uCertify

Course Outline

C Primer Plus

20 May 2024

1. Course Objective
2. Exercises, Quizzes, Flashcards & Glossary

   Number of Questions
3. Expert Instructor-Led Training
4. ADA Compliant & JAWS Compatible Platform
5. State of the Art Educator Tools
6. Award Winning Learning Platform (LMS)
7. Chapter & Lessons

   Syllabus

   Chapter 1: Introduction

   Chapter 2: Getting Ready

   Chapter 3: Introducing C

   Chapter 4: Data and C

   Chapter 5: Character Strings and Formatted Input/Output

   Chapter 6: Operators, Expressions, and Statements

   Chapter 7: C Control Statements: Looping

   Chapter 8: C Control Statements: Branching and Jumps

   Chapter 9: Character Input/Output and Input Validation

   Chapter 10: Functions

   Chapter 11: Arrays and Pointers

   Chapter 12: Character Strings and String Functions

   Chapter 13: Storage Classes, Linkage, and Memory Management

   Chapter 14: File Input/Output

   Chapter 15: Structures and Other Data Forms

   Chapter 16: Bit Fiddling

   Chapter 17: The C Preprocessor and the C Library

   Chapter 18: Advanced Data Representation

   Chapter 19: Appendix A: Reference Section

Videos and How To

8. Performance Based labs

    Lab Tasks

    Here's what you get

1. Course Objective

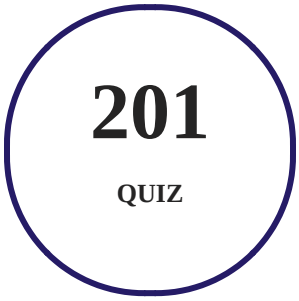
Master the fundamentals of C programming with the C Primer Plus course! This course provides a friendly and instructive approach, featuring interactive lessons, quizzes, and hands-on labs. Explore the evolution of the C language, gain a clear understanding of programming concepts, and enhance your skills through practical examples and exercises. Whether you're a beginner or seeking to reinforce your knowledge, this newest edition will serve as an enjoyable and effective introduction to the world of C programming.

2. Exercises

There is no limit to the number of times learners can attempt these. Exercises come with detailed remediation, which ensures that learners are confident on the topic before proceeding.

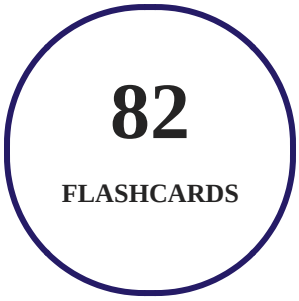
3. Quiz

Quizzes test your knowledge on the topics of the exam when you go through the course material. There is no limit to the number of times you can attempt it.



4. flashcards

Flashcards are effective memory-aiding tools that help you learn complex topics easily. The flashcard will help you in memorizing definitions, terminologies, key concepts, and more. There is no limit to the number of times learners can attempt these. Flashcards help master the key concepts.



5. Glossary of terms

uCertify provides detailed explanations of concepts relevant to the course through Glossary. It contains a list of frequently used terminologies along with its detailed explanation. Glossary defines the key terms.

6. Expert Instructor-Led Training

uCertify uses the content from the finest publishers and only the IT industry's finest instructors. They have a minimum of 15 years real-world experience and are subject matter experts in their fields. Unlike a live class, you can study at your own pace. This creates a personal learning experience and gives you all the benefit of hands-on training with the flexibility of doing it around your schedule 24/7.

7.  ADA Compliant & JAWS Compatible Platform

uCertify course and labs are ADA (Americans with Disability Act) compliant. It is now more accessible to students with features such as:

- Change the font, size, and color of the content of the course
- Text-to-speech, reads the text into spoken words
- Interactive videos, how-tos videos come with transcripts and voice-over
- Interactive transcripts, each word is clickable. Students can clip a specific part of the video by clicking on a word or a portion of the text.

JAWS (Job Access with Speech) is a computer screen reader program for Microsoft Windows that reads the screen either with a text-to-speech output or by a Refreshable Braille display. Student can easily navigate uCertify course using JAWS shortcut keys.

8.  State of the Art Educator Tools

uCertify knows the importance of instructors and provide tools to help them do their job effectively. Instructors are able to clone and customize course. Do ability grouping. Create sections. Design grade scale and grade formula. Create and schedule assessments. Educators can also move a student from self-paced to mentor-guided to instructor-led mode in three clicks.

