EXA AI Roadmap (Based on Stanford AI Graduate Certificate Program

Master the concepts and skills of the Stanford AI Graduate Certificate program, entirely for FREE! Our curated guide offers a structured path to AI excellence, tailored to your individual pace.

Hi, I'm Jean

I'm the Founder and host of Exaltitude on YouTube. I've worked in tech for the past 20 years as an engineer, an engineering manager, and a team builder. I was the 19th engineer at WhatsApp and worked with Facebook as an Engineering Manager for six years after the $19B acquisition.

Throughout my career, I've mentored and coached countless Software Engineers and Managers from diverse backgrounds, noticing common questions around direction and growth: "Where am I headed, and how do I get there?" This inspired me to share my insights, helping future engineers build purposeful, successful careers.

Stay connected for updates, industry insights, and career advice on LinkedIn and YouTube.

Have questions? Reach out on my website!

Expectations

While most students complete the Stanford AI Graduate Certificate program in 1-2 years when studying part-time, self-study can take significantly longer.
The amount of time it takes depends on various factors, including:

- **Your experience level:** If you have a strong foundation in math, programming, and related fields, you may be able to progress faster.
- **Your dedication and time commitment:** The more time you devote to studying, the quicker you can complete the program.
- **The depth of your learning:** If you want to gain a deep understanding of each topic, you may need to spend more time.

It's important to set realistic expectations and be patient with yourself. Remember that the goal is to learn and understand the material, not just to finish the program quickly.

Phase 1: Prerequisite-Math Foundations

- Calculus
    - Goals: Understand derivatives, integrals, and fundamental theorems.
    - Stanford Classes: MATH19, MATH20, MATH21
    - Free Classes/Resources (choose one):
        - Khan Academy Calculus: Khan Academy Calculus
        - MIT OpenCourseWare Single Variable Calculus
    - Recommended Book:
        - Calculus for Dummies
        - Calculus: A New Horizon, 6th edition (or later) (paid resource)
    - Estimated Time: 4-6 weeks

- Linear Algebra
    - Goals: Master matrix operations, vector spaces, and linear transformations.
    - Stanford Classes: MATH104, MATH113, CS205L
    - Free Classes/Resources (choose one):
        - Khan Academy Linear Algebra
        - MIT OpenCourseWare Introduction to Linear Algebra
    - Recommended Book:
        - Linear Algebra, by Bronson (paid resource)
        - Linear Algebra Done Right by Sheldon Axler
    - Estimated Time: 4-6 weeks
- Probability and Statistics
    - Goals
        - Learn probability topics like counting, random variables, mean variance, Bayes' theorem, distributions, limit theorems
        - Learn statistics topics like linear regression, classification, tree-based methods
    - Stanford Classes: CS109, STATS116
    - Free Classes/Resources (choose one):
        - Khan Academy Probability and Statistics
        - MIT OpenCourseWare Introduction to Probability and Statistics
    - Recommended Book: A First Course in Probability, by Sheldon Ross, Pearson (paid resource)
    - Estimated Time: 4-6 weeks
- All in one course: Mathematics for Machine Learning and Data Science Specialization
    - Linear Algebra for Machine Learning and Data Science
    - Calculus for Machine Learning and Data Science

- Probability & Statistics for Machine Learning & Data Science

Phase 2: Programming Fundamentals

- Linux Command Line
  - Goals: Master essential Linux commands and shell scripting.
  - Stanford Classes: CS193
  - Classes/Resources (choose one):
    - Linux Academy
    - Introduction to Shell by DataCamp (paid)
  - Recommended Written Tutorial: Ubuntu's The Linux command line for beginners
  - Estimated Time: 1-2 weeks
- Object-Oriented Programming (OOP) (optional)
  - Goals: Understand data types, control flow, and functions through object-oriented programming (OOP)
  - Stanford Classes: CS108, Syllabus
  - Free Classes/Resources (choose one):
    - Codecademy's Learn Java
    - Introduction to Java and Object-Oriented Programming on Coursera
  - No recommended textbook for this class
  - Estimated Time: 4-6 weeks
- Data Structures and Algorithms
  - Goals: Learn common data structures and algorithms.
  - Stanford Classes: CS166, Syllabus
  - Free Classes/Resources (choose one):

