materials are offered to provide you with a better understanding of the topics in this course. These materials are not required to complete the course.

### Library

The following optional Skillsoft resources are available via the Capella University Library.

- Campbell, J. (2014). [Eclipse: Introduction to C/C++ \[Video\]](#). Skillsoft Ireland.
- Skillsoft. (n.d.). [Agile planning: Project initiating and requirements gathering \[Tutorial\]](#).
- Skillsoft. (n.d.). [Java SE 8 fundamentals: Introduction to Java \[Tutorial\]](#). Skillsoft Ireland.
- Skillsoft. (n.d.). [Software programming fundamentals: Design and development best practices \[Tutorial\]](#).

## External Resource

Please note that URLs change frequently. While the URLs were current when this course was designed, some may no longer be valid. If you cannot access a specific link, contact your instructor for an alternative URL. Permissions for the following links have been either granted or deemed appropriate for educational use at the time of course publication.

- Allain, A. (2011). [C File I/O and binary file I/O](#). Retrieved from <http://www.cprogramming.com/tutorial/cfileio.html>
- Allain, A. (2011). [Introduction to C](#). Retrieved from <http://www.cprogramming.com/tutorial/c/lesson1.html>
- Allain, A. (2011). [Lesson 2: If statements in C](#). Retrieved from <http://www.cprogramming.com/tutorial/c/lesson2.html>
- Allain, A. (2011). [Lesson 3: Loops](#). Retrieved from <http://www.cprogramming.com/tutorial/c/lesson3.html>
- Allain, A. (2011). [Lesson 4: Functions in C](#). Retrieved from <http://www.cprogramming.com/tutorial/c/lesson4.html>
- Allain, A. (2011). [Lesson 5: Switch case in C](#). Retrieved from <http://www.cprogramming.com/tutorial/c/lesson5.html>
- Allain, A. (2011). [Lesson 6: Pointers in C](#). Retrieved from <http://www.cprogramming.com/tutorial/c/lesson6.html>
- Allain, A. (2011). [Lesson 7: Structures in C](#). Retrieved from <http://www.cprogramming.com/tutorial/c/lesson7.html>
- Allain, A. (2011). [Lesson 9: C strings](#). Retrieved from <http://www.cprogramming.com/tutorial/c/lesson9.html>
- Bluefever Software. (Producer). (2013). [Beginning C programming – Part 10 – Arrays \(2\) \[Video\]](#). Retrieved from [https://www.youtube.com/watch?v=A4nMdukNL8&index=11&list=PLZ1QII7yudbdFfKY1eKV3x\\_bag04AMPJq](https://www.youtube.com/watch?v=A4nMdukNL8&index=11&list=PLZ1QII7yudbdFfKY1eKV3x_bag04AMPJq)
- Bluefever Software. (Producer). (2013). [Beginning C programming – Part 11 – Arrays \(3\) \[Video\]](#). Retrieved from [https://www.youtube.com/watch?v=TTCSG7y0AcY&index=12&list=PLZ1QII7yudbdFfKY1eKV3x\\_bag04AMPJq](https://www.youtube.com/watch?v=TTCSG7y0AcY&index=12&list=PLZ1QII7yudbdFfKY1eKV3x_bag04AMPJq)
- Bluefever Software. (Producer). (2013). [Beginning C programming – Part 12 – Arrays \(4\) \[Video\]](#). Retrieved from [https://www.youtube.com/watch?v=injAZP7jbU&index=13&list=PLZ1QII7yudbdFfKY1eKV3x\\_bag04AMPJq](https://www.youtube.com/watch?v=injAZP7jbU&index=13&list=PLZ1QII7yudbdFfKY1eKV3x_bag04AMPJq)
- Bluefever Software. (Producer). (2013). [Beginning C programming – Part 17 – Basic Functions \(1\) \[Video\]](#). Retrieved from [https://www.youtube.com/watch?v=k8BbDiMegpl&index=18&list=PLZ1QII7yudbdFfKY1eKV3x\\_bag04AMPJq](https://www.youtube.com/watch?v=k8BbDiMegpl&index=18&list=PLZ1QII7yudbdFfKY1eKV3x_bag04AMPJq)
- Bluefever Software. (Producer). (2013). [Beginning C programming – Part 18 – Basic functions \(2\) \[Video\]](#). Retrieved from [https://www.youtube.com/watch?v=VTQUCLYXYXQ&index=19&list=PLZ1QII7yudbdFfKY1eKV3x\\_bag04AMPJq](https://www.youtube.com/watch?v=VTQUCLYXYXQ&index=19&list=PLZ1QII7yudbdFfKY1eKV3x_bag04AMPJq)
- Bluefever Software. (Producer). (2013). [Beginning C programming – Part 19 – Basic functions \(3\) \[Video\]](#). Retrieved from <https://www.youtube.com/watch?>