9.  Award Winning Learning Platform (LMS)

uCertify has developed an award winning, highly interactive yet simple to use platform. The SIIA CODiE Awards is the only peer-reviewed program to showcase business and education technology's finest products and services. Since 1986, thousands of products, services and solutions have been recognized for achieving excellence. uCertify has won CODiE awards consecutively for last 7 years:

- **2014**
    1. Best Postsecondary Learning Solution

- **2015**
    1. Best Education Solution
    2. Best Virtual Learning Solution
    3. Best Student Assessment Solution
    4. Best Postsecondary Learning Solution
    5. Best Career and Workforce Readiness Solution
    6. Best Instructional Solution in Other Curriculum Areas
    7. Best Corporate Learning/Workforce Development Solution

- **2016**
    1. Best Virtual Learning Solution
    2. Best Education Cloud-based Solution
    3. Best College and Career Readiness Solution
    4. Best Corporate / Workforce Learning Solution
    5. Best Postsecondary Learning Content Solution
    6. Best Postsecondary LMS or Learning Platform
    7. Best Learning Relationship Management Solution

- **2017**
    1. Best Overall Education Solution
    2. Best Student Assessment Solution
    3. Best Corporate/Workforce Learning Solution
    4. Best Higher Education LMS or Learning Platform

- **2018**
    1. Best Higher Education LMS or Learning Platform
    2. Best Instructional Solution in Other Curriculum Areas
    3. Best Learning Relationship Management Solution

- **2019**
    1. Best Virtual Learning Solution
    2. Best Content Authoring Development or Curation Solution
    3. Best Higher Education Learning Management Solution (LMS)

- **2020**
    1. Best College and Career Readiness Solution
    2. Best Cross-Curricular Solution
    3. Best Virtual Learning Solution

10. Chapter & Lessons

uCertify brings these textbooks to life. It is full of interactive activities that keeps the learner engaged. uCertify brings all available learning resources for a topic in one place so that the learner can efficiently learn without going to multiple places. Challenge questions are also embedded in the chapters so learners can attempt those while they are learning about that particular topic. This helps them grasp the concepts better because they can go over it again right away which improves learning.

Learners can do Flashcards, Exercises, Quizzes and Labs related to each chapter. At the end of every lesson, uCertify courses guide the learners on the path they should follow.

Syllabus

- What Computers Do
- High-level Computer Languages and Compilers
- Language Standards
- Using C: Seven Steps
- Programming Mechanics
- How This Course Is Organized
- Conventions Used in This Course
- Summary
- Review Questions
- Programming Exercise

Chapter 3: Introducing C

- A Simple Example of C
- The Example Explained
- The Structure of a Simple Program
- Tips on Making Your Programs Readable
- Taking Another Step in Using C
- While You’re at It—Multiple Functions

- Introducing Debugging
- Keywords and Reserved Identifiers
- Key Concepts
- Summary
- Review Questions
- Programming Exercises

Chapter 4: Data and C

- A Sample Program
- Data Variables and Constants
- Data: Data-Type Keywords
- Basic C Data Types
- Using Data Types
- Arguments and Pitfalls
- One More Example: Escape Sequences
- Key Concepts
- Summary
- Review Questions

- Programming Exercises

Chapter 5: Character Strings and Formatted Input/Output

- Introductory Program
- Character Strings: An Introduction
- Constants and the C Preprocessor
- Exploring and Exploiting printf() and scanf()
- Key Concepts
- Summary
- Review Questions
- Programming Exercises

Chapter 6: Operators, Expressions, and Statements

- Introducing Loops
- Fundamental Operators
- Some Additional Operators
- Expressions and Statements
- Type Conversions
- Function with Arguments

- A Sample Program
- Key Concepts
- Summary
- Review Questions
- Programming Exercises

Chapter 7: C Control Statements: Looping

- Revisiting the while Loop
- The while Statement
- Which Is Bigger: Using Relational Operators and Expressions
- Indefinite Loops and Counting Loops
- The for Loop
- More Assignment Operators: >+=, -=, *=, /=, %=
- The Comma Operator
- An Exit-Condition Loop: do while
- Which Loop?
- Nested Loops
- Introducing Arrays

- A Loop Example Using a Function Return Value
- Key Concepts
- Summary
- Review Questions
- Programming Exercises

Chapter 8: C Control Statements: Branching and Jumps

- The if Statement
- Adding else to the if Statement
- Let’s Get Logical
- A Word-Count Program
- The Conditional Operator: ?:
- Loop Aids: continue and break
- Multiple Choice: switch and break
- The goto Statement
- Key Concepts
- Summary
- Review Questions

- Programming Exercises

Chapter 9: Character Input/Output and Input Validation

- Single-Character I/O: getchar() and putchar()
- Buffers
- Terminating Keyboard Input
- Redirection and Files
- Creating a Friendlier User Interface
- Input Validation
- Menu Browsing
- Key Concepts
- Summary
- Review Questions
- Programming Exercises

Chapter 10: Functions

- Reviewing Functions
- ANSI C Function Prototyping

- Recursion
- Compiling Programs with Two or More Source Code Files
- Finding Addresses: The & Operator
- Altering Variables in the Calling Function
- Pointers: A First Look
- Key Concepts
- Summary
- Review Questions
- Programming Exercises

Chapter 11: Arrays and Pointers

- Arrays
- Multidimensional Arrays
- Pointers and Arrays
- Functions, Arrays, and Pointers
- Pointer Operations
- Protecting Array Contents
- Pointers and Multidimensional Arrays