        - Coursera's Foundations of Data Structures and Algorithms Specialization
            - Algorithms for Searching, Sorting, and Indexing (recommended)
            - Trees and Graphs: Basics (recommended)
            - Dynamic Programming, Greedy Algorithms (optional)
            - Approximation Algorithms and Linear Programming (optional)
            - Advanced Data Structures, RSA and Quantum Algorithms (optional)
        - Google Tech Dev Guide
    - Recommended Book: Introduction to Algorithms, Third Edition by Cormen, Leiserson, Rivest, and Stein (paid resource)
    - Estimated Time: 8-12 weeks
- Python
    - Goals: Become proficient in Python syntax and libraries.
    - Stanford Classes: CS41, Syllabus
    - Free Classes/Resources:
        - Jean's Python Roadmap - Free 12-page PDF
        - Jean's Python Roadmap Video on YouTube
    - Recommended Book: Automate the Boring Stuff
    - Estimated Time: 4-6 weeks
- Python Libraries
    - Goals: Familiarize yourself with popular libraries.
    - Classes/Resources:
        - Stanford's Numpy Tutorial
        - Pandas Tutorial
    - Estimated Time: 2-4 weeks

Phase 3: AI Fundamentals

- Choose your goal between Machine Learning or broader AI.
    - Both options are challenging courses that provide a deep dive into artificial intelligence but focus on different aspects of the field. The choice between the two specifications depends on your specific interests and goals within the field of AI.
    - If you're unsure, check out the Zero To Mastery's Free Career Path Quiz.
- Option 1: Machine Learning
    - Goals: Understand supervised and unsupervised learning, large language models, and reinforcement learning.
    - Stanford classes: CS229 Machine Learning, Syllabus
    - Classes/Resources:
        - Andrew Ng's Machine Learning Specialization on Coursera
        - Fast.ai's Introduction to Machine Learning for Coders
        - Cheat sheets
        - Complete A.I. Machine Learning and Data Science: Zero to Mastery (paid)
    - Recommended Book:
        - Main notes by Andrew Ng
        - Check out the Syllabus for more free written materials
    - Estimated Time: 8-12 weeks
- Option 2: Artificial Intelligence
    - Goals: Build foundational principles and tools of artificial intelligence, including Constraint Satisfaction, Games, Markov decision processes, Factor Graphs, and Logic.
    - Stanford classes CS221 Artificial Intelligence: Principles and Techniques, Syllabus
    - Classes/Resources:
        - MIT OpenCourseWare: Introduction to Artificial Intelligence
        - AI for Developers by DataCamp (paid)

- Additional Courses:
  - Modules and resources - some resources are open to the public, such as Stanford CS221 Logic Series on YouTube
  - RL Course by David Silver - Lecture 2: Markov Decision Process on YouTube
  - Coursera's Probabilistic Graphical Models Specialization
- Recommended Reading:
  - Russell and Norvig. Artificial Intelligence: A Modern Approach A comprehensive reference for all the AI topics covered in course (free online)
  - Koller and Friedman. Probabilistic Graphical Models (Paid resource)
  - Sutton and Barto. Reinforcement Learning: An Introduction: Covers Markov decision processes and reinforcement learning (free online)
  - Hastie, Tibshirani, and Friedman. The Elements of Statistical Learning: Covers machine learning from a rigorous statistical perspective (free online)
  - Tsang. Foundations of Constraint Satisfaction: Covers constraint satisfaction problems (free online)
- Estimated Time: 8-12 weeks

- Additional learning (optional):
  - TensorFlow
    - TensorFlow official tutorials
    - TensorFlow for Deep Learning by Zero To Mastery (paid)
  - PyTorch
    - PyTorch official tutorials
    - PyTorch for Deep Learning by Zero To Mastery (paid)

Phase 4: Capstone project

Developing a capstone project to master AI and ML is a great way to apply your knowledge and skills to a real-world problem and dig deeper into your learning. When choosing a project topic, you can focus on

- Application of machine learning (This is the most common)
- Develop a new algorithm
- Explore theoretical properties of algorithms (This is often quite difficult, and so very few, if any, projects will be purely theoretical.)

A top project as stanford considers it can be of publishable quality, for submission to conferences or journals. For inspiration, review recent machine learning research papers from major conferences like ICML and NeurIPS, or look at Stanford's past class projects for ideas.

Tips for the project

Here are some steps to help you develop a plan:

- **Identify your interests:** What areas of AI and ML interest you the most? This will help you narrow down your project topic.
- **Choose a problem:** Once you've identified your interests, start brainstorming potential problems you can solve using AI and ML. Look for problems that are challenging but achievable.
- **Define your project's scope: Clearly define your project's goals and objectives**. What do you want to achieve? What are the key questions you want to answer?
- **Gather data:** Collect the data you'll need to train and evaluate your AI or ML model. While using preprepared datasets (such as Kaggle) is fine, it's important to explore and analyze the data, including preprocessing and error analysis, to fully understand the problem.