- v=HhsJq4jzbzWl&index=20&list=PLZ1QII7yudbdFfKY1eKV3x\_bag04AMPJq
- Bluefever Software. (Producer). (2013). [Beginning C programming – Part 2 – Compiler errors and comments \[Video\]](https://www.youtube.com/watch?v=d_LG0PBj-xl&list=PLZ1QII7yudbdFfKY1eKV3x_bag04AMPJq&index=3). Retrieved from https://www.youtube.com/watch?v=d\_LG0PBj-xl&list=PLZ1QII7yudbdFfKY1eKV3x\_bag04AMPJq&index=3
  - Bluefever Software. (Producer). (2013). [Beginning C programming – Part 20 – Basic functions \(4\) \[Video\]](https://www.youtube.com/watch?v=vjiuswN8zcU&index=21&list=PLZ1QII7yudbdFfKY1eKV3x_bag04AMPJq). Retrieved from https://www.youtube.com/watch?v=vjiuswN8zcU&index=21&list=PLZ1QII7yudbdFfKY1eKV3x\_bag04AMPJq
  - Bluefever Software. (Producer). (2013). [Beginning C programming – Part 21 – Pointers \(1\) \[Video\]](https://www.youtube.com/watch?v=Xzsjec6Jo&index=22&list=PLZ1QII7yudbdFfKY1eKV3x_bag04AMPJq). Retrieved from https://www.youtube.com/watch?v=Xzsjec6Jo&index=22&list=PLZ1QII7yudbdFfKY1eKV3x\_bag04AMPJq
  - Bluefever Software. (Producer). (2013). [Beginning C programming – Part 22 – Pointers \(2\) \[Video\]](https://www.youtube.com/watch?v=j_dX7xaW9zQ&index=23&list=PLZ1QII7yudbdFfKY1eKV3x_bag04AMPJq). Retrieved from https://www.youtube.com/watch?v=j\_dX7xaW9zQ&index=23&list=PLZ1QII7yudbdFfKY1eKV3x\_bag04AMPJq
  - Bluefever Software. (Producer). (2013). [Beginning C programming – Part 23 – Pointers \(3\) \[Video\]](https://www.youtube.com/watch?v=_G3UmwZUhvE&index=24&list=PLZ1QII7yudbdFfKY1eKV3x_bag04AMPJq). Retrieved from https://www.youtube.com/watch?v=\_G3UmwZUhvE&index=24&list=PLZ1QII7yudbdFfKY1eKV3x\_bag04AMPJq
  - Bluefever Software. (Producer). (2013). [Beginning C programming – Part 24 – Pointers \(4\) \[Video\]](https://www.youtube.com/watch?v=QpwbCuSfJPI&index=25&list=PLZ1QII7yudbdFfKY1eKV3x_bag04AMPJq). Retrieved from https://www.youtube.com/watch?v=QpwbCuSfJPI&index=25&list=PLZ1QII7yudbdFfKY1eKV3x\_bag04AMPJq
  - Bluefever Software. (Producer). (2013). [Beginning C programming – Part 3 – Introduction to variables, operators \[Video\]](https://www.youtube.com/watch?v=sFZDsxCvEY&list=PLZ1QII7yudbdFfKY1eKV3x_bag04AMPJq&index=4). Retrieved from https://www.youtube.com/watch?v=sFZDsxCvEY&list=PLZ1QII7yudbdFfKY1eKV3x\_bag04AMPJq&index=4
  - Bluefever Software. (Producer). (2013). [Beginning C programming – Part 39 – Data structures #1 \[Video\]](https://www.youtube.com/watch?v=m1EHPnWOF8w&index=40&list=PLZ1QII7yudbdFfKY1eKV3x_bag04AMPJq). Retrieved from https://www.youtube.com/watch?v=m1EHPnWOF8w&index=40&list=PLZ1QII7yudbdFfKY1eKV3x\_bag04AMPJq
  - Bluefever Software. (Producer). (2013). [Beginning C programming – Part 4 – Floating point variables \[Video\]](https://www.youtube.com/watch?v=M1yZc0Z44vQ&list=PLZ1QII7yudbdFfKY1eKV3x_bag04AMPJq&index=5). Retrieved from https://www.youtube.com/watch?v=M1yZc0Z44vQ&list=PLZ1QII7yudbdFfKY1eKV3x\_bag04AMPJq&index=5
  - Bluefever Software. (Producer). (2013). [Beginning C programming – Part 40 – Data structures #2 – typedef \[Video\]](https://www.youtube.com/watch?v=YPtwEwGgXW4&index=41&list=PLZ1QII7yudbdFfKY1eKV3x_bag04AMPJq). Retrieved from https://www.youtube.com/watch?v=YPtwEwGgXW4&index=41&list=PLZ1QII7yudbdFfKY1eKV3x\_bag04AMPJq
  - Bluefever Software. (Producer). (2013). [Beginning C programming – Part 41 – Data structures #3 pointers \[Video\]](https://www.youtube.com/watch?v=f2vHkeKH99E&index=42&list=PLZ1QII7yudbdFfKY1eKV3x_bag04AMPJq). Retrieved from https://www.youtube.com/watch?v=f2vHkeKH99E&index=42&list=PLZ1QII7yudbdFfKY1eKV3x\_bag04AMPJq
  - Bluefever Software. (Producer). (2013). [Beginning C programming – Part 42 – Data structures & linked lists \[Video\]](https://www.youtube.com/watch?v=X1Ge86RQBRs&index=43&list=PLZ1QII7yudbdFfKY1eKV3x_bag04AMPJq). Retrieved from https://www.youtube.com/watch?v=X1Ge86RQBRs&index=43&list=PLZ1QII7yudbdFfKY1eKV3x\_bag04AMPJq
  - Bluefever Software. (Producer). (2013). [Beginning C programming – Part 43 – Data structures & linked lists #2 \[Video\]](https://www.youtube.com/watch?v=HxdZhdZjwQ&index=44&list=PLZ1QII7yudbdFfKY1eKV3x_bag04AMPJq). Retrieved from https://www.youtube.com/watch?v=HxdZhdZjwQ&index=44&list=PLZ1QII7yudbdFfKY1eKV3x\_bag04AMPJq
  - Bluefever Software. (Producer). (2013). [Beginning C programming – Part 44 – Data structures & linked lists #3 \[Video\]](https://www.youtube.com/watch?v=1joffYDrIIA&index=45&list=PLZ1QII7yudbdFfKY1eKV3x_bag04AMPJq). Retrieved from https://www.youtube.com/watch?v=1joffYDrIIA&index=45&list=PLZ1QII7yudbdFfKY1eKV3x\_bag04AMPJq