- Variable-Length Arrays (VLAs)
- Compound Literals
- Key Concepts
- Summary
- Review Questions
- Programming Exercises

Chapter 12: Character Strings and String Functions

- Representing Strings and String I/O
- String Input
- String Output
- The Do-It-Yourself Option
- String Functions
- A String Example: Sorting Strings
- The ctype.h Character Functions and Strings
- Command-Line Arguments
- String-to-Number Conversions
- Key Concepts

- Summary
- Review Questions
- Programming Exercises

Chapter 13: Storage Classes, Linkage, and Memory Management

- Storage Classes
- A Random-Number Function and a Static Variable
- Roll ’Em
- Allocated Memory: malloc() and free()
- ANSI C Type Qualifiers
- Key Concepts
- Summary
- Review Questions
- Programming Exercises

Chapter 14: File Input/Output

- Communicating with Files
- Standard I/O
- A Simple-Minded File-Condensing Program

- File I/O: fprintf(), fscanf(), fgets(), and fputs()
- Adventures in Random Access: fseek() and ftell()
- Behind the Scenes with Standard I/O
- Other Standard I/O Functions
- Key Concepts
- Summary
- Review Questions
- Programming Exercises

Chapter 15: Structures and Other Data Forms

- Sample Problem: Creating an Inventory of Books
- Setting Up the Structure Declaration
- Defining a Structure Variable
- Arrays of Structures
- Nested Structures
- Pointers to Structures
- Telling Functions About Structures
- Saving the Structure Contents in a File

- Structures: What Next?
- Unions: A Quick Look
- Enumerated Types
- typedef: A Quick Look
- Fancy Declarations
- Functions and Pointers
- Key Concepts
- Summary
- Review Questions
- Programming Exercises

Chapter 16: Bit Fiddling

- Binary Numbers, Bits, and Bytes
- Other Number Bases
- C’s Bitwise Operators
- Bit Fields
- Alignment Features (C11)
- Key Concepts

- Summary
- Review Questions
- Programming Exercises

Chapter 17: The C Preprocessor and the C Library

- First Steps in Translating a Program
- Manifest Constants: #define
- Using Arguments with #define
- Macro or Function?
- File Inclusion: #include
- Other Directives
- Inline Functions (C99)
- _Noreturn Functions (C11)
- The C Library
- The Math Library
- The General Utilities Library
- The Assert Library
- memcpy() and memmove() from the string.h Library

- Programming Exercises

Chapter 19: Appendix A: Reference Section

- Section I: Additional Reading
- Section II: C Operators
- Section III: Basic Types and Storage Classes
- Section IV: Expressions, Statements, and Program Flow
- Section V: The Standard ANSI C Library with C99 and C11 Additions
- Section VI: Extended Integer Types
- Section VII: Expanded Character Support
- Section VIII: C99/C11 Numeric Computational Enhancements
- Section IX: Differences Between C and C++

11. Performance Based Labs

uCertify's performance-based labs are simulators that provides virtual environment. Labs deliver hands on experience with minimal risk and thus replace expensive physical labs. uCertify Labs are cloud-based, device-enabled and can be easily integrated with an LMS. Features of uCertify labs:

- Provide hands-on experience in a safe, online environment
- Labs simulate real world, hardware, software & CLI environment
- Flexible and inexpensive alternative to physical Labs

- Comes with well-organized component library for every task
- Highly interactive - learn by doing
- Explanations and remediation available
- Videos on how to perform

Lab Tasks

- Using printf() and scanf()
- Using Multiple Functions
- Displaying the Value of a Variable
- Understanding Floating-Point Conversions
- Converting an Integer to its ASCII Code
- Performing String Formatting
- Using Strings and Unit Conversion
- Understanding Unit Conversion
- Using Functions with Arguments
- Using Operators
- Performing Calculations Using a Function Return Value
- Using Character Arrays
- Using Nested Loops
- Using the switch Statement
- Using Multiple Choice else if Statement
- Using the if Statement
- Understanding Character Input/Output
- Using the ctype.h Library
- Using a Loop to Calculate Fibonacci Numbers
- Computing the Harmonic Mean of Two Numbers
- Using Multidimensional Arrays
- Performing Calculations on Multiple Arrays
- Using Single-Dimensional Arrays
- Using String Input
- Understanding Memory Allocation
- Writing and Reading Numbers from a File

- Understanding Nested Structures
- Passing Structure Members as Arguments
- Using Compound Literals and Structures
- Using Bitwise Operators
- Understanding Binary Conversion
- Using Macro Functions
- Demonstrating the Usage of the atexit() Function
- Displaying a String in Reverse Order Using Stacks

Here's what you get

34

PERFORMANCE BASED LAB

GET IN TOUCH:

3187 Independence Drive
Livermore, CA 94551,
United States

+1-415-763-6300

support@ucertify.com

www.ucertify.com