
UDACITY
FOR ENTERPRISE
THE SCHOOL OF AUTONOMOUS SYSTEMS
Intro to Self-Driving Cars
UDACITY
ENROLL NOW
UDACITY.COM
UDACITY
NANODEGREE SYLLABUS

Overview

Intro to Self-Driving Cars Nanodegree Program

In this program, you'll sharpen your Python skills, apply C++, apply matrices and calculus in code, and touch on computer vision and machine learning. These concepts will be applied to solving self-driving car problems. At the end, you'll be ready for our Self-Driving Car Engineer Nanodegree program!

Prerequisites: Some experience with programming — writing short scripts in any programming language — and algebra is required. You should feel comfortable reading and modifying code that you are given, but it's all right if solving problems in code is still challenging.

Program Information

ESTIMATED TIME
4 months
Study 10 hours/week

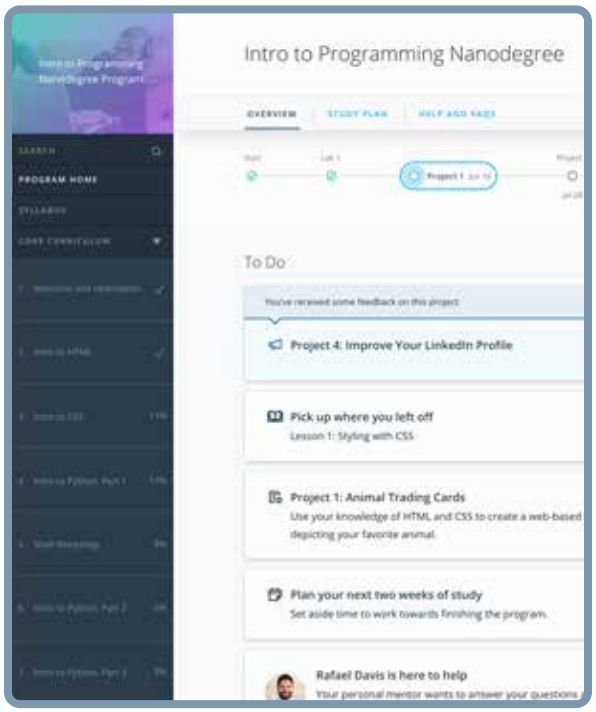
LEVEL
Practicioner

PREREQUISITES
Programming &
Mathematics

HARDWARE/SOFTWARE REQUIRED
None

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

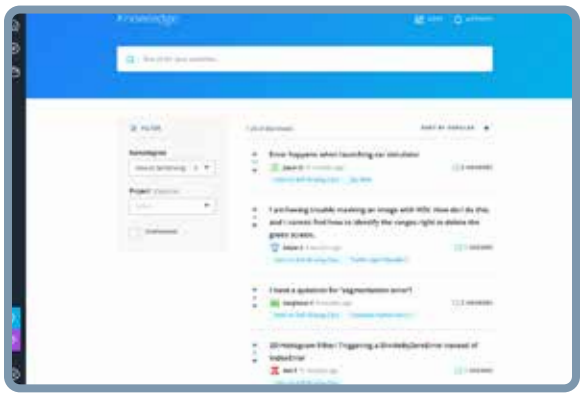
Learners leverage the power of community through a simple, yet powerful chat interface built within the classroom. Learner Hub connects learners with their technical mentor and fellow learners.

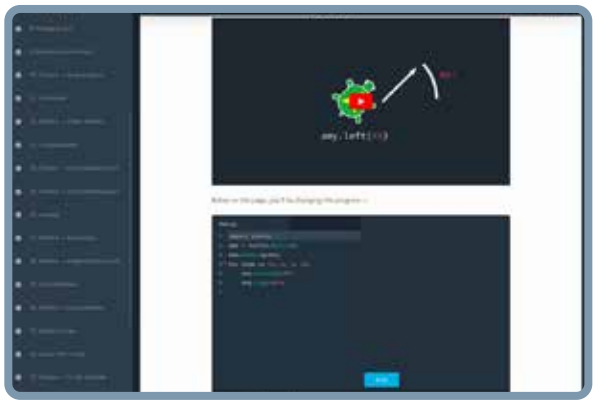
WORKSPACES

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

Sebastian Thrun

INSTRUCTOR

As the founder and president of Udacity, Sebastian's mission is to democratize education. He is also the founder of Google X, where he led projects including the Self-Driving Car, Google Glass, and more.

Andy Brown

CURRICULUM LEAD

Andy has a bachelor's degree in physics from MIT, and taught himself to program after college (mostly with Udacity courses). He has been helping Udacity make incredible educational experiences since the early days of the company.

Cezanne Camacho

COURSE DEVELOPER

Cezanne is an expert in computer vision with an M.S. in Electrical Engineering from Stanford University. Inspired by anyone with the drive and imagination to learn something new, she aims to create more inclusive and effective STEM education.

Andrew Paster

INSTRUCTOR

Andrew has an engineering degree from Yale, and has used his data science skills to build a jewelry business from the ground up. He has additionally created courses for Udacity's Self-Driving Car Engineer Nanodegree program.

Learn with the Best

Anthony Navarro

PRODUCT LEAD

Anthony is a US Army combat veteran with an M.S. in Computer Engineering from Colorado State University. Prior to being a Product Lead at Udacity, he was a Senior Software Engineer at Lockheed Martin in their Autonomous Systems R&D division.

Elecia White

ENGINEER, AUTHOR, HOST

Elecia is an embedded software engineer at Logical Elegance, Inc, the author of O'Reilly's Making Embedded Systems, and host of the Embedded.fm podcast. She enjoys sharing her enthusiasm for engineering and devices.

