UDACITY

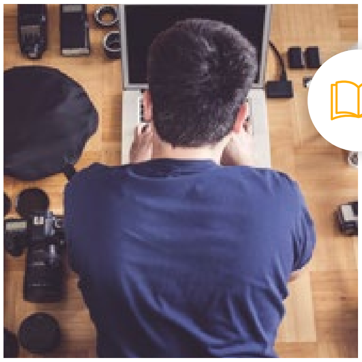
NANODEGREE PROGRAM SYLLABUS

Artificial Intelligence

Overview

This program will teach you how to become a better Artificial Intelligence or Machine Learning Engineer by teaching you classical AI algorithms applied to common problem types. You will complete projects and exercises incorporating search, optimization, planning, and probabilistic graphical models which have been used in Artificial Intelligence applications for automation, logistics, operations research, and more. These concepts form the foundation for many of the most exciting advances in AI in recent years. Each project you build will be an opportunity to demonstrate what you've learned in your lessons, and become part of a career portfolio that will demonstrate your mastery of these skills to potential employers.

Estimated Time:
3 Months at
12-15hrs/week



Prerequisites:
Algebra, Calculus, Statistics, & Python

Flexible Learning:
Self-paced, so you can learn on the schedule that works best for you

Technical Mentor Support:
Our knowledgeable mentors guide your learning and are focused on answering your questions, motivating you and keeping you on track

Course 1: Introduction to Artificial Intelligence

In this course, you'll learn about the foundations of AI. You'll configure your programming environment to work on AI problems with Python. At the end of the course you'll build a Sudoku solver and solve constraint satisfaction problems.

Course Project :
Build a Sudoku Solver

Humans use reason to solve problems by decomposing the problem statement and incorporating domain knowledge to limit the possible solution space. In this project you'll use a technique called constraint propagation together with backtracking search to make an agent that only considers reasonable solution candidates and efficiently solves any Sudoku puzzle. This approach appears in many classical AI problems, and the solution techniques have been extended and applied to diverse problems in bioinformatics, logistics, and operations research. In this project you will demonstrate some basic algorithms knowledge, and learn to use constraint satisfaction to solve general problems.

	LEARNING OUTCOMES	
LESSON ONE	**Welcome to Artificial Intelligence**	• Learn about the resources available to help you succeed
LESSON TWO	**Intro to Artificial Intelligence**	• Consider the meaning of "artificial intelligence" • Be able to define core concepts from AI including "agents", "environments", and "states" • Learn the concept of "rational" behavior for AI agents
LESSON THREE	**Solving Sudoku with AI**	• Express logical constraints as Python functions • Use constraint propagation & search to solve all Sudoku puzzles
LESSON FOUR	**Setting Up your Environment**	• Use Conda to configure and manage Python packages and dependencies
LESSON FIVE	**Constraint Satisfaction Problems**	• Learn to represent problems in terms of logical constraints • Use constraint propagation to limit the potential solution space • Incorporate backtracking search to find a solution when the set of constraints is incomplete • Use Z3 Solver to solve constraint satisfaction problems

Course 2: Classical Search

In this course you'll learn classical graph search algorithms--including uninformed search techniques like breadth-first and depth-first search and informed search with heuristics including A*. These algorithms are at the heart of many classical AI techniques, and have been used for planning, optimization, problem solving, and more. Complete the lesson by teaching PacMan to search with these techniques to solve increasingly complex domains.

	LEARNING OUTCOMES	
LESSON ONE	**Introduction**	• Learn about the significance of search in AI
LESSON TWO	**Uninformed Search**	• Learn uninformed search techniques including Depth-First • Search, Breadth-First Search, and Uniform Cost Search
LESSON THREE	**Informed Search**	• Learn informed search techniques (using heuristics) including A* Search • Understand admissibility and consistency conditions for heuristics
LESSON FOUR	**Classroom Exercise: Search**	• Implement informed & uninformed search for Pacman
LESSON FIVE	**Additional Topics: Search**	• List of external resources for you to continue learning about search

Course 3: Automated Planning

In this course you'll learn to represent general problem domains with symbolic logic and use search to find optimal plans for achieving your agent's goals. Planning & scheduling systems power modern automation & logistics operations, and aerospace applications like the Hubble telescope & NASA Mars rovers.

