Cisco Networking Academy

C Advanced

Course Overview

This course is designed for learners to explore the evolution of C, variable parameter handling, memory and string manipulation, internationalization (I18N), processes and threads, advanced data type operations, and network socket programming.

Benefits

With advanced C skills, your learners will write optimized code that runs faster and uses resources more efficiently, a skill that is valuable in performance-critical applications like gaming engines or scientific computations. Proficiency in C can also set your learners apart in the job market.

- ✓ Develop skills for entry-level programming roles
- ✓ Prepare for CLP certification exam
- ✓ Explore programming challenges like portability issues and undefined behaviors

Course Details

Target Audience: 2-year and 4-year college and university students

Estimated Time to Completion: 70 hours

Prerequisites: CLA: Programming Essentials in C course, CLA certification, or equivalent
Recommended Preparation: C Essentials 1 & 2

Course Delivery: Instructor-led

Learning Component Highlights:
- 8 modules of interactive instructional content
- 18 practice labs
- Quizzes, chapter and final exams

Course Recognitions: None

Certification Alignment: CLP: C Certified Professional Programmer

Requirements

- ASC Alignment: Recommended
- Instructor Training: Optional
- Physical Equipment: Not Required

Developed in collaboration with
The C++ Institute

C++ Institute
Open Education & Development Group

C Advanced

Scope and Sequence

Version 1.0

Developed in collaboration with
OpenEDG C++ Institute

Contents

Target Audience

C Advanced (CA) is specifically designed for individuals who have successfully completed the intermediate-level *C Essentials 2* course and are now aiming to elevate their proficiency in C programming to a professional level.

This course is exceptionally suited for:

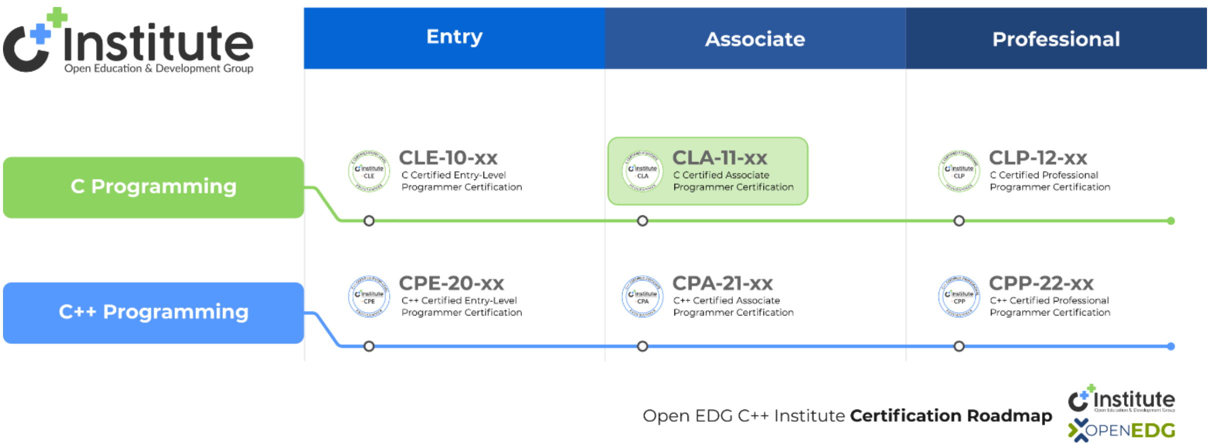
- Programmers with an intermediate knowledge of C seeking to master advanced programming concepts.
- Students and professionals targeting the [CLP – C Certified Professional Programmer] certification.
- Individuals aspiring to advance their careers in areas such as software development, network programming, or operating system development.
- Self-taught programmers looking to significantly broaden their understanding and capabilities in C programming.

Course Prerequisites

- Successful completion of C Essentials 2, CLA certification, or possessing equivalent intermediate knowledge in C programming.
- A solid grasp of intermediate programming concepts including but not limited to functions, structures, file operations, and error handling.

Certification Alignment

C Advanced is aligned with the CLP – C Certified Professional Programmer certification, serving as the direct preparatory stage for this advanced certification. Building upon the foundations laid by *C Essentials 1* and *C Essentials 2*, this course readies candidates for the CLP exam by diving deep into complex programming concepts, advanced features of the C language, and practical problem-solving techniques necessary for the certification.



The CLP certification is a testament to an individual's comprehensive expertise in advanced C programming, covering areas such as memory management, complex data structures, low-level system operations, and advanced error handling. It benchmarks proficiency for demanding programming roles and is an esteemed credential for those aiming to distinguish themselves in the field of C programming.

