

Learn C++



LearnCpp.com is a free website devoted to teaching you how to program in modern C++. The lessons on this site will walk you through all the steps needed to write, compile, and debug your C++ programs. No prior programming experience is necessary, but programmers of all levels will benefit from our best practices, tips, and insights.

Becoming an expert won't happen overnight, but with a bit of patience, you'll get there. And LearnCpp.com will show you the way.

Chapter 2

C++ Basics: Functions and Files

Chapter 3

Debugging C++ Programs

Chapter 4

Fundamental Data Types

4.1

[Introduction to fundamental data types](#)

4.2

[Void](#)

4.3

[Object sizes and the sizeof operator](#)

4.4

[Signed integers](#)

4.5

[Unsigned integers, and why to avoid them](#)

4.6

[Fixed-width integers and size_t](#)

4.7

[Introduction to scientific notation](#)

4.8

[Floating point numbers](#)

4.9

[Boolean values](#)

4.10

Introduction to if statements

4.11

Chars

4.12

Introduction to type conversion and static_cast

4.x

Chapter 4 summary and quiz

Chapter 5

Constants and Strings

5.1

Constant variables (named constants)

5.2

Literals

Moved

5.3

N numeral systems (decimal, binary, hexadecimal, and octal)

Moved

5.4

Constant expressions and compile-time optimization

Moved

5.5

Constexpr variables

Split

5.6

The conditional operator

Split

5.7

Inline functions and variables

Updated

5.8

Constexpr and constexpr functions

Updated

5.9

Introduction to std::string

5.10

Introduction to std::string_view

5.11

std::string_view (part 2)

5.x

Chapter 5 summary and quiz

Updated

Chapter 6

Operators

6.1

Operator precedence and associativity

6.2

Arithmetic operators

6.3

Remainder and Exponentiation

6.4

Increment/decrement operators, and side effects

6.5

The comma operator

6.6

Relational operators and floating point comparisons

6.7

Logical operators

6.x

Chapter 6 summary and quiz

Chapter O

Bit Manipulation (optional chapter)

O.1

Bit flags and bit manipulation via std::bitset

O.2

Bitwise operators

O.3

Bit manipulation with bitwise operators and bit masks

O.4

Converting integers between binary and decimal representation

Chapter 7

Scope, Duration, and Linkage

7.1

Compound statements (blocks)

7.2

User-defined namespaces and the scope resolution operator

7.3

Local variables

7.4

Introduction to global variables

7.5

Variable shadowing (name hiding)

7.6

Internal linkage

7.7

External linkage and variable forward declarations

7.8

Why (non-const) global variables are evil

7.9

Sharing global constants across multiple files (using inline variables)

7.10

Static local variables

7.11

Scope, duration, and linkage summary

7.12

Using declarations and using directives

7.13

Unnamed and inline namespaces

7.x

Chapter 7 summary and quiz

Chapter 8

Control Flow

8.1

Control flow introduction

8.2

If statements and blocks

8.3

Common if statement problems

8.4

Constexpr if statements

8.5

Switch statement basics

8.6

Switch fallthrough and scoping

8.7

Goto statements

8.8

Introduction to loops and while statements

8.9

Do while statements

8.10

For statements

8.11

Break and continue

8.12

Halts (exiting your program early)

8.13

Introduction to random number generation

Moved

8.14

Generating random numbers using Mersenne Twister

Moved

8.15

Global random numbers (Random.h)

Split

8.x

Chapter 8 summary and quiz

Chapter 9

Error Detection and Handling

9.1

Introduction to testing your code

9.2

Code coverage

9.3

Common semantic errors in C++

9.4

Detecting and handling errors

9.5

std::cin and handling invalid input

9.6

Assert and static_assert

9.x

Chapter 9 summary and quiz

Chapter 10

Type Conversion, Type Aliases, and Type Deduction

10.1

Implicit type conversion

10.2

Floating-point and integral promotion

10.3

Numeric conversions

10.4

Narrowing conversions, list initialization, and constexpr initializers

10.5

Arithmetic conversions

10.6

Explicit type conversion (casting) and static_cast

10.7

Typedefs and type aliases

10.8

Type deduction for objects using the auto keyword

10.9

Type deduction for functions

10.x

Chapter 10 summary and quiz

Chapter 11

Function Overloading and Function Templates

11.1

Introduction to function overloading

11.2

Function overload differentiation

11.3

Function overload resolution and ambiguous matches

11.4

Deleting functions

New

11.5

Default arguments

11.6

Function templates

11.7

Function template instantiation

11.8

Function templates with multiple template types

11.9

Non-type template parameters

11.x

Chapter 11 summary and quiz

Chapter 12

Compound Types: References and Pointers

12.1

Introduction to compound data types

12.2

Value categories (lvalues and rvalues)

12.3

Lvalue references

12.4

Lvalue references to const

12.5

Pass by lvalue reference

12.6

Pass by const lvalue reference

12.7

Introduction to pointers

12.8

Null pointers

12.9

Pointers and const

12.10

Pass by address

12.11

Pass by address (part 2)