Course Project:
Build a Forward Planning Agent

Intelligent agents are expected to act in complex domains where their goals and objectives may not be immediately achievable. They must reason about their goals and make rational choices of actions to achieve them. In this project you will build a system using symbolic logic to represent general problem domains and use classical search to find optimal plans for achieving your agent's goals. Planning & scheduling systems power modern automation & logistics operations, and aerospace applications like the Hubble telescope & NASA Mars rovers.

In this project you will demonstrate an understanding of classical optimization & search algorithms, symbolic logic, and domain-independent planning.

	LEARNING OUTCOMES	
LESSON ONE	**Symbolic Logic & Reasoning**	• Learn Propositional Logic (propositions & statements) • Learn First-Order logic (quantifiers, variables, & objects) • Encode problems with symbolic constraints using First Order Logic
LESSON TWO	**Introduction to Automated Planning**	• Learn to define planning problems
LESSON THREE	**Classical Planning**	• Learn high-level features of automated planning techniques using search & symbolic logic including forward planning, backwards planning, & hierarchical planning • Explore planning heuristics & planning graphs
LESSON FOUR	**Additional Topics: Search**	• List of external resources for you to continue learning about search

Course 4: Optimization Problems

In this course you'll learn about iterative improvement optimization problems and classical algorithms emphasizing gradient-free methods for solving them. These techniques can often be used on intractable problems to find solutions that are "good enough" for practical purposes, and have been used extensively in fields like Operations Research & logistics. You'll finish the lesson by completing a classroom exercise comparing the different algorithms' performance on a variety of problems.

	LEARNING OUTCOMES	
LESSON ONE	**Introduction**	• Introduce iterative improvement problems that can be solved with optimization
LESSON TWO	**Hill Climbing**	• Learn Random Hill Climbing for local search optimization problems
LESSON THREE	**Simulated Annealing**	• Learn to use Simulated Annealing for global optimization problems
LESSON FOUR	**Genetic Algorithms**	• Explore and implement Genetic Algorithms that keep a pool of candidates to solve optimization problems
LESSON FIVE	**Classroom Exercise: Optimization Problems**	• Compare optimization techniques on a variety of problems
LESSON SIX	**Additional Optimization Topics**	• Learn about improvements & optimizations to optimization search including Late Acceptance Hill Climbing, Basin Hopping, & Differential Evolution

Course 5: Adversarial Search

In this course you'll learn how to search in multi-agent environments (including decision making in competitive environments) using the minimax theorem from game theory. Then build an agent that can play games better than any humans.

Course Project:
Build an Adversarial Game Playing Agent

AI agents acting in the real world have to "hope for the best, but prepare for the worst." In this project you will write an agent that uses that idea to make rational choices to achieve super-human performance in games competing against adversarial agents. The principles of adversarial search provide a foundation for autonomous agents acting in the real world, and for understanding modern advances in AI like DeepMind's AlphaGo Zero.

In this project you will demonstrate advanced algorithms knowledge, including minimax with alpha-beta pruning for adversarial search.

	LEARNING OUTCOMES	
LESSON ONE	**Search in Multi-Agent Domains**	• Understand "adversarial" problems & applications (e.g., multi-agent environments) • Extend state space search techniques to domains your agents do not fully control • Learn the minimax search technique
LESSON TWO	**Optimizing Minimax Search**	• Apply depth-limiting to overcome limitations in basic minimax search • Apply alpha-beta pruning to overcome limitations in basic minimax search
LESSON THREE	**Extending Minimax Search**	• Extend adversarial search to non-deterministic domains and domains with more than two players
LESSON FOUR	**Additional Adversarial Search Topics**	• Understand other adversarial search techniques such as Monte Carlo Tree Search • List of external resources for you to continue learning about adversarial search