Candidates can take the *CLP – C Certified Professional Programmer* certification exam via the TestNow™ – OpenEDG Testing Service Platform, and through the network of Pearson VUE Testing Centers.

Course Description

Created by the OpenEDG C++ Institute and delivered in partnership with the Cisco Networking Academy, *C Advanced* is **the subsequent progression from the "C Essentials" series**.

This advanced course delves into sophisticated C programming topics, equipping learners to tackle complex programming challenges and prepare for the CLP certification.

Through a blend of comprehensive **study materials**, **quizzes**, and **hands-on labs**, participants not only absorb theoretical knowledge but also apply these concepts to practical programming scenarios. By the course's conclusion, learners are thoroughly prepared for the CLP certification and possess the skills to excel in advanced programming roles.

Modules Overview:

- Module 1: Traces the development of the C language through key milestones (ANSI C, C89, C95, C99, C11), exploring obsolete yet valid elements and the evolution of function declarations. Dives into new C11 keywords like _Noreturn, _Alignof, _Alignas, _Bool, _Exit, _Complex, _Pragma, __func__, and _Generic.
- Module 2: Focuses on advanced techniques for handling a variable number of parameters, covering va_start(), va_arg(), va_end(), va_copy(), and functions like vsprintf(), vprintf(), vfprintf(), vscanf(), vsscanf(), vfscanf() to manage dynamic parameter lists effectively.
- Module 3: Explores low-level input/output operations, including the use of POSIX, API, ABI, WINAPI, etc., and functions like access(), open(), errno, close(), read(), write(), lseek(), dprintf(), stat(), symlink(), link(), readlink(), unlink(), fcntl(), ioctl() for direct system interactions.
- Module 4: Delivers insights into memory block manipulation and string handling techniques, including functions like strchr(), strrchr(), strstr(), strtok(), qsort(), bsearch(), and memory allocation methods (aligned_alloc(), calloc(), malloc(), realloc()), alongside internationalization (I18N) and Unicode considerations.
- Module 5: Covers definitions, implementations, and the history of processes and threads, emphasizing thread safety, system calls (system(), getenv(), setenv()), Unix and Windows process management (fork(), exit(), execxx(), wait(), CreateProcess(), WaitForSingleObject()), and threading models (POSIX threads, Windows threads, C11 threads).
- Module 6: Revisits floating-point and integer types in depth, discussing IEEE-754 standards, NaN, infinity, zero, the reliability of floating-point numbers, numerical

anomalies vs. precision, ULP, and the pragma directive, including FENV_ACCESS pragma for floating-point exceptions and rounding.
- Module 7: Introduces basic network programming concepts, TCP/IP and UDP protocols, connection-oriented and connectionless transmissions, endianness, socket addressing (IPv4, IPv6), and fundamental socket programming functions (getaddrinfo(), socket(), connect(), bind(), listen(), accept(), send(), recv()), along with client-server and peer-to-peer communication examples.
- Module 8: Addresses a variety of advanced topics including the distinction between const and volatile variables, the use and implications of goto, long jumps (setjmp(), longjmp()), advanced initialization techniques (static array indices, designated initializers, compound literals), and the importance of understanding sequence points, the asm keyword, and addressing portability issues and undefined behaviors.

Course Objectives

1. Analyze the Evolution and Standards of the C Language: Critically examine the historical progression and key milestones of the C language, including the evolution of function declarations and the impact of new C11 keywords, to understand their significance in modern C programming (Module 1).
2. Apply Techniques for Handling Variable Parameters: Utilize advanced techniques for managing variable numbers of parameters, demonstrating the ability to design flexible and dynamic C functions that cater to diverse programming needs (Module 2).
3. Evaluate Low-Level Input/Output Operations: Assess and implement low-level I/O operations using system calls and interfaces such as POSIX, enhancing direct interaction with system resources and understanding the implications of various I/O strategies (Module 3).
4. Synthesize Memory and String Manipulation Strategies: Integrate advanced memory management and string handling techniques, including the handling of internationalization and Unicode, to construct efficient and robust C applications (Module 4).
5. Compare Processes and Thread Management Across OS: Contrast and implement process and thread management techniques in Unix and Windows environments, demonstrating an understanding of threading models and thread safety to develop concurrent applications (Module 5).
6. Interpret Data Types and Numerical Precision: Examine the behavior of floats and integers in depth, including standards like IEEE-754 and concepts such as NaN and infinity, to develop a nuanced understanding of numerical precision and its implications in programming (Module 6).
7. Design Basic Networked Applications: Apply foundational network programming concepts to design and implement simple client-server and peer-to-peer applications, using TCP/IP and UDP protocols, and addressing socket programming challenges (Module 7).