- Bluefever Software. (Producer). (2013). [Beginning C programming – Part 45 – Data structures & linked lists #4 \[Video\]](https://www.youtube.com/watch?v=SPHcwn6Jds&index=46&list=PLZ1QII7yudbdFfKY1eKV3x_bag04AMPJq). Retrieved from https://www.youtube.com/watch?v=SPHcwn6Jds&index=46&list=PLZ1QII7yudbdFfKY1eKV3x\_bag04AMPJq
- Bluefever Software. (Producer). (2013). [Beginning C programming – Part 46 – Read text from a file \[Video\]](https://www.youtube.com/watch?v=9KVqEpiKxTY&index=47&list=PLZ1QII7yudbdFfKY1eKV3x_bag04AMPJq). Retrieved from https://www.youtube.com/watch?v=9KVqEpiKxTY&index=47&list=PLZ1QII7yudbdFfKY1eKV3x\_bag04AMPJq
- Bluefever Software. (Producer). (2013). [Beginning C programming – Part 47 – Read text from a file by character \[Video\]](https://www.youtube.com/watch?v=HkM4kV7upWA&index=48&list=PLZ1QII7yudbdFfKY1eKV3x_bag04AMPJq). Retrieved from https://www.youtube.com/watch?v=HkM4kV7upWA&index=48&list=PLZ1QII7yudbdFfKY1eKV3x\_bag04AMPJq
- Bluefever Software. (Producer). (2013). [Beginning C programming – Part 48 – Writing strings to file \[Video\]](https://www.youtube.com/watch?v=QS7AiCN3KjQ&index=49&list=PLZ1QII7yudbdFfKY1eKV3x_bag04AMPJq). Retrieved from https://www.youtube.com/watch?v=QS7AiCN3KjQ&index=49&list=PLZ1QII7yudbdFfKY1eKV3x\_bag04AMPJq
- Bluefever Software. (Producer). (2013). [Beginning C programming – Part 48.5 – Writing text by character to a file \[Video\]](https://www.youtube.com/watch?v=HLeZfmQOgaY&index=50&list=PLZ1QII7yudbdFfKY1eKV3x_bag04AMPJq). Retrieved from https://www.youtube.com/watch?v=HLeZfmQOgaY&index=50&list=PLZ1QII7yudbdFfKY1eKV3x\_bag04AMPJq
- Bluefever Software. (Producer). (2013). [Beginning C programming – Part 49 – Writing data to a file \[Video\]](https://www.youtube.com/watch?v=JYivqdPsUU&index=51&list=PLZ1QII7yudbdFfKY1eKV3x_bag04AMPJq). Retrieved from https://www.youtube.com/watch?v=JYivqdPsUU&index=51&list=PLZ1QII7yudbdFfKY1eKV3x\_bag04AMPJq
- Bluefever Software. (Producer). (2013). [Beginning C programming – Part 5 – If, else, and, not, or \[Video\]](https://www.youtube.com/watch?v=wF2ViHZLa9Y&list=PLZ1QII7yudbdFfKY1eKV3x_bag04AMPJq&index=6). Retrieved from https://www.youtube.com/watch?v=wF2ViHZLa9Y&list=PLZ1QII7yudbdFfKY1eKV3x\_bag04AMPJq&index=6
- Bluefever Software. (Producer). (2013). [Beginning C programming – Part 50 – Reading data from a file \[Video\]](https://www.youtube.com/watch?v=FoO1jCCT5Yg&index=52&list=PLZ1QII7yudbdFfKY1eKV3x_bag04AMPJq). Retrieved from https://www.youtube.com/watch?v=FoO1jCCT5Yg&index=52&list=PLZ1QII7yudbdFfKY1eKV3x\_bag04AMPJq
- Bluefever Software. (Producer). (2013). [Beginning C programming – Part 51 – Commandline arguments \[Video\]](https://www.youtube.com/watch?v=jl4r7u7lfJY&index=53&list=PLZ1QII7yudbdFfKY1eKV3x_bag04AMPJq). Retrieved from https://www.youtube.com/watch?v=jl4r7u7lfJY&index=53&list=PLZ1QII7yudbdFfKY1eKV3x\_bag04AMPJq
- Bluefever Software. (Producer). (2013). [Beginning C programming – Part 6 – Switch statements and user input \[Video\]](https://www.youtube.com/watch?v=M0U2b3HkHXs&list=PLZ1QII7yudbdFfKY1eKV3x_bag04AMPJq&index=7). Retrieved from https://www.youtube.com/watch?v=M0U2b3HkHXs&list=PLZ1QII7yudbdFfKY1eKV3x\_bag04AMPJq&index=7
- Bluefever Software. (Producer). (2013). [Beginning C programming – Part 7 – Loops \[Video\]](https://www.youtube.com/watch?v=Q2TAdG5IOC8&index=8&list=PLZ1QII7yudbdFfKY1eKV3x_bag04AMPJq). Retrieved from https://www.youtube.com/watch?v=Q2TAdG5IOC8&index=8&list=PLZ1QII7yudbdFfKY1eKV3x\_bag04AMPJq
- Bluefever Software. (Producer). (2013). [Beginning C programming – Part 8 – User input with fgets\(\) and strncmp\(\) \[Video\]](https://www.youtube.com/watch?v=vDUqc1j73gs&index=9&list=PLZ1QII7yudbdFfKY1eKV3x_bag04AMPJq). Retrieved from https://www.youtube.com/watch?v=vDUqc1j73gs&index=9&list=PLZ1QII7yudbdFfKY1eKV3x\_bag04AMPJq
- Bluefever Software. (Producer). (2013). [Beginning C programming – Part 9 – Arrays \(1\) \[Video\]](https://www.youtube.com/watch?v=65hNn5kPfiQ&index=10&list=PLZ1QII7yudbdFfKY1eKV3x_bag04AMPJq). Retrieved from https://www.youtube.com/watch?v=65hNn5kPfiQ&index=10&list=PLZ1QII7yudbdFfKY1eKV3x\_bag04AMPJq
- Computer Hope. (2014). [MS-DOS](http://www.computerhope.com/jargon/m/msdos.htm). Retrieved from http://www.computerhope.com/jargon/m/msdos.htm
- Cprogramming.com. (2011). [C tutorial](http://www.cprogramming.com/tutorial/c-tutorial.html). Retrieved from http://www.cprogramming.com/tutorial/c-tutorial.html
- Learn C.org. (n.d.). [For loops](http://www.learn-c.org/en/For_loops). Retrieved from http://www.learn-c.org/en/For\_loops
- Learn C.org. (n.d.). [Strings](http://www.learn-c.org/en/Strings). Retrieved from http://www.learn-c.org/en/Strings
- Learn C.org. (n.d.). [Variables and types](http://www.learn-c.org/en/Variables_and_Types). Retrieved from http://www.learn-c.org/en/Variables\_and\_Types

