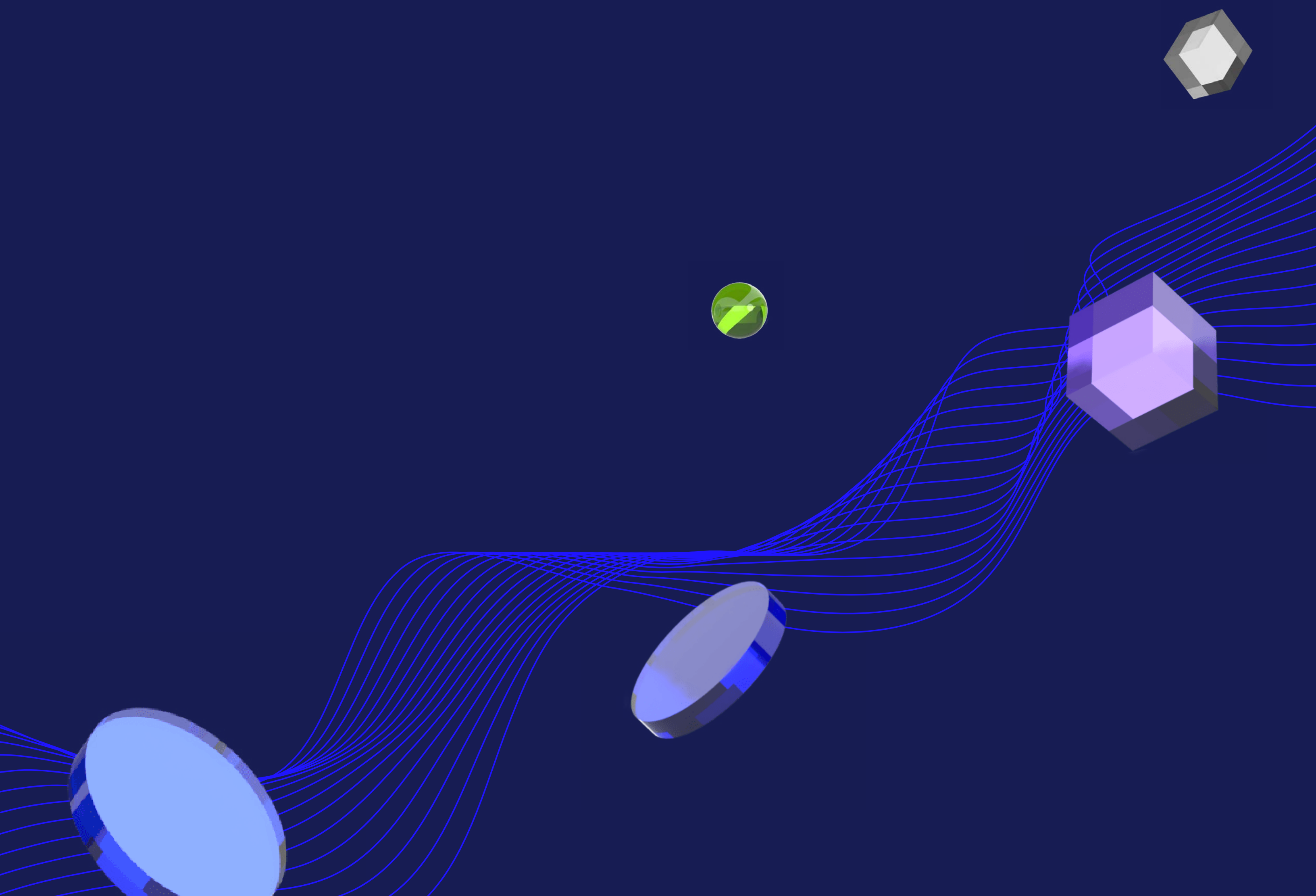
INDIVIDUAL LEARNERS

UDACITY

SCHOOL OF PROGRAMMING & DEVELOPMENT

Data Structures & Algorithms

Nanodegree Program Syllabus



Overview

The Data Structures and Algorithms Nanodegree program will help learners excel at solving everything from well-defined problems, like how to calculate the efficiency of a specific algorithm, to more open-ended problems, like building one's own private blockchain or writing a web crawler. In this program, students will learn data structures and algorithms by solving over 80 practice problems. Learners will begin each course by learning to solve defined problems related to a particular data structure and algorithm. By the end of each course, learners will would be able to evaluate and assess different data structures and algorithms for any open-ended problem and implement a solution based on the design choices.

Program information

Estimated Time

4 months at 10hrs/week*

Skill Level

Intermediate

Prerequisites

A well-prepared learner should have intermediate Python programming knowledge and basic algebra skills.