8. Critique and Implement Advanced Programming Constructs: Evaluate the use of advanced programming constructs such as const and volatile variables, goto and long jumps, and explore the implications of sequence points and portability issues, aiming to improve code efficiency and maintainability (Module 8).
9. Create Sophisticated C Programs by Integrating Advanced Concepts: Demonstrate the ability to synthesize information from across the course to design sophisticated C programs that leverage advanced features and concepts, preparing for complex programming challenges and the CLP certification (All Modules).

Equipment Requirements

- Computer with Internet Connection: Essential for accessing online course materials and participating in course activities.
- Modern Web Browser: Required for optimal interaction with online course content.
- Integrated Development Environment (IDE): For certain labs and concepts, a local IDE setup is necessary. Although the course is accessible online, engaging with specific programming tasks mandates the use of an IDE installed on your local machine. Detailed guidance on selecting and setting up a suitable IDE, tailored to the advanced concepts covered, will be provided at the beginning of the course.

Course Outline

MODULE NUMBER AND NAME	RESOURCES	OBJECTIVES COVERED
Welcome to C Advanced	Course Structure, Certification Opportunities, and Getting Started with C Advanced	Discover the course framework, its advanced content, the pathway to the CLP certification, and the necessary tools for this advanced learning journey. Understand how to navigate the course effectively, leveraging the provided resources to build upon the knowledge gained from previous C programming courses.
Module 1: Evolution of C – From Past to Eternity	Study Pages, Labs, and Module 1 Test	Master the historical milestones and evolution of the C language, including the introduction and impact of new C11 keywords. This module provides a comprehensive understanding of how the C language has grown and how its past influences current programming practices.

Module 2: Handling Variable Number of Parameters	Study Pages, Labs, and Module 2 Test	Learn to adeptly manage functions with a variable number of parameters, enhancing your ability to write versatile and dynamic C code. This module equips you with the skills to implement advanced parameter handling techniques effectively.
Module 3: Low-Level IO Operations	Study Pages, Labs, and Module 3 Test	Gain proficiency in performing low-level input/output operations, a crucial skill for direct system interaction and manipulation in C programming. This module focuses on using system calls and interfaces for efficient IO operations.
Module 4: Memory and Strings	Study Pages, Labs, and Module 4 Test	Deepen your understanding of advanced memory management and string handling, including strategies for dealing with internationalization and Unicode in C programs. This module aims to enhance your ability to manipulate memory and strings with high precision.
Module 5: Processes and Threads	Study Pages, Labs, and Module 5 Test	Explore the complex world of processes and threading in C programming, understanding how to manage concurrent processes and threads across different operating systems. This module prepares you for advanced programming tasks involving multi-threading and process management.
Module 6: Floats and Ints Revisited	Study Pages, Labs, and Module 6 Test	Revisit and explore in-depth the behaviors of floating-point and integer data types, focusing on standards, precision, and anomalies. This module enhances your numerical programming skills

		in C, preparing you for complex computational tasks.
Module 7: Network Sockets – The Absolute Basics	Study Pages, Labs, and Module 7 Test	Introduce yourself to the fundamentals of network programming, covering the basics of TCP/IP, UDP, and socket programming. This module lays the foundation for developing networked applications in C.
Module 8: Miscellaneous Advanced Topics	Study Pages, Labs, and Module 8 Test	Address a range of advanced programming concepts, including the use of *const* and *volatile*, understanding *goto*, managing long jumps, and tackling portability issues. This module aims to round off your advanced C programming education, preparing you for a variety of complex programming challenges.
Course Completion	C Advanced – Final Test (Score 70% or more to qualify for a 50% discount on the CLP exam) + Become CLP certified (Paid Option)	Upon successfully completing the C Advanced final test, you will have demonstrated a thorough understanding and application of advanced C programming concepts. This achievement showcases your ability to integrate and apply complex programming solutions, setting you up for success in the CLP certification exam and enabling you to tackle sophisticated challenges in software development, network programming, and system development.

Why Continue Learning C with Advanced C

Delving deeper into C programming through the *C Advanced* course propels your capabilities beyond the fundamentals, positioning you as a highly skilled professional across a spectrum of technological domains. This advanced knowledge equips you for complex challenges in software development, network systems, embedded technologies, and operating system architecture, among other areas.

Achieving the *CLP – C Certified Professional Programmer* certification signifies a deep commitment to excellence in C programming. It not only validates your advanced skill set but also unlocks potential for significant career progression and engagement with innovative technology projects. The certification acts as a testament to your proficiency in handling sophisticated programming tasks, making you a valuable asset in the ever-evolving tech landscape.