Dear student,

Welcome to University! As your professor, I am excited to guide you through the next four years of your educational journey. I have designed a comprehensive school program that will provide you with a well-rounded education and prepare you for your future career.

The following is an overview of the four-year school program:

**First Year:** In your first year, you will focus on building a strong foundation in computer science. You will take courses in programming fundamentals, data structures and algorithms, computer systems and organization, software engineering principles, and computer networks and security. You will also be introduced to mathematics and logic.

**Second Year:** In your second year, you will continue to build on your foundational knowledge while also exploring more advanced topics. You will take courses in artificial intelligence, computer graphics, databases, computer architecture, and computer operating systems. You will also have the opportunity to participate in a team-based project.

**Third Year**: In your third year, you will begin to specialize in a specific area of computer science. You will take courses in your chosen area of focus, such as robotics, machine learning, software development, or cybersecurity. You will also have the opportunity to take courses in other areas of interest and participate in an internship or co-op program.

**Fourth Year:** In your fourth year, you will continue to specialize in your chosen area of focus and take advanced courses in that field. You will also complete a capstone project that will allow you to apply your knowledge and skills to a real-world problem. You will also have the opportunity to participate in research or complete an additional internship or co-op program.

Throughout the four-year program, you will be assigned various assignments and projects that will help you build your skills and knowledge. You will also have the opportunity to attend lectures, participate in discussions and collaborate with fellow students.

As your professor, I will provide regular feedback on your progress and offer guidance and support as needed. It is important to prioritize your education and manage your time effectively to ensure successful completion of this program.

I look forward to helping you achieve your academic goals and prepare you for your future career.

Best regards, Professor at University

First Year:

Fall Semester:

* Programming Fundamentals I: introduction to computer programming
* Discrete Mathematics: logic and mathematical reasoning
* Calculus I: functions and limits
* Elective: an introductory course in a related field, such as physics or statistics

Learning Curve: This semester is designed to introduce you to the fundamentals of computer science, mathematics, and logic. The learning curve is steep but manageable with consistent effort and dedication.

Spring Semester:

* Programming Fundamentals II: data types, control structures, and functions
* Data Structures and Algorithms: fundamental data structures and algorithms
* Calculus II: integration and differentiation
* Elective: an introductory course in a related field, such as biology or economics

Learning Curve: This semester builds on the foundation laid in the fall semester, and the learning curve continues to be steep. Time management and consistent effort are key to success.

Second Year:

Fall Semester:

* Computer Systems and Organization: computer architecture, assembly language, and operating systems
* Artificial Intelligence: concepts and techniques in artificial intelligence
* Linear Algebra: matrices, vector spaces, and linear transformations
* Elective: a course in a related field, such as psychology or philosophy

Learning Curve: This semester focuses on advanced topics in computer science, and the learning curve is steeper than in the first year. It is important to stay organized and focused to manage the workload.

Spring Semester:

* Software Engineering: principles of software development and design
* Computer Graphics: principles of computer graphics and rendering
* Probability and Statistics: probability theory and statistical inference
* Elective: a course in a related field, such as sociology or linguistics

Learning Curve: This semester continues to explore advanced topics, and the learning curve remains steep. It is important to develop good study habits and time management skills.

Third Year:

Fall Semester:

* Elective: a course in your chosen area of focus, such as machine learning or cybersecurity
* Database Systems: principles of database design and implementation
* Computer Architecture: principles of computer architecture and design
* Elective: a course in a related field, such as economics or political science

Learning Curve: This semester allows you to specialize in your chosen area of focus and the learning curve becomes more manageable with a more specific focus.

Spring Semester:

* Elective: a course in your chosen area of focus, such as robotics or software development
* Operating Systems: principles of operating systems and design
* Elective: a course in a related field, such as art or music
* Elective: a course in a related field or a free elective

Learning Curve: This semester provides more opportunities to specialize in your chosen area of focus, and the learning curve continues to be manageable with consistent effort.

Fourth Year:

Fall Semester:

* Elective: a course in your chosen area of focus, such as cybersecurity or machine learning
* Capstone Project: apply your knowledge and skills to a real-world problem
* Elective: a course in a related field or a free elective
* Elective: a course in a related field or a free elective

Learning Curve: This semester provides opportunities to apply your knowledge and skills, and the learning curve is manageable with a more focused workload.