- Learn-C.org. (n.d.). [While loops](http://www.learn-c.org/en/While_loops). Retrieved from [http://www.learn-c.org/en/While\\_loops](http://www.learn-c.org/en/While_loops)
- Learn-C.org. (n.d.). [Function arguments by reference](http://www.learn-c.org/en/Function_arguments_by_reference). Retrieved from [http://www.learn-c.org/en/Function\\_arguments\\_by\\_reference](http://www.learn-c.org/en/Function_arguments_by_reference)
- Learn-C.org. (n.d.). [Function pointers](http://www.learn-c.org/en/Function_Pointers). Retrieved from [http://www.learn-c.org/en/Function\\_Pointers](http://www.learn-c.org/en/Function_Pointers)
- Learn-C.org. (n.d.). [Functions](http://www.learn-c.org/en/Functions). Retrieved from <http://www.learn-c.org/en/Functions>
- Learn-C.org. (n.d.). [Hello, world!](http://www.learn-c.org/en>Hello%2C_World%21) Retrieved from [http://www.learn-c.org/en/Hello%2C\\_World%21](http://www.learn-c.org/en>Hello%2C_World%21)
- Learn-C.org. (n.d.). [Pointers](http://www.learn-c.org/en/Pointers). Retrieved from <http://www.learn-c.org/en/Pointers>
- Learn-C.org. (n.d.). [Structures](http://www.learn-c.org/en/Structures). Retrieved from <http://www.learn-c.org/en/Structures>
- Sanfoundry. (2014). [C program to implement a queue using an array](http://www.sanfoundry.com/c-program-queue-using-array/). Retrieved from <http://www.sanfoundry.com/c-program-queue-using-array/>
- Sanfoundry. (2014). [C program to implement a stack](http://www.sanfoundry.com/c-program-stack-implementation). Retrieved from <http://www.sanfoundry.com/c-program-stack-implementation>.
- Sanfoundry. (2014). [C program to implement queue data structure using a linked list](http://www.sanfoundry.com/c-program-stack-using-linked-list/). Retrieved from <http://www.sanfoundry.com/c-program-stack-using-linked-list/>
- Scrum.org. (2013). [The Scrum guide](http://scrumguides.org/docs/scrumguide/v1/Scrum-Guide-US.pdf#zoom=100). Retrieved from <http://scrumguides.org/docs/scrumguide/v1/Scrum-Guide-US.pdf#zoom=100>
- [Scrumalliance](http://www.scrumalliance.org). (2014). Retrieved from <http://www.scrumalliance.org>
- Tutorialspoint. (2014). [C – Error handling](http://www.tutorialspoint.com/cprogramming/c_error_handling.htm). Retrieved from [http://www.tutorialspoint.com/cprogramming/c\\_error\\_handling.htm](http://www.tutorialspoint.com/cprogramming/c_error_handling.htm)
- Wikibooks. (2013). [C programming/Error handling](http://en.wikibooks.org/wiki/C_Programming/Error_handling). Retrieved from [http://en.wikibooks.org/wiki/C\\_Programming/Error\\_handling](http://en.wikibooks.org/wiki/C_Programming/Error_handling)