Course 6: Fundamentals of Probabilistic Graphical Models

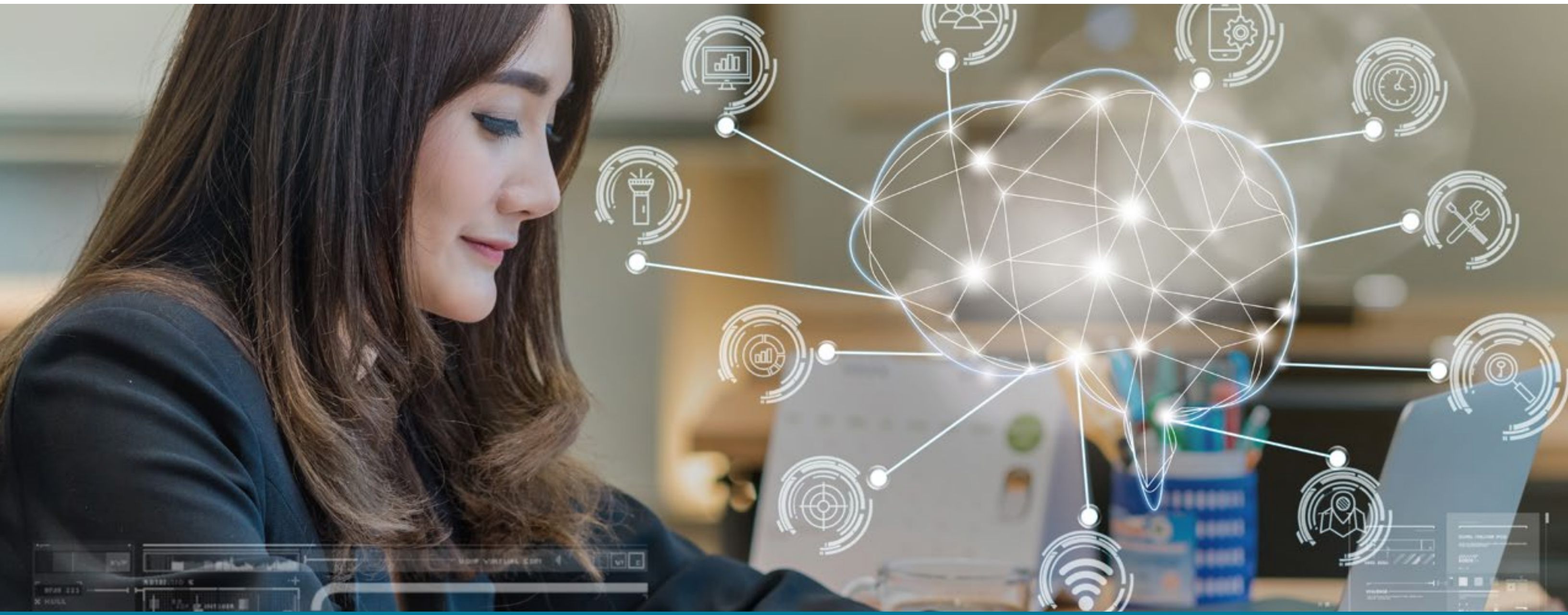
In this course you'll learn to use Bayes Nets to represent complex probability distributions, and algorithms for sampling from those distributions. Then learn the algorithms used to train, predict, and evaluate Hidden Markov Models for pattern recognition. HMMs have been used for gesture recognition in computer vision, gene sequence identification in bioinformatics, speech generation & part of speech tagging in natural language processing, and more.