- **Choose an algorithm or model:** Select the appropriate AI or ML algorithm or model for your project. Consider the nature of your data and the problem you're trying to solve.
- **Implement your model:** Use a programming language like Python and libraries like TensorFlow or PyTorch to implement your AI or ML model.
- **Train and evaluate your model:** Train it on your data and assess its performance using appropriate metrics.
- **Iterate and improve:** If your model is not performing as well as you'd like, iterate on your approach and make improvements.
- **Present your results:** Create a presentation or report summarizing your project, the methods you used, and your results. Refer to the CS 229 Final Report Guidelines for more information on writing reports.

Ultimately, the best project for you will depend on your specific interests and career goals. Consider your previous coursework, your strengths and weaknesses, and the areas of AI that excite you the most.
Also, check out the guideline: CS229 Final Project Spring 2022.

Phase 5: Choose Electives (Advanced Topics)

For Stanford, you'd choose at most 3 elective classes from this list. To go deeper into specific paths choose topics that interests you and select appropriate projects in the area. Here are the recommended courses from Stanford for each option:

Machine learning:

- CS224W: Machine Learning with Graphs
    - Summary: Diseases, information, traffic, and weather can be predicted by analyzing large datasets, which are often represented as graphs of relationships and interactions. Algorithms like neural networks process

these graph structures to perform tasks such as classification, clustering, and regression.

- Topics:
  - Representation Learning and Graph Neural Networks
  - Algorithms for the World Wide Web
  - Reasoning over Knowledge Graphs
  - Influence maximization
  - Disease outbreak detection
  - Social network analysis
- Prerequisites:
  - Knowledge of computer science principles at a level sufficient to write a non-trivial computer program. CS107 Computer Organization and Systems, CS145 Data Management and Data Systems, or equivalent)
  - Basic probability theory and linear algebra
- Free Classes/Resources: CS224W Machine Learning with Graphs Playlist on YouTube

- CS230: Deep Learning
  - Summary: This course equips you with deep learning foundations, covering neural network construction and machine learning project leadership while exploring topics like CNNs, RNNs, and LSTMs and applying these concepts to real-world case studies using Python and TensorFlow.
  - Topics:
    - Foundations of neural networks and deep learning
    - Techniques to improve neural networks: regularization and optimizations, hyperparameter tuning, and deep learning frameworks (Tensorflow and Keras.)
    - Strategies to organize and successfully build a machine learning project

        - Convolutional Neural Networks, their applications (object classification, object detection, face verification, style transfer), and related methods
        - Recurrent Neural Networks, their applications (natural language processing, speech recognition), and related methods
        - Advanced topics: Generative Adversarial Networks, Deep Reinforcement Learning, Adversarial Attacks
        - Insights from the AI industry, from academia, and advice to pursue a career in AI
    - Prerequisites: Python and Linear Algebra (matrix/vector multiplications)
    - Free Classes/Resources: Lecture series on YouTube
    - Other Resources:
        - Practical Deep Learning for Coders on Fast ai
        - Deep Learning in Python by DataCamp (paid)

Computer Vision

- CS231N: Deep Learning for Computer Vision
    - Summary: Computer vision, integral to applications like search, medicine, and self-driving cars, relies on neural network advancements for tasks like image classification and object detection to improve visual recognition performance.
    - Topics:
        - End-to-end models
        - Image classification, localization, and detection
        - Implementation, training, and debugging
        - Learning algorithms, such as backpropagation
        - Long Short Term Memory (LSTM)
        - Recurrent Neural Networks (RNN)
        - Supervised and unsupervised learning
    - Prerequisites:

        - Python (and NumPy)
        - Calculus, Linear Algebra, Basic Probability and Statistics
    - Free Classes/Resources: Lectures on Class Central
- CS231A: Computer Vision: From 3D Reconstruction to Recognition
    - Summary: This course introduces concepts and applications in computer vision, primarily dealing with geometry and 3D understanding.
    - Topics:
        - Cameras and projection models
        - Low-level image processing methods such as filtering and edge detection
        - Segmentation and clustering
        - Shape reconstruction from stereo
        - Depth estimation and optical/scene flow
        - 6D pose estimation
        - Object tracking
    - Prerequisites:
        - Python (and numpy), C/C++
        - Calculus, linear algebra, basic probability and statistics

Robotics:

- CS223A: Introduction to Robotics
    - Summary: This class introduces robotics methodologies and tools through topics like geometry, dynamics, and motion planning, using physical robots, simulations, and video examples to illustrate concepts and research applications.
    - Topics:
        - Design a robot with an optimal workspace
        - Model a robot to sufficient precision
        - Implement and tune a robot motion controller that exposes desired behavior