Required Hardware/Software

Learners need access to the internet and a 64-bit computer and the following software:

- Python 3
- A code/text editor, such as vim, Sublime Text, Atom, or VSCode
- A web browser
- A command line interface, such as Terminal (on Mac) or Git Bash (on Windows)

*The length of this program is an estimation of total hours the average student may take to complete all required coursework, including lecture and project time. If you spend about 5-10 hours per week working through the program, you should finish within the time provided. Actual hours may vary.

Course 1

Introduction

Get an overview of your program. Meet the instructors, and refresh one's Python skills. Learn the framework to deconstruct any open-ended problem and then understand the concepts of time and space complexity, essential tools for evaluating different data structures and algorithms.

Course Project

Unscramble Computer Science Problems

Deconstruct a series of open-ended problems into smaller components (e.g, inputs, outputs, series of functions).

Lesson 1

Introduction

Lesson 2

Python Refresher

Lesson 3

How to Solve Problems

Lesson 4

Big Notation

Course 2

Data Structures

Learn different data structures that can be used to store data. Implement different methods used to manipulate these data structures and examine the efficiency. Understand the advantages and applications of different data structures. Learn how to approach open-ended problems (either in interviews or in real-world scenarios) and select appropriate data structures based on requirements.

Course Project

Show Me the Data Structures

Solve a series of open-ended practice problems such as LRU cache, private blockchain, file recursion and many more. Hone the skills to identify and implement appropriate data structures and corresponding methods which meet given constraints.

Lesson 1

Collection Data Structures (lists, arrays, linked lists, queues, stack)

Lesson 2

Recursion

Lesson 3

Trees

Lesson 4

Maps & Hashing

- Reverse Strings
- Hamming Distance
- Reverse a Linked List
- Linked List Loop Detection,
- Balancing Brackets
- Building Queue using Stacks
- Tree Traversals
- Checking Binary Search Tree
- String Key Hash table

Course 3

Basic Algorithms

Learn and implement basic algorithms such as searching and sorting on different data structures and examine the efficiency of these algorithms. Use recursion to implement these algorithms and then learn how some of these algorithms can be implemented without recursion. Practice selecting and modifying these algorithms for a variety of interview problems.

Course Project

Problems vs. Algorithms

A series of real-world open ended problems such as request routing for web server, search term autocompletion and Fibonacci heap which train learners to apply suitable data structures and algorithms under different context.

Lesson 1

Binary Search

Lesson 2

Sorting Algorithms

Lesson 3

Divide & Conquer Algorithms

- Randomized Binary Search
- K-smallest elements using Heaps
- Build Red-Black Tree
- Bubble sort, merge sort, quick sort, sorting strings
- Linear-time median finding

Course 4

Advanced Algorithms

Build on one's algorithm skills by learning more advanced algorithms such as brute-force and greedy algorithms, graph algorithms, and dynamic programming, which optimizes recursion by storing results to sub problems.

Course Project

Route Planner

In this project, learners will build a route-planning algorithm like the one used in Google Maps to calculate the shortest path between two points on a map. Learners will first select and implement the appropriate data structure to represent points on a map and then implement the A* algorithm to find shortest path.

Lesson 1

Greedy Algorithms

Lesson 2

Graph Algorithms

Lesson 3

Dynamic Programming

Lesson 4

Linnear Programming

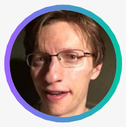
- Graph Traversals
- Dijkstra's Algorithm
- Shortest Hops
- A* Search
- Longest Palindromic subsequence
- Web crawler

Meet your instructors.

Brynn Claypoole

Instructor

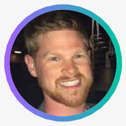
Brynn is a former Udacity employee who worked as lead data analyst at Udacity before joining Facebook as a data engineer. Currently, she is working as a software engineer with 10x Genomics.



Abe Feinberg

Content Developer

Abe is a content developer at Udacity and previously taught university courses in psychology and computer science. He loves both learning and teaching, and has a particular passion for breaking down difficult concepts and making them easier to master.

