UDACITY
FOR ENTERPRISE

THE SCHOOL OF PROGRAMMING AND DEVELOPMENT

C++

NANODEGREE SYLLABUS

Overview

C++ Nanodegree Program

Learn C++, a high-performance programming language used in the world's most exciting engineering jobs from self-driving cars and robotics, to web browsers, media platforms, servers, and even video games.

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 multithreaded traffic simulator and coding your own C++ application.

Program Information

TIME
4 months
Study 10 hours/week

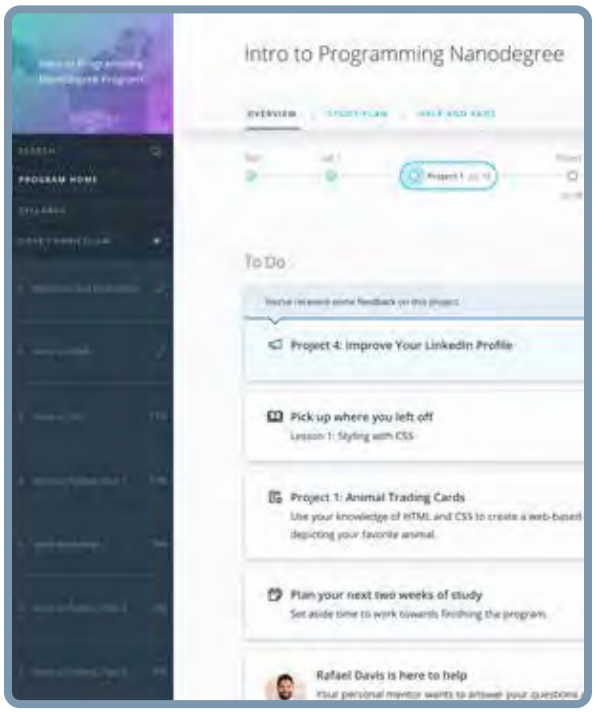
LEVEL
Practitioner

PREREQUISITES
Intermediate knowledge of object-oriented programming language.

HARDWARE/SOFTWARE REQUIRED
For this Nanodegree program, you will code with C++17. An internet connection is required. All coding can be done in our GPU-enabled Linux Workspace that runs in your browser.

LEARN MORE ABOUT THIS NANODEGREE
Contact us at enterpriseNDs@udacity.com.

Our Classroom Experience



REAL-WORLD PROJECTS

Learners build new skills through industry-relevant projects and receive personalized feedback from our network of 900+ project reviewers. Our simple user interface makes it easy to submit projects as often as needed and receive unlimited feedback.

KNOWLEDGE

Answers to most questions can be found with Knowledge, our proprietary wiki. Learners can search questions asked by others and discover in real-time how to solve challenges.

LEARNER HUB

Learners leverage the power of community through a simple, yet powerful chat interface built within the classroom. Learner Hub connects learners with their technical mentor and fellow learners.

WORKSPACES

Learners can check the output and quality of their code by testing it on interactive workspaces that are integrated into the classroom.

QUIZZES

Understanding concepts learned during lessons is made simple with auto-graded quizzes. Learners can easily go back and brush up on concepts at anytime during the course.

CUSTOM STUDY PLANS

Mentors create a custom study plan tailored to learners' needs. This plan keeps track of progress toward learner goals.

PROGRESS TRACKER

Personalized milestone reminders help learners stay on track and focused as they work to complete their Nanodegree program.

Learn with the Best

David Silver

HEAD OF CURRICULUM

David Silver leads the Udacity Curriculum Team. Before Udacity, David was a research engineer on the autonomous vehicle team at Ford. He has an MBA from Stanford, and a BSE in Computer Science from Princeton.

Stephen Welch

INSTRUCTOR

Stephen is a Content Developer at Udacity and has worked on the C++ and Self-Driving Car Engineer Nanodegree programs. He started teaching and coding while completing a Ph.D. in mathematics, and has been passionate about engineering education ever since.