Course Project:
Part of Speech Tagging

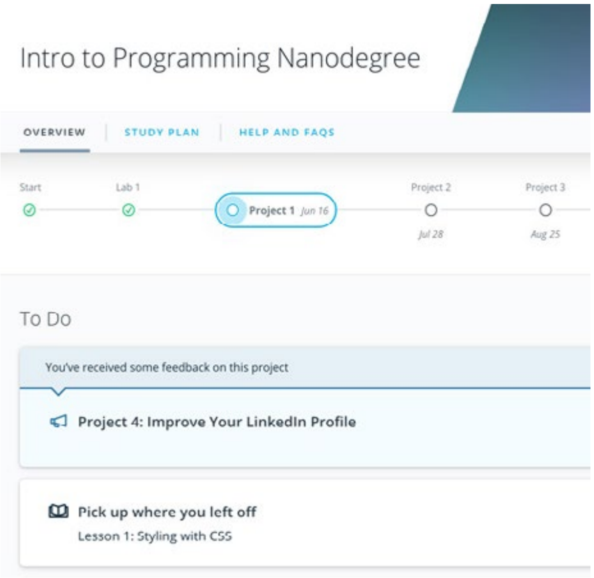
Probabilistic models allow your agents to better handle the uncertainty of the real world by explicitly modeling their belief state as a distribution over all possible states. In this project you'll use a Hidden Markov Model (HMM) to perform part of speech tagging, a common pre-processing step in Natural Language Processing. HMMs have been used extensively in NLP, speech recognition, bioinformatics, and computer vision tasks.

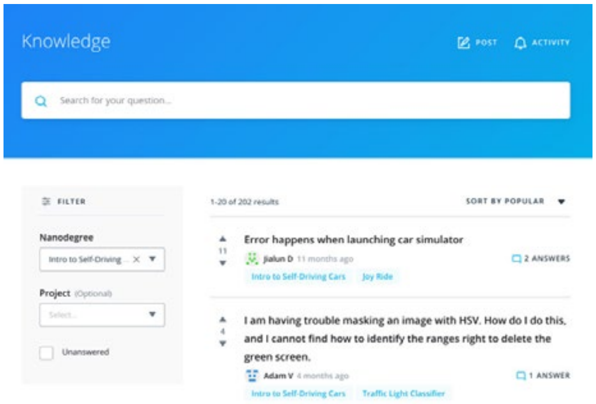
	LEARNING OUTCOMES	
LESSON ONE	**Introduction to Probabilistic Models**	• Model probability distributions based on a given set of parameters in a real-world use case, using discrete distributions
LESSON TWO	**Probability**	• Review key concepts in probability including discrete distributions, joint probabilities, and conditional probabilities
LESSON THREE	**Bayes Nets**	• Efficiently encode joint probabilities in Bayes networks
LESSON FOUR	**Inference in Bayes Nets**	• Learn about inference in Bayes networks through exact enumeration with optimizations • Learn techniques for approximate inference in more complex Bayes networks

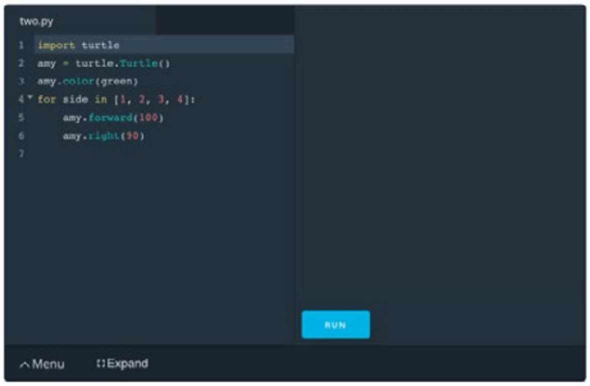
LESSON FIVE	**Hidden Markov Models**	• Learn parameters to maximize the likelihood of model parameters to training data • Determine the likelihood of observing test data given a fixed model • Learn an algorithm to Identify the most likely sequence of states in a model given some data
LESSON SIX	**Dynamic Time Warping**	• Learn the dynamic time warping algorithm for time-series analysis



Our Classroom Experience







REAL-WORLD PROJECTS
Build your skills through industry-relevant projects. Get personalized feedback from our network of 900+ project reviewers. Our simple interface makes it easy to submit your projects as often as you need and receive unlimited feedback on your work.

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

WORKSPACES
See your code in action. Check the output and quality of your code by running them on workspaces that are a part of our classroom.

QUIZZES
Check your understanding of concepts learned in the program by answering simple and auto-graded quizzes. Easily go back to the lessons to brush up on concepts anytime you get an answer wrong.