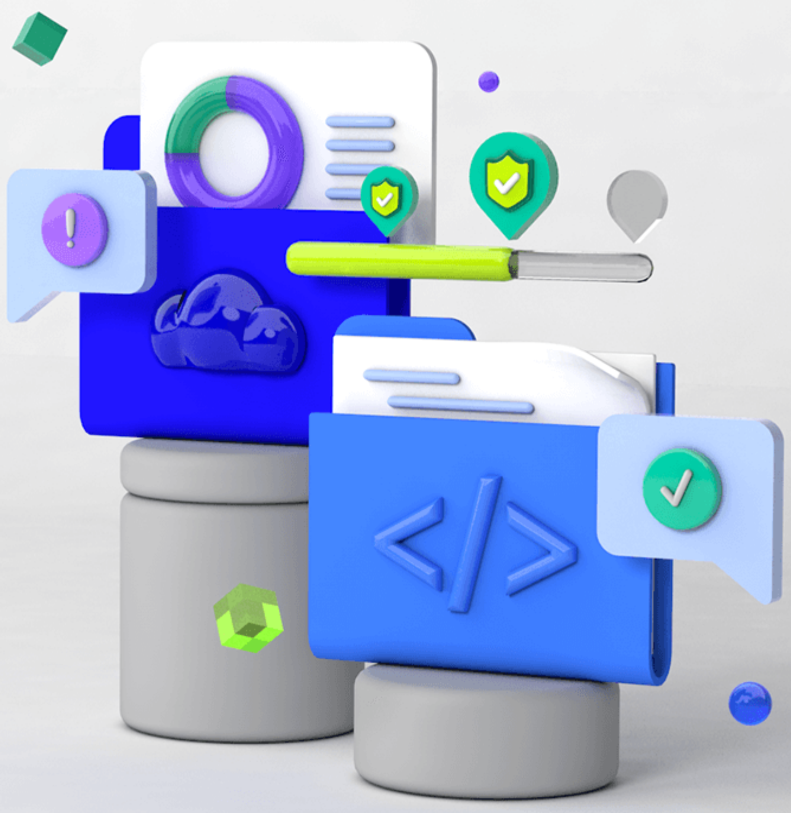


Kyle Stewart-Franz

Content Developer

Kyle has developed projects for a variety of Udacity's Nanodegree programs, such as Self-Driving Car Engineer, Robotics, and Blockchain. Kyle, a self-taught developer, is always striving towards creating great learning experience for students.

Udacity's learning experience



Hands-on Projects

Open-ended, experiential projects are designed to reflect actual workplace challenges. They aren't just multiple choice questions or step-by-step guides, but instead require critical thinking.

Quizzes

Auto-graded quizzes strengthen comprehension. Learners can return to lessons at any time during the course to refresh concepts.

Knowledge

Find answers to your questions with Knowledge, our proprietary wiki. Search questions asked by other students, connect with technical mentors, and discover how to solve the challenges that you encounter.

Custom Study Plans

Create a personalized study plan that fits your individual needs. Utilize this plan to keep track of movement toward your overall goal.

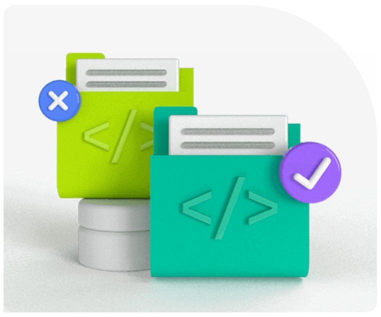
Workspaces

See your code in action. Check the output and quality of your code by running it on interactive workspaces that are integrated into the platform.

Progress Tracker

Take advantage of milestone reminders to stay on schedule and complete your program.

Our proven approach for building job-ready digital skills.



Experienced Project Reviewers

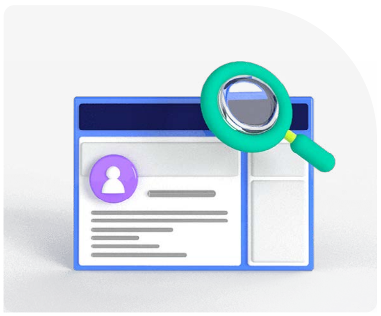
Verify skills mastery.

- Personalized project feedback and critique includes line-by-line code review from skilled practitioners with an average turnaround time of 1.1 hours.
- Project review cycle creates a feedback loop with multiple opportunities for improvement—until the concept is mastered.
- Project reviewers leverage industry best practices and provide pro tips.

Technical Mentor Support

24/7 support unblocks learning.

- Learning accelerates as skilled mentors identify areas of achievement and potential for growth.
- Unlimited access to mentors means help arrives when it's needed most.
- 2 hr or less average question response time assures that skills development stays on track.



Personal Career Services

Empower job-readiness.

- Access to a Github portfolio review that can give you an edge by highlighting your strengths, and demonstrating your value to employers.*
- Get help optimizing your LinkedIn and establishing your personal brand so your profile ranks higher in searches by recruiters and hiring managers.

Mentor Network

Highly vetted for effectiveness.

- Mentors must complete a 5-step hiring process to join Udacity's selective network.
- After passing an objective and situational assessment, mentors must demonstrate communication and behavioral fit for a mentorship role.
- Mentors work across more than 30 different industries and often complete a Nanodegree program themselves.

*Applies to select Nanodegree programs only.

UDACITY

Learn more at

www.udacity.com/online-learning-for-individuals →

12.02.22 | V1.0