12.12

Return by reference and return by address

12.13

In and out parameters

12.14

Type deduction with pointers, references, and const

12.15

std::optional

12.x

Chapter 12 summary and quiz

Chapter 13

Compound Types: Enums and Structs

13.1

Introduction to program-defined (user-defined) types

13.2

Unscoped enumerations

13.3

Unscoped enumerator integral conversions

13.4

Converting an enumeration to and from a string

13.5

Introduction to overloading the I/O operators

13.6

Scoped enumerations (enum classes)

13.7

Introduction to structs, members, and member selection

13.8

Struct aggregate initialization

13.9

Default member initialization

13.10

Passing and returning structs

13.11

Struct miscellany

13.12

Member selection with pointers and references

13.13

Class templates

13.14

Class template argument deduction (CTAD) and deduction guides

13.15

Alias templates

13.x

Chapter 13 summary and quiz

13.y

Using a language reference

Chapter 14

Introduction to Classes

14.1

Introduction to object-oriented programming

Updated

14.2

Introduction to classes

Updated

14.3

Member functions

Updated

14.4

Const class objects and const member functions

Moved

14.5

Public and private members and access specifiers

Updated

14.6

Access functions

Updated

14.7

Member functions returning references to data members

Split

14.8

The benefits of data hiding (encapsulation)

Updated

14.9

Introduction to constructors

Updated

14.10

Constructor member initializer lists

Updated

14.11

Default constructors and default arguments

Updated

14.12

Delegating constructors

Updated

14.13

Temporary class objects

Updated

14.14

Introduction to the copy constructor

Updated

14.15

Class initialization and copy elision

Updated

14.16

Converting constructors and the explicit keyword

Updated

14.x

Chapter 14 summary and quiz

Updated

Chapter 15

More on Classes

15.1

The hidden “this” pointer and member function chaining

Updated

15.2

Classes and header files

Updated

15.3

Nested types (member types)

Updated

15.4

Introduction to destructors

Updated

15.5

Class templates with member functions

Updated

15.6

Static member variables

Updated

15.7

Static member functions

Updated

15.8

Friend non-member functions

Updated

15.9

Friend classes and friend member functions

Updated

15.10

Ref qualifiers

Split

15.x

Chapter 15 summary and quiz

Updated

Chapter 16

Dynamic arrays: `std::vector`

16.1

Introduction to containers and arrays

New

16.2

Introduction to `std::vector` and list constructors

Updated

16.3

`std::vector` and the unsigned length and subscript problem

New

16.4

Passing `std::vector`

New

16.5

Returning `std::vector`, and an introduction to move semantics

Split

16.6

Arrays and loops

Updated

16.7

Arrays, loops, and sign challenge solutions

New

16.8

Range-based for loops (for-each)

Updated

16.9

Array indexing and length using enumerators

Updated

16.10

std::vector resizing and capacity

Updated

16.11

std::vector and stack behavior

Updated

16.12

std::vector<bool>

New

16.x

Chapter 16 summary and quiz

Updated

Chapter 17

Fixed-size arrays: std::array and C-style arrays

17.1

Introduction to std::array

Updated

17.2

std::array length and indexing

New

17.3

Passing and returning std::array

New

17.4

std::array of class types, and brace elision

Updated

17.5

Arrays of references via std::reference_wrapper

New

17.6

std::array and enumerations

New

17.7

Introduction to C-style arrays

Updated

17.8

[C-style array decay](#)

Updated

17.9

[Pointer arithmetic and subscripting](#)

Updated

17.10

[C-style strings](#)

Updated

17.11

[C-style string symbolic constants](#)

Updated

17.12

[Multidimensional C-style Arrays](#)

Updated

17.13

[Multidimensional std::array](#)

New

17.x

[Chapter 17 summary and quiz](#)

Updated

Chapter 18

Iterators and Algorithms (under construction)

18.1

[Sorting an array using selection sort](#)

18.2

[Introduction to iterators](#)

18.3

[Introduction to standard library algorithms](#)

18.4

[Timing your code](#)

Chapter 19

Dynamic Allocation (under construction)

19.1

[Dynamic memory allocation with new and delete](#)

19.2

[Dynamically allocating arrays](#)

19.3

[Destructors](#)

19.4

[Pointers to pointers and dynamic multidimensional arrays](#)

19.5

[Void pointers](#)

Chapter 20

Functions

20.1

Function Pointers

20.2

The stack and the heap

20.3

Recursion

20.4

Command line arguments

20.5

Ellipsis (and why to avoid them)

20.6

Introduction to lambdas (anonymous functions)

20.7

Lambda captures

20.x

Chapter 20 summary and quiz

Chapter 21

Operator Overloading

21.1

Introduction to operator overloading

21.2

Overloading the arithmetic operators using friend functions

21.3

Overloading operators using normal functions

21.4

Overloading the I/O operators

21.5

Overloading operators using member functions

21.6

Overloading unary operators +, -, and !

21.7

Overloading the comparison operators

21.8

Overloading the increment and decrement operators

21.9

Overloading the subscript operator

21.10

Overloading the parenthesis operator

21.11

Overloading typecasts

21.12

Overloading the assignment operator

21.13

Shallow vs. deep copying

21.14

Overloading operators and function templates

21.x

Chapter 21 summary and quiz

21.y

Chapter 21 project

Chapter 22

Move Semantics and Smart Pointers

Chapter 23

Object Relationships

Appendix C

The End

C.1

The end?