Tarin Ziyaee

VOYAGE, DIRECTOR OF AI

As the Director of Artificial Intelligence at Voyage Auto, Tarin works to deliver low-cost, self-driving taxis. He brings a total of 14 years experience in perception and deep neural networks working with companies such as Apple.

Course 1: Bayesian Thinking

In this course, you will learn a mathematical framework known as Bayesian Inference. This is the same framework that underlies a self-driving car's understanding of itself and the world around it. It's what allows a car to use unreliable sensor data to achieve surprisingly accurate estimates of its own location in the world (known as localization). It also underlies the tracking algorithms that self-driving cars use to predict what other traffic on the road will do in the future. By the end of this course you will not only understand Bayesian Inference, you will be able to see the world the way a self-driving car does.

Project Joy Ride

Jump into writing code that controls a simulated vehicle. Send throttle and steering commands to the car to try and get it to navigate around a test track.

Project 2D Histogram Filter in Python

In this first project, you will write the `sense` and `move` functions for a 2-dimensional histogram filter in Python.

Course 2: Working with Matrices

This course will focus on two tools which are vital to self-driving car engineers: object oriented programming and linear algebra.

Object Oriented Programming (OOP) is an approach to programming that is especially useful when the things we are modeling in our code have obvious real-world counterparts (as is often the case when writing code for something as real as a car). In this course you'll learn the basics of OOP with a focus on how to use classes which others have created.

Linear Algebra is the mathematics of matrices. For our purposes in this course we will treat it as a tool. The main goal will be to gain a basic proficiency with this powerful tool by making the notation of linear algebra approachable and understandable. With this tool in hand you'll be able to implement any of the numerous algorithms you'll encounter as you continue your self driving car career (including the ubiquitous Kalman Filter).

The goal of this course is very pragmatic: by the end you will know how to use the tools— deeper theoretical understanding is something you will acquire gradually as you continue your journey through this Nanodegree program, and the rest of your career.

Project: Implement a Matrix Class

In this project you'll practice using your object oriented programming and matrix math skills by filling out the methods in a partially-completed `Matrix` class.

Course 3: C++ Basics

C++ is the language of self-driving cars, and code written in this language can run incredibly fast. This course will be the first step in a long and rewarding journey towards C++ expertise. The goal for this course is translation: given a program written in Python, you will be able to translate it into C++.

Project Translate Python to C++

In this project you'll apply your knowledge of C++ syntax by translating the Histogram Filter code from the first course into C++.

Course 4: Performance Programming in C++

In C++ basics, you focused on the bare minimum required to write code that runs correctly. In this course, you will start to explore how to write good code that runs correctly. We'll focus primarily on the low level language features of C++ which can make C++ fast, but we'll also discuss other best practices as well.

Project Performant C++

A self-driving car can't afford to waste any cycles or memory unnecessarily. In this project you'll take some functioning (but inefficient) C++ code and optimize it.

Course 5: Navigating Complex Data Structures

What data structure should I use to model this relationship? What algorithm will accomplish that goal? These are the questions that self-driving car engineers think about on a daily basis. Algorithmic thinking is a skill you'll continue to refine throughout your programming career. In this course you'll focus on some of the data structures and algorithms that show up most frequently in self-driving cars.

Project Planning an Optimal Path

You turn on your self-driving car, buckle up, and enter a destination. Navigating from A B is not an easy problem. In this project you'll use your knowledge of data structures (in particular, graph data structures) and search algorithms to write an algorithm which uses a map and traffic information to find the quickest route between two points.

Course 6: Visualizing Calculus and Controls

It's sometimes convenient to represent the world as a discrete grid of cells. But that isn't quite right. And when it comes time to actually issue control commands about steering, throttle, and braking we have to stop pretending the world is discrete because, in reality, the world is continuous. In this course, you'll learn the basics of calculus, the mathematics of continuity. To visualize the continuous trajectories of the real-world you'll also learn to use some of Python's most popular visualization libraries.

Project Trajectory Visualizer

As a self-driving car engineer, a lot of the code you write involves simulation, visualization, testing, and debugging. In this project you'll write a visualization tool that will let you visualize the continuous trajectories that come from various search and control algorithms.

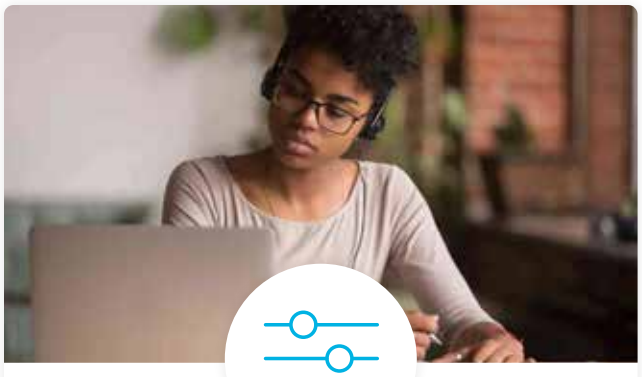
Course 7: Machine Learning and Computer Vision

How do you teach a computer the difference between a photo of a car and a photo of a human? As humans we can make the distinction without trying, but teaching this intuition to a computer is much more work. In this course, you'll learn how a computer sees an image and how we can use machine learning to teach a computer to identify images programmatically.

Project: Image Classifier from Scratch

In this project you'll build an image classifier from scratch. When you're done you'll have an algorithm that can reliably classify an image as "pedestrian" or "car".

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.

UDACITY
FOR ENTERPRISE

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

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