## Unit 1 >> An Introduction to Programming and the C Language

### Introduction

Computers are powerful tools that are a part of daily life. That power, though, is only useful when running programs consisting of a structured set of instructions to apply their abilities to solving problems. In this unit, you will start learning how to create your own programs and learn to use the computer's capabilities so you can move from a user of programs to a creator of programs. You will learn about basic elements used in programs, such as variables, data types, expressions, and functions for the basic input and output of text. You will also gain experience in writing C code, compiling the code to produce an executable program, and test-running the program in the development environment.

### Learning Activities

#### u01s1 - Studies

### Readings

Use your *C Programming: Absolute Beginners Guide* text to read the following:

Introduction, pages 1–4.

- Chapter 1, "What Is C Programming, and Why Should I Care," pages 5–11.
- Chapter 2, "Writing Your First C Program," pages 13–22.
- Chapter 3, "What Does This Do? Clarifying Your Code with Comments," pages 23–29.
- Chapter 4, "Your World Premiere – Putting Your Program's Results Up On the Screen," pages 31–40.
- Chapter 5, "Adding Variables to Your Programs," pages 41–48.
- Chapter 6, "Adding Words to Your Programs," pages 49–56.
- Chapter 7 (part), "Making Your Programs More Powerful With #include and #define," pages 57–61.
- Chapter 8, "Interacting With Users," pages 65–72.
- Chapter 9, "Crunching the Number – Letting C Handle the Math for You," pages 73–81.

Use your *Programming in C* text to read the following:

- Chapter 1, "Some Fundamentals," pages 5–10.
- Chapter 2, "Compiling and Running Your First Program," pages 11–19.
- Chapter 3 (part), "Variables, Data Types, and Arithmetic Expressions," pages 21–36

In this course, you will use CodeBlocks to edit, compile, and run your code, so the specific steps for compiling and running code mentioned in Chapters 1 and 2 above do not apply.

## Optional Reading

For more information on `printf()` and `scanf()`, see the following reference section in *Programming in C*:

- Chapter 15 (part), "Input and Output Operations in C," pages 346–358.

You may wish to refer to the section "Formatted I/O: `printf()` and `scanf()`" throughout the course for fuller details on the `printf()` and `scanf()` functions.

## Optional Skillsoft Resources

- Campbell, J. (2014). [Eclipse: Introduction to C/C++ \[Video\]](#). Skillsoft Ireland.

## u01s2 - Software Preparation and Technology Access

In this course, you will be using software and technology that is needed to complete designated activities and assignments. There is no additional cost for this software and technology. Some software packages will be made available to you at no additional cost through Capella's subscription with Microsoft, while other software packages are available for free download through open-source licensing.



Capella University requires learners to meet certain minimum [computer requirements](#). Please note that some software required for a course may exceed these minimum requirements. Check the requirements for the software you may need to download and install to make sure it will work on your device. Most software will require a Windows PC. If you use a Mac, refer to [Installing a Virtual Environment and Windows on a Mac](#).

The software and technologies below are strongly recommended to support you in completing the course objectives. If you have access to other tools that you believe may still meet the requirements of this course, please discuss your selected alternatives with your instructor.

If you use assistive technology or any alternative communication methods to access course content, please contact [Disability Services](#) with any access-related questions or to request accommodations.

For this course, follow the instructions provided through the links below to download and install software or register for an account, as required.

## Open-Source Software

- Code::Blocks: Go to [Downloading and Installing Code::Blocks With C/C++](#).

If you encounter any difficulties in the download and installation process, post a detailed question in the Ask Your Instructor section of the course. Your instructor should be able to help you or point you in the right direction for the answers you need.

### **u01s3 - End of Chapter Exercises**

To prepare for the programming assignments in this course, complete the end of the chapter exercises in your text using Code::Blocks. These assignments have been designed in a way that ensure that the book exercises will support the completion of the assignment exercises.

If you have questions about completing these exercises, please ask your instructor.

### **u01d1 - The Programming Process**

Based on your reading and your initial experiments with writing C using Code::Blocks, describe the process of creating a C program. Discuss the process of moving from a concept to code to an executable program. Can you relate any personal or professional experience to this process of creating software? What recommendations do you have for your classmates who are also trying to learn this process?

Please focus on communicating in writing that is professional in tone and free of spelling and grammatical errors. **Complete your initial post by Wednesday, 12:00 midnight Central Time.**

## Response Guidelines

Prior to Sunday evening, read and provide high-quality replies to least two of your peers' posts. In a constructive manner, point out areas of concern or of note. Include an example from your personal experience or reasoning, if applicable.

### Course Resources

[Undergraduate Discussion Participation Scoring Guide](#)

## u01a1 - Programming in C: Unit 1 Exercises

This assignment consists of four short programs. Use Code::Blocks to complete your work. You will submit the four C source code files for this assignment. Please zip the files and include your name in the name of the compressed file.

