CLA: Programming Essentials in C

Scope and Sequence

Last updated January 18, 2017

Scope and Sequence - Table of Contents

Target Audience

The *CLA: Programming Essentials in C* curriculum is designed for students who want to learn the fundamentals of programming using the C language.

Prerequisites

There are no prerequisites for this course.

Target Certification

The *CLA: Programming Essentials in C* curriculum helps students prepare for the *CLA – C Programming Language Certified Associate* certification exam. *C Programming Language Certified Associate* (CLA) is a professional certification that measures the ability to accomplish coding tasks related to the basics of programming in the C language and the fundamental notions, as well as fundamental programming techniques, customs and vocabulary, including the most common library functions and the usage of the preprocessor.

Curriculum Description

This course covers all the basics of programming in the C programming language, including the syntax and semantics of the C language as well as data types offered by the language.

The course is broken down into 9 modules:

- Module 0: explains the process of installing and using the programming environment.
- Module 1: introduces common computer programming concepts, e.g. integers and variables.
- Module 2: introduces the concepts of data types and flow control.
- Module 3: further discusses the concepts of flow control, introducing more data types and logic in computer science.
- Module 4: touches on the subject of aggregating data into arrays.
- Module 5: explains the differences between arrays and structures.
- Module 6: introduces the subject of functions.
- Module 7: discusses files and streams.
- Module 8: elaborates on the subject of the pre-processor and declarations.

Each student has access to hands-on practice materials, quizzes and assessments to learn how to utilize the skills and knowledge gained on the course and interact with some real-life programming tasks and situations.

Curriculum Objectives

The aim of the course is to:

- familiarize the student with the universal concepts of computer programming,
- present the syntax, semantics and basic data types of the C language,
- discuss the customs and vocabulary of the C language, including the most common library functions and the usage of the pre-processor
- align the course to the C++ Institute CLA – C Programming Language Certified Associate certification.

During the course, students will study the following objectives:

- Introduction to compiling and software development,
- Basic scalar data types and their operators,
- Flow control,

- Complex data types: arrays, structures and pointers,
- Memory management,
- Functions,
- Files and streams,
- Structuring the code: functions and modules,
- Preprocessor directives and complex declarations.

Course Outline

Learning Module	CLA – C Programming Language Certified Associate Certification Objectives Covered
0 – Installing and using your programming environment	• introduction to compiling and software development.
1 – Introduction to computer programming	• languages: natural and artificial, • machine languages, • high-level programming languages, • obtaining the machine code: compilation process, • writing simple programs, • variables, • integer values in real life and in C, • integer literals.
2 – Data types	• floating point values in real life and in C, • float literals, • arithmetic operators, • priority and binding, • post- and pre-incrementation and decrementation, • operators of type op=, • char type and ASCII code, • char literals, • equivalence of int and char data, • comparison operators, • conditional execution and if keyword, • printf() and scanf() functions.
3 – Flow control	• conditional execution: the “else” branch, • integer and float types, • conversions, • typecast and its operators, • loops – while, do and for, • controlling the loop execution – break and continue, • logical and bitwise operators.
4 – Arrays	• switch: different faces of ‘if’, • arrays (vectors), • sorting in real life and in a computer memory, • initiators, • pointers, • an address, a reference, a dereference and the sizeof operator, • simple pointer and pointer to nothing (NULL), • & operator, • pointers arithmetic,

	• pointers vs. arrays: different forms of the same phenomenon, • using strings, • basic functions dedicated to string manipulation.
5 – Memory management and structures	• array indexing, • the usage of pointers: perils and disadvantages, • void type, • arrays of arrays and multidimensional arrays, • memory allocation and deallocation: malloc() and free() functions, • arrays of pointers vs. multidimensional arrays, • structures, • declaring, using and initializing structures, • pointers to structures and arrays of structures, • basics of recursive data collections.
6 – Functions	• functions, • how to declare, define and invoke a function, • variables' scope, local variables and function parameters, • pointers, arrays and structures as function parameters, • function result and return statement, • void as a parameter, pointer and result, • parameterizing the main function, • external function and the extern declarator, • header files and their role.
7 – Files and streams	• files vs. streams, • header files needed for stream operations, • FILE structure, • opening and closing a stream, open modes, errno variable, • reading and writing to/from a stream, • predefined streams: stdin, stdout and stderr, • stream manipulation: fgetc(), fputc(), fgets() and fputs() functions, • raw input/output: fread() and fwrite() functions.
8 – Preprocessor and complex declarations	• preprocessor, • #include: how to make use of a header file, • #define: simple and parameterized macros, • #undef directive, • predefined preprocessor symbols, • macrooperators: # and ##, • conditional compilation: #if and #ifdef directives, • avoiding multiple compilations of the same header files, • scopes of declarations, storage classes, • user –defined types, • pointers to functions, • analyzing and creating complex declarations.

Minimum System Requirements

The course content modules, labs, quizzes and assessments can be accessed online through any Internet browser. For the best learning experience, we recommend using the most recent versions of Mozilla Firefox or Internet Explorer/Microsoft Edge.

Industry certification

The course curriculum helps students prepare for the C++ Institute *CLA – C Programming Language Certified Associate* certification.

A Statement of Achievement will be issued to participants who successfully complete the *CLA: Programming Essentials in C* course. The Statement of Achievement will acknowledge that the individual has completed the course and is now ready to attempt the qualification *CLA – C Programming Language Certified Associate Certification*, taken through Pearson VUE computer-based testing, at a 51% discount.

To receive the Statement of Achievement, instructors must mark the student as having successfully passed the course.

For additional information about the *C++ Institute CLA – C Programming Language Certified Associate certification*, please visit www.cppinstitute.org/certification.