CUSTOM STUDY PLANS
Create a custom study plan to suit your personal needs and use this plan to keep track of your progress toward your goal.

PROGRESS TRACKER
Stay on track to complete your Nanodegree program with useful milestone reminders.

Learn with the Best

Peter Norvig

RESEARCH DIRECTOR

Peter Norvig is a Director of Research at Google and is co-author of Artificial Intelligence: A Modern Approach, the leading textbook in the field.

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.

Thad Starner

PROFESSOR OF COMPUTER SCIENCE

Thad Starner is the director of the Contextual Computing Group (CCG) at Georgia Tech and is also the longest-serving Technical Lead/Manager on Google's Glass project.

All Our Nanodegree Programs Include:

EXPERIENCED PROJECT REVIEWERS

REVIEWER SERVICES

- Personalized feedback & line by line code reviews
- 1600+ Reviewers with a 4.85/5 average rating
- 3 hour average project review turnaround time
- Unlimited submissions and feedback loops
- Practical tips and industry best practices
- Additional suggested resources to improve

TECHNICAL MENTOR SUPPORT

MENTORSHIP SERVICES

- Questions answered quickly by our team of technical mentors
- 1000+ Mentors with a 4.7/5 average rating
- Support for all your technical questions

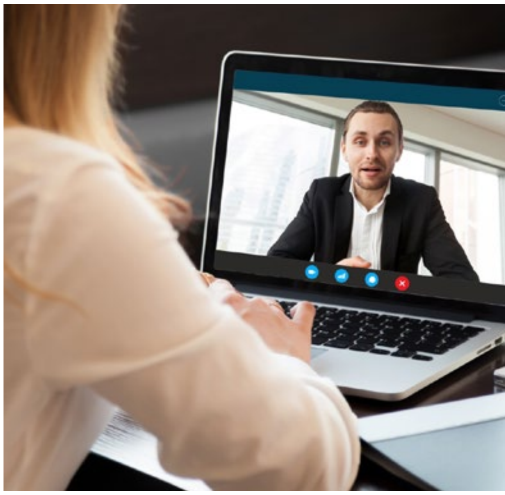
PERSONAL CAREER SERVICES

CAREER SUPPORT

- Github portfolio review
- LinkedIn profile optimization

Frequently Asked Questions

PROGRAM OVERVIEW



WHY SHOULD I ENROLL?

The Artificial Intelligence Nanodegree program features expert instructors, and world-class curriculum built in collaboration with top companies in the field. The program offers a broad introduction to the field of artificial intelligence, and can help you maximize your potential as an artificial intelligence or machine learning engineer. If you're ready for an efficient and effective immersion in the world of AI, with the goal of pursuing new opportunities in the field, this is an excellent program for you.

WHAT JOBS WILL THIS PROGRAM PREPARE ME FOR?

This Nanodegree program is designed to build on your existing skills as an engineer or developer. As such, it doesn't prepare you for a specific job, but instead expands your skills with artificial intelligence algorithms. These skills can be applied to various applications such as video game AI, pathfinding for robots, and recognizing patterns over time like handwriting and sign language.

HOW DO I KNOW IF THIS PROGRAM IS RIGHT FOR ME?

In this Nanodegree program, you will learn from the world's foremost AI experts, and develop a deep understanding of algorithms being applied to real-world problems in Natural Language Processing, Computer Vision, Bioinformatics and more. If your goal is to become an AI expert, then this program is ideal, because it teaches you some of the most important algorithms in AI. You'll benefit from a structured approach for applying these techniques to new challenges, and emerge from the program fully prepared to advance in the field.

ENROLLMENT AND ADMISSION

DO I NEED TO APPLY? WHAT ARE THE ADMISSION CRITERIA?

There is no application. This Nanodegree program accepts everyone, regardless of experience and specific background.