## Part 1: Commenting Code

Including comments in source code that explains what the code does and how it does it are an important professional best practice. Code that is well-commented and well-formatted facilitates communication between programmers and others needing to work with the code. Commenting your code needs to be a habit from the beginning. Adding comments to code will also help reinforce important C programming concepts in your own mind.

Type in the code provided in the assignment Resources, build/compile the program, and run it. Typing the code in yourself rather than copying and pasting will help you get to know the C language. When you have the code running, go through the code and add comments to explain what the code does and how it does it. Be sure to add a header like the following to all code files (with your name and the date).

```
/* Ann Smith - IT2240 - U01a1 - Part 1
```

\*/

You should add at least five comments to the code provided. Be sure to recompile and test the code after you have added your comments to make sure you have not introduced any syntax errors.

## Part 2: About You

Create a program that prints three lines of text about yourself. Be sure to include appropriate spacing and new line characters to make the output in the terminal window readable. As with all programming code, but sure to format the program correctly and include comments and a header for the file, such as shown above.

## Part 3: Integer and Floating Point Arithmetic

Write a single program that prints the results of two arithmetic expressions using integers and the results of two floating-point calculations (for a total of 4 calculations). Print out both the arithmetic expressions and the answers calculated by the code. For this program, it is fine to use "hard-coded" values (the numbers used in the calculations are provided in the code) rather than user input. Be sure to format the code correctly and include relevant comments in the source code (including the comment at the top of the file with your name and identification of the program).

## Part 4: User Input

Write a program that prompts the user to enter a character, an integer, and floating point number. After reading the input, the program should print out the values entered. Be sure to format the code correctly and include relevant comments.

When you are ready to turn in your assignment, please zip your .c source code files together for submission.

Course Resources

Unit 1 Part 1 Code for Commenting

## Unit 2 >> Logic and Flow Control

### Introduction

Computers are rigorously logical devices. Combining the computer's logical abilities with the ability to evaluate expressions yields the ability to solve complex problems. In this unit, you will learn about key logical elements in

C programs.

## Learning Activities

### u02s1 - Studies

Use your *C Programming: Absolute Beginners Guide* text to read the following:

- Chapter 10, "Powering Up Your Variables With Assignments and Expressions," pages 83–90.
- Chapter 11, "The Fork in the Road—Testing Data to Pick a Path," pages 91–101.
- Chapter 12, "Juggling Several Choices With Logical Operators," pages 103–113.
- Chapter 13, "A Bigger Bag of Tricks—Some More Operators for Your Programs," pages 115–122.
- Chapter 17, "Making the Case for the Switch Statement," pages 149–162.

Use your *Programming in C* text to read:

- Chapter 3 (part), "Variables, Data Types, and Arithmetic Expressions," pages 36–39.
- Chapter 5, "Making Decisions," pages 65–92.

A couple of the examples in Chapter 5 include for loops. A loop is a programming construct that allows repeating of a sequence of commands a specific number of times. You will learn about them in the next unit. For now, just focus on the use of the conditional constructs.

## Optional Skillsoft Resource

- Campbell, J. (2014). [Software development fundamentals: Introduction to math operators \[Video\]](#). Skillsoft Ireland.

### u02d1 - Planning the Logic of the Program

Before starting to write code, it is important to have a clear picture of what the program should do and the steps in the process for reaching the desired result. Choose either the program for part 2 or part 3 of this unit's assignment and write out, in plain English, what the program needs to do and describe the steps in the process to solve the problem. Focus on a clear explanation of the steps in solving the problem, not writing C. While it is a good idea to repeat this process for both parts 2 and 3 of the assignment, you need to cover one program in this discussion post.

Connect with 1–2 other learners and ask them to read your post and provide constructive critique. In your initial post, be sure to incorporate their feedback.

Please focus on communicating in writing that is professional in tone and free of spelling and grammatical errors. **Complete your initial post by Wednesday, 12:00 midnight Central Time.**

## Response Guidelines

Prior to Sunday evening, read and provide high-quality replies to at least two of your peers' posts. In a constructive manner, point out areas of concern or of note. Are there areas that are unclear or need to be developed further?

Collaborate with two of your classmates by replying to their posts with a constructive critique. Are there areas that are unclear or need to be developed further? Are there potential benefits to one approach over another?

### Course Resources

[Undergraduate Discussion Participation Scoring Guide](#)

[Critical Thinking](#)

## u02a1 - Programming in C: Unit 2 Exercises

### Part 1: Commenting Code

Enter the code provided in the assignment Resources in CodeBlocks, compile, and run. Pay attention to the formatting of the code. Add at least eight comments to explain what the program does and how it works. Be sure to include a comment at the top the file as a header like the one used in Unit 1. After adding your comments, build and test the program to make sure it still builds and runs correctly.

### Part 2: Pay Calculator

Write a program that requests the user to enter hours worked in a week and the hourly wage. The program should then print the gross pay, the taxes, and the net pay. Assume the following:

1. Overtime (in excess of 40 hours for week) = time and a half.
2. Tax rate: 15% of the first \$600, 20% of the rest.

Remember that correct code formatting and meaningful comments in the source code are required for all code that you write. Be sure to include extended comments about the logic used in your program.