Andreas Haja

INSTRUCTOR

Andreas Haja is an engineer, educator, and autonomous vehicle enthusiast. Andreas now works as an engineering professor in Germany. Previously, he developed computer vision algorithms and autonomous vehicle prototypes using C++.

Ermin Kreponic

SOFTWARE ENGINEER AT ABSTRACT THINKING

Ermin Kreponic is a skilled Java & C++ developer who has taught dozens of online courses in multiple coding languages. Ermin currently works as a cyber-security training architect and is a strong proponent of open-source technologies.

Course 1: C++ Foundations

Learn basic C++ syntax, functions, containers, and compiling and linking with multiple files. Use OpenStreetMap and the 2D visualization library IO2D to build a route planner that displays a path between two points on a map.

Project: Build an OpenStreetMap Route Planner

You'll learn about OpenStreetMap data and look at IO2D map display code. 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. When the project is finished, you will be able to select starting and ending areas on a city map, and your program will find a path along the city streets to connect the start and end. display code. 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. When the project is finished, you will be able to select starting and ending areas on a city map, and your program will find a path along the city streets to connect the start and end.

LESSON TITLE	LEARNING OUTCOME
INTRODUCTION TO THE C++ LANGUAGE	• Build on your previous programming experience to learn the basics of the C++ language. • Use vectors, loops, and I/O libraries to parse data from a file and print an ASCII board. You will use this board in the next lesson for a simplified route planning application.
A* SEARCH	• Learn about the A* search algorithm. • Use your A* search implementation to plan a path through the obstacles in the ASCII board. The program will also be able to print the solution to the screen with clean ASCII formatting.
WRITING MULTIFILE PROGRAMS	• Learn the syntax for C++ language features. • Complete an overview of header files, pointers, build tools, and classes.

Course 2: 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. In the end, you'll build a Linux system monitor application to demonstrate C++ OOP in action.

Project: System Monitor

In this project, you'll get a chance to put C++ OOP into action! You'll write a Linux system monitor with similar functionality to the widely used htop application. This will not only provide you more familiarity the Linux operating system, but also give you insights into how a collection of objects can function together in C++ to form an exciting and complete application.

LESSON TITLE	LEARNING OUTCOME
INTRODUCTION TO OOP IN C++	• Meet your instructors, get some context for what object oriented programming (OOP) is. • Practice implementing some of the basic features of OOP, like encapsulation and abstraction.
ACCESS MODIFIERS AND INHERITANCE	• C++ classes have extensive functionality when it comes to what kind of members you can define within a class and how you can prevent or provide access to those members. In addition, like many other languages, one class can inherit from another. In this lesson, you'll investigate the intricacies of access modifiers and inheritance to build more complex C++ classes.
POLYMORPHISM AND TEMPLATES	• Write member functions for a class that do different things depending on what parameters you pass to them. • Using templates, write generic functions that accept many different kinds of input parameter types. With these tools you will add more diverse functionality to your C++ classes.

Course 3: 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. By the end, you'll be ready to work on a chatbot using modern C++ memory management techniques.