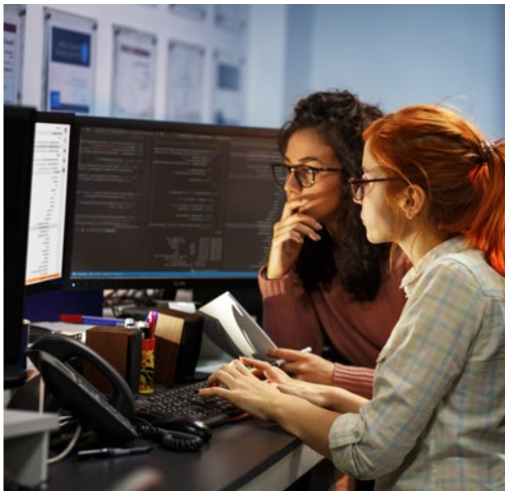
WHAT ARE THE PREREQUISITES FOR ENROLLMENT?

If you are new to Python programming, we suggest our **AI Programming with Python Nanodegree** program. You must have completed a course in computer science algorithms equivalent to the **Data Structures & Algorithms Nanodegree** program prior to entering the program. Additionally, you should have the following knowledge:

FAQs Continued

Prerequisites: A well-prepared learner has experience with:

- Basic Algorithms
- Basic Calculus
- Basic Probability
- Command Line Interface Basics
- Intermediate Python
- Linear Algebra
- Object-Oriented Programming Basics
- Basic Data Structures and Algorithms
- Basic Descriptive Statistics
- Differential Calculus
- Jupyter Notebooks
- Scripting
- Constraint Satisfaction Problems
- Optimization Algorithms
- Search Algorithms

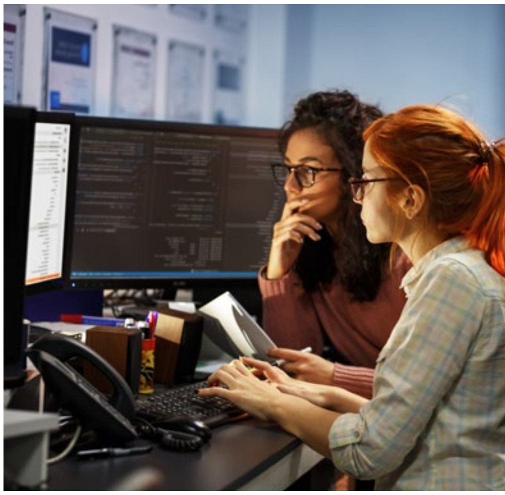


IF I DO NOT MEET THE REQUIREMENTS TO ENROLL, WHAT SHOULD I DO?
We have a number of courses and programs we can recommend that will help prepare you for the program, depending on the areas you need to address. For example:

- Intro to Machine Learning
- Artificial Intelligence Programming with Python Nanodegree program
- Data Structures & Algorithms Nanodegree program
- Data Analyst Nanodegree program
- Intro to Machine Learning
- Machine Learning Engineer Nanodegree program

FAQs Continued

TUITION AND TERM OF PROGRAM



HOW IS THIS NANODEGREE PROGRAM STRUCTURED?
The Artificial Intelligence Nanodegree program is comprised of content and curriculum to support four (4) projects. We estimate that students can complete the program in three (3) months working 12-15 hours per week.

Each project will be reviewed by the Udacity reviewer network. Feedback will be provided and if you do not pass the project, you will be asked to resubmit the project until it passes.

HOW LONG IS THIS NANODEGREE PROGRAM?
Access to this Nanodegree program runs for the length of time specified in the payment card above. If you do not graduate within that time period, you will continue learning with month to month payments. See the **Terms of Use** and **FAQs** for other policies regarding the terms of access to our Nanodegree programs.

CAN I SWITCH MY START DATE? CAN I GET A REFUND?
Please see the Udacity Nanodegree program **FAQs** for policies on enrollment in our programs.

SOFTWARE AND HARDWARE

WHAT SOFTWARE AND VERSIONS WILL I NEED IN THIS PROGRAM?
You will need a computer running a 64-bit operating system (most modern Windows, OS X, and Linux versions will work) with at least 8GB of RAM, along with administrator account permissions sufficient to install programs including Anaconda with Python 3.6 and supporting packages. Your network should allow secure connections to remote hosts (like SSH). We will provide you with instructions to install the required software packages.