## Part 3: Cab Fare Calculator

This program should calculate the fare for a cab ride based on the rates provided below. Ask user to input distance (measured to 1/10 of mile), number of passengers, and if going to/coming from the airport. Calculate fare for trip using these rates:

1. Minimum fare: \$5.00 (fares less than this rounded up to \$5.00).
2. \$1.80 per mile (charged by 1/10 of mile. 2.1 miles = \$3.78).
3. 1st additional passenger, if more than 1 rider: \$1.00.
4. Each additional passenger (after 1st additional passenger): \$0.50.
5. Airport surcharge: \$2.00.

Remember that correct code formatting and meaningful comments in the source code are required for all code that you write. Be sure to include extended comments about the logic used in your program.

When you submit your assignment, please be sure to include the .c source code files for all three programs.

Course Resources

Unit 2 Part 1 Code for Commenting

## Unit 3 >> Loops and Arrays

### Introduction

Many problems can be solved by carrying out a sequence of steps repetitively. The computer's ability to carry out repetitive operations quickly is one of its strengths. In this unit you will learn about the looping constructs available to the C programmer. In addition to loops, this unit also covers a data structure called an array.

### Learning Activities

#### u03s1 - Studies

Use your *C Programming: Absolute Beginners Guide* text to read the following:

- Chapter 14, "Code Repeat—Using Loops to Save Time and Effort," pages 123–129.
- Chapter 15, "Looking for Another Way to Create Loops," pages 131–140.
- Chapter 16, "Breaking In and Out of Looped Code," pages 141–147.

- Chapter 20 (part), "Advanced Math (For the Computer, Not You)," pages 187–191.
- Chapter 21, "Dealing With Arrays," pages 193–200.
- Chapter 22, "Searching Arrays," pages 201–208.
- In Chapter 20, read the section "Getting Random" on the pages indicated.

Use your *Programming in C* text to read the following:

- Chapter 4, "Program Looping," pages 43–64.
- Chapter 6, "Working With Arrays," pages 95–118.

## Optional Skillsoft Resources

- Skillsoft. (n.d.). [Software programming fundamentals: Design and development best practices \[Tutorial\]](#).
- Skillsoft. (n.d.). [Agile planning: Project initiating and requirements gathering \[Tutorial\]](#).

### u03d1 - Planning the Logic of the Program With a Loop

Before starting to write code, it is important to have a clear picture of what the program should do and the steps in the process for reaching the desired result. Choose either the program for part 2 or part 3 of this unit's assignment and write out, in plain English, what the program needs to do and describe the steps in the process to solve the problem. Focus on a clear explanation of the process of solving the problem, not writing C. While it is a good idea to do this for both programs, you only need to discuss one in this post.

You should also discuss the type of loop you plan to use for your program and why. What role does the loop play in the program?

Please focus on communicating in writing that is professional in tone and free of spelling and grammatical errors. **Complete your initial post by Wednesday, 12:00 midnight Central Time.**

## Response Guidelines

Prior to Sunday evening, read and provide high-quality replies to at least two of your peers' posts. In a constructive manner, point out areas of concern or of note. Include an example from your personal experience or reasoning, if applicable.

Course Resources

Undergraduate Discussion Participation Scoring Guide

## u03a1 - Programming in C: Unit 3 Exercises

### Part 1: Commenting Code

Enter the code provided in the assignment Resources in CodeBlocks, compile, and run. Pay attention to the formatting of the code. Add at least eight comments to explain what the program does and how it works. Be sure to include a comment at the top of the file as a header like the one used in Unit 1. After adding your comments, build and test the program to make sure it still builds and runs correctly.

### Part 2: Guessing Game

This program is a guessing game that counts how many tries it takes for a player to guess a random number in the range from 1 to 20 (inclusive) that the computer has generated. The program should prompt the user for a guess and then indicate if the guess was too high, too low, or correct. When the player guesses correctly, the program should print out how many guesses the player made. Your C code must be both well-formatted and well-commented.

Be sure to include extended comments about the loop used in your code.

### Part 3: Simple Grade Book

This program should use an array to store percentage grades in the range from 0 to 100 (inclusive). The program should allow the user to indicate when he or she is done entering grades. When the user is done entering grades, the program should print out the grades entered. Your C code must be both well-formatted and well-commented.

Be sure to include extended comments about the loop used in your code.

When you submit your assignment, please be sure to include the .c source code files for all three programs.

#### Course Resources

[Unit 3 Part 1 Code for Commenting](#)



## Introduction

Functions in C allow the programmer to reuse sequences of instructions by name rather than having to retype the actual underlying code. There are many useful functions included in the C standard library, but programmers also create their own functions. Functions can both receive input data from and return data to the sections of the program that call them.

Pointers provide a powerful and flexible way to access data in memory and share it between different parts of a program. Pointers are a very useful tool, but programmers must use them carefully to avoid problems.

## Learning Activities

### u04s1 - Studies

Use your *C Programming: Absolute Beginners Guide* text to read the following:

- Chapter 24, "Solving the Mystery of Pointer," pages 221–229.
- Chapter 25, "Arrays and Pointer," pages 231–241.
- Chapter 26, "Maximizing Your Computer's Memory," pages 243–255.
- Chapter 30, "Organizing Your Code With Functions," pages 285–292.
- Chapter 31, "Passing Variables to Your Functions," pages 293–303.
- Chapter 32, "Returning Data From Your Functions," pages 305–312.