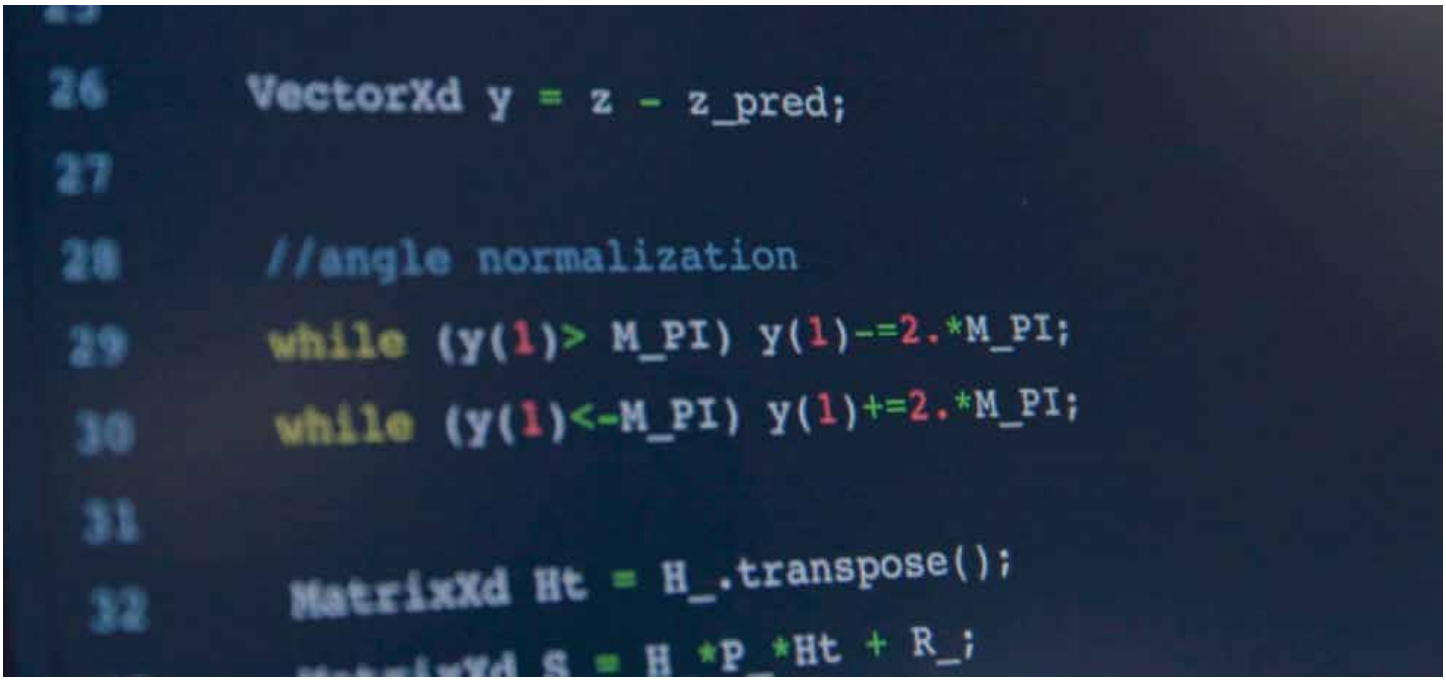
Project: ChatBot

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 memory management in mind using modern concepts such as smart pointers and move semantics.

LESSON TITLE	LEARNING OUTCOME
OVERVIEW OF MEMORY TYPES	• Understand the memory hierarchy in computer systems, which is the basis for efficient memory management. • Cover basic concepts such as cache, virtual memory, and the structure of memory addresses. • Demonstrate how the debugger can be used to read data from memory.
VARIABLES AND MEMORY	• In this section, the process memory model is introduced, which contains the two fundamental memory areas, heap and stack, which play an important role in C++. • Review the concepts of call-by-value and call-by-reference to lay the foundations for the memory-efficient passing of parameters.
DYNAMIC MEMORY ALLOCATION (THE HEAP)	• This section introduces dynamic memory allocation on the heap. Understand the main difference between stack and heap — the latter requires the programmer to take decisions about the correct allocation and deallocation of memory. • Learn the commands malloc and free, as well as new and delete, that are available for allocation of memory. • Review some of the most common problems with manual memory management.

Course 3: Memory Management, cont.

LESSON TITLE	LEARNING OUTCOME
RESOURCE COPYING POLICIES	• Customize resource copying using the Rule of Three. • Learn the basis for move semantics, lvalue and rvalue • Understand how the mechanism for memory efficient programming is one of the most important innovations in C++ and enables fast and low-cost data transfers between program scopes. • Understand the Rule of Five, which helps develop a thorough memory management strategy in your code.
SMART POINTERS	• Understand why smart pointers are a valuable tool for C++ programmers and how they help to avoid memory leaks and make it possible to establish a clear and concise resource ownership model. • Compare the three types of smart pointers in C++. • Learn how to transfer ownership from one program part to another using copy and move semantics.



Course 4: 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: 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 using state-of-the-art concurrency concepts.