Spring Semester:

* Elective: a course in your chosen area of focus, such as robotics or software development
* Elective: a course in a related field or a free elective
* Elective: a course in a related field or a free elective
* Elective: a course in a related field or a free elective

Learning Curve: This semester allows for more flexibility in course selection, and the learning curve is manageable

# Year one

Week 1:

* Introduction to Programming Fundamentals I: In this course, we will learn the basics of programming using the Python programming language. To get started, I recommend the book "Python Crash Course" by Eric Matthes as a helpful resource. We will also be using the online platform Codecademy to practice and reinforce our skills.
* Discrete Mathematics: This course focuses on mathematical reasoning and logic. For a helpful introduction to the subject, I recommend the book "Discrete Mathematics and Its Applications" by Kenneth Rosen.
* Calculus I: This course covers the basics of calculus, including functions and limits. For a helpful resource, I recommend the book "Calculus: Early Transcendentals" by James Stewart.
* Elective: This semester, we will each choose an introductory course in a related field. For those interested in physics, I recommend the book "University Physics" by Francis Sears and Mark Zemansky.

Week 2:

* Programming Fundamentals I: We will continue to practice programming using Python and Codecademy. In addition, we will begin working on a small programming project to apply our skills.
* Discrete Mathematics: This week, we will cover propositional logic and truth tables. I recommend the online resource "Discrete Mathematics - Propositional Logic" by Tutorials Point as a helpful guide.
* Calculus I: We will continue to study functions, focusing on limits and continuity. I recommend the online resource "Calculus I - Limits" by Paul Dawkins as a helpful supplement.
* Elective: For those interested in statistics, I recommend the book "Statistics for Engineers and Scientists" by William Navidi.

Week 3:

* Programming Fundamentals I: We will continue to work on our programming project, applying our knowledge of control structures and functions.
* Discrete Mathematics: This week, we will cover predicate logic and quantifiers. I recommend the online resource "Discrete Mathematics - Predicate Logic" by Tutorials Point as a helpful guide.
* Calculus I: We will study derivatives and their applications, including optimization problems. I recommend the online resource "Calculus I - Derivatives" by Paul Dawkins as a helpful supplement.
* Elective: For those interested in economics, I recommend the book "Principles of Economics" by Gregory Mankiw.

Week 4:

* Programming Fundamentals I: We will continue to work on our programming project, practicing debugging and troubleshooting techniques.
* Discrete Mathematics: This week, we will cover sets and set operations. I recommend the online resource "Discrete Mathematics - Sets" by Tutorials Point as a helpful guide.
* Calculus I: We will study integrals and their applications, including area and volume problems. I recommend the online resource "Calculus I - Integrals" by Paul Dawkins as a helpful supplement.
* Elective: For those interested in biology, I recommend the book "Biology: A Global Approach" by Neil A. Campbell.

Week 5:

* Programming Fundamentals I: We will complete our programming project and share our code with the class for feedback and review.
* Discrete Mathematics: This week, we will cover functions and relations. I recommend the online resource "Discrete Mathematics - Functions" by Tutorials Point as a helpful guide.
* Calculus I: We will study techniques of integration, including substitution and integration by parts. I recommend the online resource "Calculus I - Techniques of Integration" by Paul Dawkins as a helpful supplement.
* Elective: For those interested in psychology, I recommend the book "Psychology: From Inquiry to Understanding" by Scott O. Lilienfeld.

Week 6:

* Programming Fundamentals I: This week, we will start Programming Fundamentals II, which builds on the skills we learned in Programming Fundamentals I. We will focus on more advanced topics such as object-oriented programming and data structures. To get started, I recommend the book "Python Programming: An Introduction to Computer Science" by John Zelle.
* Discrete Mathematics: This week, we will cover graphs and trees. I recommend the online resource "Discrete Mathematics - Graph Theory" by Tutorials Point as a helpful guide.
* Calculus I: We will study applications of integration, including arc length and surface area problems. I recommend the online resource "Calculus I - Applications of Integration" by Paul Dawkins as a helpful supplement.
* Elective: For those interested in computer graphics, I recommend the book "Computer Graphics: Principles and Practice" by James D. Foley.

Week 7:

* Programming Fundamentals II: We will continue to study object-oriented programming, including inheritance and polymorphism. We will also learn about common data structures such as lists, stacks, and queues.
* Discrete Mathematics: This week, we will cover algorithms and complexity theory. I recommend the book "Introduction to Algorithms" by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein.
* Calculus I: We will begin studying sequences and series. I recommend the online resource "Calculus II - Sequences and Series" by Paul Dawkins as a helpful supplement.
* Elective: For those interested in artificial intelligence, I recommend the book "Artificial Intelligence: A Modern Approach" by Stuart Russell and Peter Norvig.

Week 8:

* Programming Fundamentals II: This week, we will work on a programming project that incorporates the skills we have learned so far in Programming Fundamentals II. We will also learn about file input/output and exception handling.
* Discrete Mathematics: This week, we will continue our study of algorithms and complexity theory, focusing on big-O notation. I recommend the online resource "Discrete Mathematics - Big O Notation" by Tutorials Point as a helpful guide.
* Calculus I: We will continue to study sequences and series, including tests for convergence and divergence. I recommend the online resource "Calculus II - Sequences and Series" by Paul Dawkins as a helpful supplement.
* Elective: For those interested in robotics, I recommend the book "Robotics, Vision and Control: Fundamental Algorithms in MATLAB" by Peter Corke.