        - Implement and tune a compliant robot motion/force controller that exposes desired behavior
        - Implement and tune a vision-based robot motion controller that is robust to noise
        - Assess limitations of traditional, model-based approaches, visualize these failure cases, and propose a strategy on how they can be addressed (as assessed by bonus exercises in homework assignments)
    - Prerequisites: matrix algebra
    - Free Classes/Resources: Full lecture Series on Stanford Everywhere
- CS237A: Principles of Robot Autonomy I
    - Summary: This course teaches mobile autonomous robot perception, planning, and decision-making, covering topics like localization, motion planning, learning-based control, and uncertainty reasoning, with practical use of the Robot Operating System (ROS).
    - Topics:
        - Motion control
        - Perception, from classic to deep learning approaches
        - Localization and SLAM
        - Planning, decision-making, and system architecture
    - Prerequisites: Calculus, linear algebra, basic probability, and statistics
    - Free Classes/Resources:
- CS237B: Principles of Robot Autonomy II
    - Summary: This course teaches advanced principles for endowing mobile autonomous robots with capabilities to learn new skills autonomously and to interact with the environment and with human physically
    - Topics:
        - Reinforcement Learning and its relationship to optimal control
        - Contact and dynamics models for prehensile and non-prehensile robot manipulation,

        - Imitation learning and human intent inference
        - Different system architectures and their verification
    - Prerequisites:
        - Python 3/ PyCharm
        - Linear algebra, probability
        - Principles of Robot Autonomy I (AA274A) or equivalents

Natural language processing:

- CS224N: Natural Language Processing with Deep Learning
    - Summary: This course explores fundamental NLP concepts and advanced neural networks for language processing, focusing on deep learning techniques for tasks like word-level processing, question answering, and machine translation, culminating in a final project applying neural networks to a large-scale NLP problem.
    - Topics:
        - Computational properties of natural languages
        - Coreference, question answering, and machine translation
        - Processing linguistic information
        - Syntactic and semantic processing
        - Modern quantitative techniques in NLP
        - Neural network models for language understanding tasks
    - Prerequisites: Calculus and linear algebra
    - Free Classes/Resources: Stanford CS224N: NLP with Deep Learning | Winter 2021 | Lecture Series on YouTube

Reinforcement learning:

- CS234: Reinforcement Learning
    - Summary: This project-oriented class focuses on developing algorithms and systems for machine understanding of human language, covering

topics like semantics, relation extraction, sentiment analysis, and dialogue agents, with practical applications and industry connections.
    - Topics: Key ideas and techniques for RL
    - Prerequisites:
        - Python
        - Calculus, Linear Algebra, Basic Probability and Statistics (Gaussian distributions, mean, standard deviation, etc.)
        - Cost functions, taking derivatives and performing optimization with gradient descent, convex optimization.
    - Classes/Resources: Stanford CS234: Reinforcement Learning | Winter 2019 | Lecture Series on YouTube
- CS224R: Deep Reinforcement Learning
    - Summary: This course covers deep reinforcement learning algorithms, focusing on practical methods that use deep neural networks to learn behavior from experience and high-dimensional observations, considering the impact of decisions on the world.
    - Topics:
        - Methods for learning from demonstrations
        - Both model-based and model-free deep RL methods
        - Methods for learning from offline datasets and more advanced techniques for learning multiple tasks, such as goal-conditioned RL, meta-RL, and unsupervised skill discovery
    - Prerequisites:
        - PyTorch
        - Deep learning (backpropagation, convolutional networks, and recurrent neural networks)
        - RL and Markov decision processes (MDPs)
    - Classes/Resources:
        - Deep Learning and Reinforcement Learning on Coursera
        - Deep Reinforcement Learning in Python by DataCamp (paid)

Probabilistic graphical models:

- CS228: Probabilistic Graphical Models: Principles and Techniques
    - Topics:
        - Bayesian networks, undirected graphical models, and their temporal extensions
        - Exact and approximate inference methods
        - Estimation of the parameters and the structure of graphical models.
    - Prerequisites:
        - Basic probability theory and statistics
        - Programming, including algorithm design and analysis
    - Free Classes/Resources: Probabilistic Graphical Models Specialization on Coursera

Other:

- CS246: Mining Massive Data Sets
- CS330: Deep Multi-task and Meta Learning
- CS236: Deep Generative Models
- CS157: Computational Logic
- AA228: Decision Making Under Uncertainty

Additional Tips:

- Consistency is vital: Dedicate a specific time each day for studying.
- Take breaks: Avoid burnout by taking short breaks.

- Join online communities: Connect with other learners for support and collaboration.
- Build projects: Apply your knowledge by creating small projects.
- Stay motivated: Set achievable goals and celebrate your progress.

Remember, the actual time will vary depending on your learning pace and prior knowledge.

Good luck!