LESSON TITLE	LEARNING OUTCOME
MANAGING THREADS	• Learn the differences between processes and threads. • Start your own threads in various ways and pass data to them. • Write your own concurrent program running multiple threads at the same time.
PASSING DATA BETWEEN THREADS	• Use promises and futures as a safe communication channel between threads. • Use tasks as an easy alternative to threads. • Understand data races and learn about strategies to avoid them.
MUTEXES, LOCKS, AND CONDITION VARIABLES	• Use mutexes and locks to safely access shared data from various threads. • Use condition variables as a basic synchronization tool between threads. • Understand and implement a concurrent message queue for flexible inter-thread communication.

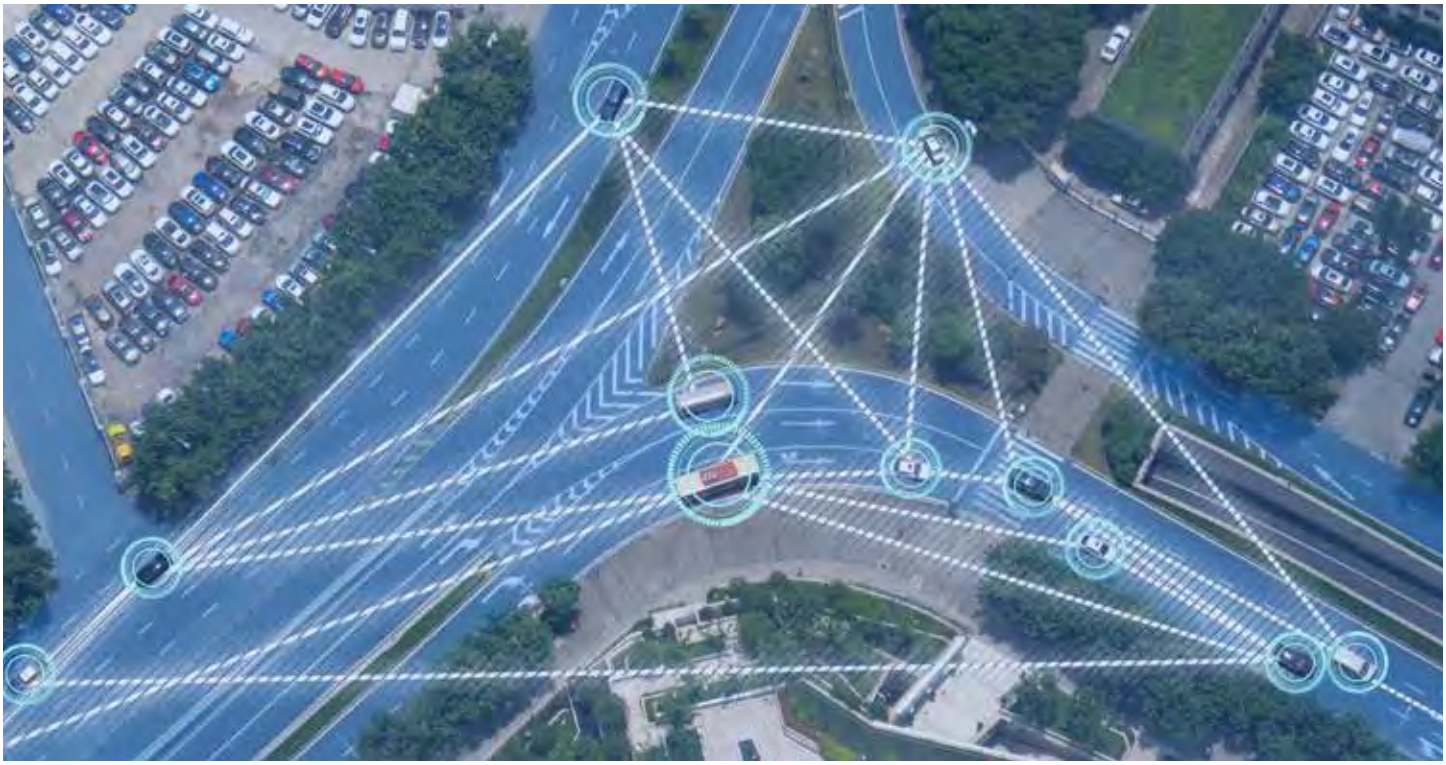
Nanodegree Program Overview

Capstone Project

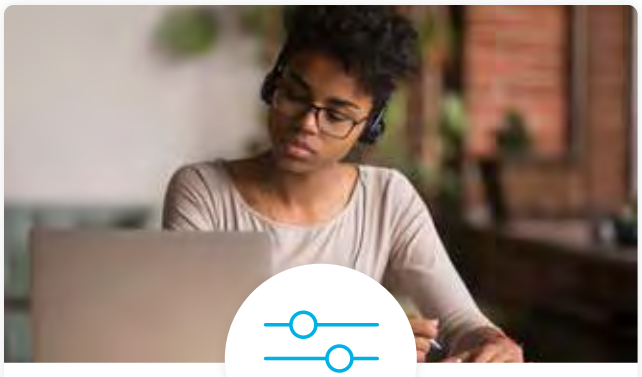
Build Your Own C++ Application

Put your C++ skills to use on a project of your own! You'll utilize the core concepts from this Nanodegree program — object-oriented programming, memory management, and concurrency — to build your own application using C++.

LESSON TITLE	LEARNING OUTCOME
PART ONE	• Choose your application. • Design the architecture. • Build a prototype.
PART TWO	• Complete your application, utilizing the core skills you have developed: C++ fundamentals, object-oriented programming, memory management, and concurrency.



Our Nanodegree Programs Include:



Pre-Assessments

Our in-depth workforce assessments identify your team's current level of knowledge in key areas. Results are used to generate custom learning paths designed to equip your workforce with the most applicable skill sets.

Dashboard & Progress Reports

Our interactive dashboard (enterprise management console) allows administrators to manage employee onboarding, track course progress, perform bulk enrollments and more.

Industry Validation & Reviews

Learners' progress and subject knowledge is tested and validated by industry experts and leaders from our advisory board. These in-depth reviews ensure your teams have achieved competency.

Real World Hands-on Projects

Through a series of rigorous, real-world projects, your employees learn and apply new techniques, analyze results, and produce actionable insights. Project portfolios demonstrate learners' growing proficiency and subject mastery.

