

CS107: C++ PROGRAMMING

Log in or Sign up to track your course progress, gain access to final exams, and get a free certificate of completion!

[↑ Back to 'Study Guides and Review Exercises'](#)

Unit 4 Study Guide and Review: Advanced Concepts

4a. Write class and function templates

1. What is a template and what is it used for?
2. How do you code a template?

A template is used to allow a variety of different data types to be used instead of restricting the data type to a single type. Instead of creating a method that only receives an integer, you can create a method that accepts any data type, by using a template. This video explains the purpose and meaning of a template.

These slides provide examples of how to code a template.

4b. Code with a class that manipulates the files

1. What is the difference between character I/O and formatting I/O?
2. How do you code input and output?

Depending on the purpose of your input and output will determine the methods that you utilize. Whether you use `getChar()` and `putChar()`, or `printf()` and `scanf()` is dependent on whether you are reading and writing a file, one character at a time or all at once. To explore these options, review this article

This video shows a great example of how to write the code for reading from a file. To explore how to write to a file, review this video.

4c. Use namespaces and exceptions in C++ code

1. What is a namespace and what is its purpose?
2. What is an exception and what is its purpose and when is it used?

Namespaces are declared regions that define scope. The purpose of a namespace is to organize code into logical groups. You can create your own namespaces, which would then allow you to organize your content into a logical scope. For more information on how to use namespaces, watch this video

An exception is an error that is thrown when specific conditions are met that prevent processing to continue. Many classes throw an exception when a problem occurs. An exception allows the processing to continue or allow the program to exit gracefully instead of causing a run-time error. If you run your program and it throws an error, perhaps you should create a try catch that catches this error, while also seeking to find the reason for the error. For further information, review this article.

4d. Use recursive functions

1. What is a recursive function and how is it used?

A recursive function is a function that calls itself as many times as necessary until a specific condition is met. It behaves much like a loop in that it repeats itself, but is not usually part of a loop structure. A popular use of the recursive method is the application of the fibonacci number. A fibonacci number is the sum of the two numbers before it: 1 2 3 5 8 13....Each iteration through the loop captures the sum of the previous two numbers and then returns and repeats the process. For more information about recursion, read this page.

4e. Write preprocessor instructions

1. What are the different types of preprocessor functions and when are each type used?

The preprocessors consist of the directives, which give instruction to the compiler to preprocess the information before actual compilation starts. Start all preprocessor directives with a #, which should appear at the top of the page, with only blank lines in front of it. Preprocessor directives are not C++ statements, so they do not end in a semicolon (;). You already have seen a **#include** directive in all the examples. This macro is used to include a header file into the source file. There are number of preprocessor directives supported by C++ like #include, #define, #if, #else, #line, etc. For a list of preprocessor commands, review this article. Be sure to know the different types and under what conditions they are used..

Unit 4 Vocabulary

This vocabulary list includes terms that might help you with the review items above and some terms you should be familiar with to be successful in completing the final exam for the course.

Try to think of the reason why each term is included.

- Catch block
- Exception
- Exception handling
- Namespace
- Pre-processor
- Raised exception
- Recursion
- Scope
- Standard error
- Standard input
- Standard output

- Streams
- Template
- Thrown
- Try block

Last modified: Wednesday, December 7, 2016, 2:31 PM