Use your *Programming in C* text to read:

Chapter 7, "Working With Functions," pages 119–160.

- Chapter 10 (part), "Pointers," pages 233–239, 251–275.

Chapter 10 in *Programming in C* includes a discussion of using pointers with structures (structs) on pages 239–251. This course covers structs in the next unit, so you may want to wait until the next unit to read this section of the chapter.

## Optional Skillsoft Resource

- Skillsoft. (n.d.). [Java SE 8 fundamentals: Introduction to Java \[Tutorial\]](#).

### u04d1 - Planning the Logic of the Program

Before starting to write code, it is important to have a clear picture of what the program should do and the steps in the process for reaching the desired result. Choose either the program for part 1 or part 2 of this unit's assignment and write out in plain English what the program needs to do and describe the steps in the process to solve the problem. Focus on a clear explanation of the process of solving the problem, not writing C. While it is a good idea to do this for both programs, you only need to discuss one in this post.

Please focus on communicating in writing that is professional in tone and free of spelling and grammatical errors. **Complete your initial post by Wednesday, 12:00 midnight Central Time.**

## Response Guidelines

Prior to Sunday evening, read and provide high-quality replies to at least two of your peers' posts. In a constructive manner, point out areas of concern or of note. Include an example from your personal experience or reasoning, if applicable.

Course Resources

Undergraduate Discussion Participation Scoring Guide

### u04a1 - Programming in C: Unit 4 Exercises

#### Part 1: Simple Grade Book, Version 1.1

Modify the grade book code from Unit 3 so that it uses heap memory to store percentage grades in the range from 0 to 100 (inclusive). The program should allow the user to indicate when he or she is done entering grades (since the user may not have grades to fill the whole array). When the user is done entering grades, the program should print out the grades entered by the user. Be sure to free the heap memory before the program ends.

#### Part 2: Simple Grade Book, Version 2.0

Modify the grade book code from Unit 3 so that it uses 3 custom functions to calculate the average grade (arithmetic mean, not letter grade), report the highest grade, and find the lowest grade entered. This version of the program does not need to use heap memory, though you are welcome to do so. The program should allow the user to indicate when he or she is done entering grades (since the user may not have grades to fill the whole array). When the user is done entering grades, the program should print out the grades entered by the user. The program should also display the average grade, highest grade, and lowest grade.

Be sure to include extended comments to explain the roles of your custom functions.

When you submit your assignment, please be sure to include the .c source code files for both programs.

## Unit 5 >> Data Structures and a Simple Sort

### Introduction

This unit introduces you to the use of C data structures called structs used to organize more complex data. You will also learn that structs can be used like any data type, including with arrays and pointers.

You will also learn a relatively simple technique called a bubble search to sort data stored as an array of structs.

### Learning Activities

#### u05s1 - Studies

Use your *C Programming: Absolute Beginners Guide* text to read the following:

- Chapter 23, "Alphabetizing and Arranging Your Data," pages 209–220.
- Chapter 27, "Setting Up Your Data With Structures," pages 258–266.

Use your *Programming in C* text to read the following:

- Chapter 8, "Working With Structures," pages 163–189.
- Chapter 10 (part), "Pointers," pages 239–251.

### Optional Skillsoft Resource

- Skillsoft. (n.d.). [Software programming fundamentals: Design and development best practices \[Tutorial\]](#).

#### u05d1 - Planning the Logic of the Program

Before starting to write code, it is important to have a clear picture of what the program should do and the steps in the process for reaching the desired result. Choose either the program for part 1 or part 2 of this unit's

assignment and write out in plain English what the program needs to do and describe the steps in the process to solve the problem. Focus on a clear explanation of the process of solving the problem, not writing C. While it is a good idea to do this for both programs, you only need to discuss one in this post.

Please focus on communicating in writing that is professional in tone and free of spelling and grammatical errors. **Complete your initial post by Wednesday, 12:00 midnight Central Time.**

## Response Guidelines

Prior to Sunday evening, read and provide high-quality replies to at least two of your peers' posts. In a constructive manner, point out areas of concern or of note. Include an example from your personal experience or reasoning, if applicable.

### Course Resources

[Undergraduate Discussion Participation Scoring Guide](#)

## u05a1 - Programming in C: Unit 5 Exercises

### Part 1: Date Calculation

Write a program that asks the user for a date in the format mm/dd/yyyy, using a custom struct to store the date information entered in memory. The program should call a custom function that adds seven days to the user's date, passing in the data in and out of the custom function using the custom struct for the dates. The program should then print out the original date entered and the date one week later. As always, be sure to format your C code correctly and include meaningful comments to explain what the code does and how it works.

### Part 2: Simple Grade Book, Version 3.0

Modify the grade book program from Unit 3 to use a custom struct to hold the student's ID number and the percentage score for each item in the grade book. The program should accept the entry of ID numbers and percentage grades (0–100) until the user signals that he or she is done entering grades. The program should then print out the ID numbers and grades entered by the user, sorted by ID number. Again, be sure to format the code properly and provide comments in the C code.

When you submit your assignment, please be sure to include the .c source code files for all three programs.