Our Review Process

Real-life Reviewers for Real-life Projects

Real-world projects are at the core of our Nanodegree programs because hands-on learning is the best way to master a new skill. Receiving relevant feedback from an industry expert is a critical part of that learning process, and infinitely more useful than that from peers or automated grading systems. Udacity has a network of over 900 experienced project reviewers who provide personalized and timely feedback to help all learners succeed.

Vaibhav
UDACITY LEARNER

"I never felt overwhelmed while pursuing the Nanodegree program due to the valuable support of the reviewers, and now I am more confident in converting my ideas to reality."

now at
CODING VISIONS INFOTECH

All learners benefit from:

- Line-by-line feedback for coding projects
- Industry tips and best practices
- Advice on additional resources to research
- Unlimited submissions and feedback loops

How it Works

Real-world projects are integrated within the classroom experience, making for a seamless review process flow.

- Go through the lessons and work on the projects that follow
- Get help from your technical mentor, if needed
- Submit your project work
- Receive personalized feedback from the reviewer
- If the submission is not satisfactory, resubmit your project
- Continue submitting and receiving feedback from the reviewer until you successfully complete your project

About our Project Reviewers

Our expert project reviewers are evaluated against the highest standards and graded based on learners' progress. **Here's how they measure up to ensure your success.**

900+
Expert Project Reviewers
Are hand-picked to provide detailed feedback on your project submissions.

1.8M
Projects Reviewed
Our reviewers have extensive experience in guiding learners through their course projects.

3
Hours Average Turnaround
You can resubmit your project on the same day for additional feedback.

4.85 /5
Average Reviewer Rating
Our learners love the quality of the feedback they receive from our experienced reviewers.

2440 W El Camino Real, #101
Mountain View, CA 94040, USA - HQ

For more information visit: www.udacity.com/enterprise