C++ Syllabus

Contact Info

While going through the program, if you have questions about anything, you can reach us at support@udacity.com. For help from Udacity Mentors and your peers visit the Udacity Classroom.

Nanodegree Program Info

Get hands-on experience by coding five real-world projects. Learn to build a route planner using OpenStreetMap data, write a process monitor for your computer, and implement your own smart pointers. Finally, showcase all your newfound skills by building a chatbot and coding your own C++ application.

Required Software

- Jupyter notebook 6.0.1 or latest
- SDL V2.0
- Visual Studio Code
- C++ Extension
- Clang-format
- ncurses library
- Make tool
- TensorFlow C++ API

Version: 1.0.0

Length of Program: 104 Days*

* This is a self-paced program and the length is an estimation of total hours the average student may take to complete all required coursework, including lecture and project time. Actual hours may vary.

Part 1: Welcome

Welcome to the program! In this part, you'll get an orientation into using our classroom and services.

Learn basic C++ syntax, functions, containers, and compiling and linking with multiple files.

Project: Build an OpenStreetMap Route Planner

In this project you will extend the IO2D map display code to use A*, so your program will be able to find a path between two points on the map.

Supporting Lessons

Lesson	Summary
Welcome	Meet your instructors, David and Stephen, and hear from the inventor of C++, Bjarne Stroustrup.
Introduction to the C++ Language	In this lesson, you will build on your previous programming experience to learn the basics of the C++ language.
A* Search	In this lesson, you will learn about the A* search algorithm. Over a sequence of exercises, you will use your A* search implementation to plan a path through the obstacles in the ASCII board.
Writing Multifile Programs	In this lesson, you will learn the syntax for C++ language features that you will need for the next steps in the course. This includes an overview of header files, pointers, build tools, and classes.

Part 3: Object-Oriented Programming

Explore Object-Oriented Programming (OOP) in C++ with examples and exercises covering the essentials of OOP like abstraction and inheritance all the way through to advanced topics like polymorphism and templates.

Project: System Monitor

Time to build the project! In this lesson, you'll get the tools you need to build the project for this course, a system monitor application similar to htop!

Supporting Lessons

Lesson	Summary
Welcome	
Intro to OOP	In this lesson, you will explore some of the basic object oriented functionality of the C++ language.
Advanced OOP	In this lesson, we'll get into some of the more advanced topics in object oriented programming, including inheritance, polymorphism and templates.

Part 4: Memory Management

Discover the power of memory management in C++ by diving deep into stack vs. heap, pointers, references, new, delete, smart pointers, and much more.

Project: Memory Management Chatbot

The ChatBot project creates a dialogue where users can ask questions about some aspects of memory management in C++. Your task will be to optimize the project with modern memory management in mind.

Supporting Lessons

Lesson	Summary
Introduction	
Overview of Memory Types	This lesson covers basic concepts such as cache, virtual memory, and the structure of memory addresses. In addition, it is demonstrated how the debugger can be used to read data from memory.
Variables and Memory	In this lesson the process memory model is introduced, which contains the two fundamental memory areas heap and stack, which play an important role in C++.
Dynamic Memory Allocation (The Heap)	This lesson introduces dynamic memory allocation on the heap. The commands malloc and free as well as new and delete are introduced for this purpose.
Resource Copying Policies	This section describes how to customize resource copying using the Rule of Three. Also, the Rule of Five is introduced, which helps develop a thorough memory management strategy in your code.
Smart Pointers	In this lesson the three types of smart pointers in C++ are presented and compared. In addition, it is shown how to transfer ownership from one program part to another using copy and move semantics.

Part 5: Concurrency

Concurrent programming runs multiple threads of execution in parallel. Concurrency is an advanced programming technique that, when properly implemented, can dramatically accelerate your C++ programs.

Project: Program a Concurrent Traffic Simulation

Build a multithreaded traffic simulator using a real urban map. Run each vehicle on a separate thread, and manage intersections to facilitate traffic flow and avoid collisions.

Supporting Lessons

Lesson	Summary
Introduction and Running Threads	Learn the difference between processes and threads, and write your own concurrent program running multiple threads at the same time.
Passing Data Between Threads	Learn to use promises and futures to create safe communication between threads. Also learn about tasks as an alternative to threads. Finally, understand data races and how to avoid them.
Mutexes and Locks	Use mutexes and locks to safely access shared data from various threads.
Condition Variables and Message Queues	Use condition variables as a basic synchronization tool between threads. Understand and implement a concurrent message queue for flexible inter-thread communication

Part 6: Career Services

Project: Optimize Your GitHub Profile

Other professionals are collaborating on GitHub and growing their network. Submit your profile to ensure your profile is on par with leaders in your field.

Supporting Lessons

Lesson	Summary
Industry Research	You're building your online presence. Now learn how to share your story, understand the tech landscape better, and meet industry professionals.

Project: Improve Your LinkedIn Profile

Find your next job or connect with industry peers on LinkedIn. Ensure your profile attracts relevant leads that will grow your professional network.

Part 7: Capstone

Put your C++ skills to use on a project of your own!

Project: C++ Capstone Project

In this lesson, you will build your own C++ application, following the principles you have learned throughout this Nanodegree Program.

Udacity

Generated Thu Jan 21 05:01:39 PST 2021