Week 9:

* Programming Fundamentals II: We will continue to work on our programming project, focusing on efficient algorithms and data structures.
* Discrete Mathematics: This week, we will cover recursion and induction. I recommend the online resource "Discrete Mathematics - Recursion and Induction" by Tutorials Point as a helpful guide.
* Calculus I: We will study power series and Taylor series. I recommend the online resource "Calculus II - Power Series and Taylor Series" by Paul Dawkins as a helpful supplement.
* Elective: For those interested in game development, I recommend the book "Game Programming Patterns" by Robert Nystrom.

Week 9:

* Algorithms and Data Structures: This week, we will continue our study of algorithms and data structures. We will focus on more advanced topics such as graph algorithms, dynamic programming, and NP-completeness. I recommend the book "Introduction to Algorithms" by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein.
* Probability and Statistics: We will also continue with probability and statistics, specifically exploring random variables and probability distributions. I recommend the book "Probability and Statistics for Engineering and the Sciences" by Jay L. Devore.
* Programming Lab: In the programming lab, we will work on implementing algorithms and data structures in practice. This week, we will focus on dynamic programming and graph algorithms.
* Assignment: You will be given a programming assignment to implement a graph algorithm using dynamic programming.

Week 10:

* Operating Systems: This week, we will begin our study of operating systems. We will learn about processes, threads, scheduling, and memory management. I recommend the book "Operating System Concepts" by Abraham Silberschatz, Peter B. Galvin, and Greg Gagne.
* Computer Networks: We will also begin our study of computer networks. We will learn about network protocols, layered architectures, and network topologies. I recommend the book "Computer Networking: A Top-Down Approach" by James F. Kurose and Keith W. Ross.
* Programming Lab: In the programming lab, we will work on implementing processes and threads in practice.
* Assignment: You will be given a programming assignment to implement a basic process scheduling algorithm.

Welcome everyone to the first day of the school program at University. I'm excited to have you all here with me and I hope you're ready to learn.

Today, we'll be introducing the course syllabus and expectations for the semester. We'll cover topics such as programming fundamentals, data structures, and algorithms. We'll also discuss the resources available to you and how to make the most of your time here at University.

For today's reading assignment, I recommend that you review Chapter 1 of "Introduction to Algorithms" by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. It's an essential book for any computer science student and will provide a solid foundation for the topics we'll be covering this semester.

Additionally, please make sure to review the course policies and expectations outlined in the syllabus. I will be available for any questions or concerns you may have throughout the semester, so please don't hesitate to reach out.

Let's get started with the first topic, programming fundamentals. In this class, we'll be focusing on programming in Python, which is a popular language used in many industries today. Python is known for its simplicity and readability, which makes it a great language for beginners to learn.

To start off, we'll be covering the basics of Python syntax, such as variables, data types, loops, and conditionals. We'll also be discussing how to write functions and modules in Python.

For this week's reading assignment, I recommend that you review Chapters 1-3 of "Python for Everybody" by Charles Severance. This book provides a great introduction to Python and will help you get up to speed quickly.

In addition to reading the assigned material, I encourage you to practice writing Python code on your own. There are many online resources and tutorials available, such as Codecademy and W3Schools, that can help you learn and practice Python programming.

Functions are reusable blocks of code that perform specific tasks. They can help simplify your code and make it easier to read and debug. We'll be covering how to write and call functions in Python, as well as how to use parameters and return values.

Modules, on the other hand, are collections of functions and classes that can be imported into your Python code. They allow you to reuse code across multiple projects and make your code more modular and organized. We'll be discussing how to import modules, as well as how to create your own custom modules.

Review Chapters 4-6 of "Python for Everybody" by Charles Severance. These chapters cover functions and modules in more detail and will provide you with a solid foundation for writing and using Python code.

In addition to reading the assigned material, practice writing your own functions and modules. You can try creating a custom module that contains functions related to a specific task or problem, such as file manipulation or data analysis.

Function is a block of code that performs a specific task. It can take inputs (arguments) and can also return outputs.

In Python, you define a function using the def keyword followed by the function name and a set of parentheses that may contain parameters. Here's an example:

def greet(name):

print("Hello, " + name + "!")

This function takes a name parameter and prints a greeting message to the console. To call this function, you simply need to pass in an argument for the name parameter:

greet("John")

This will output Hello, John! to the console.

Modules. A module is a file containing Python definitions and statements. You can use modules to organize your code into reusable components.

Python comes with many built-in modules that you can use, such as math for mathematical functions and random for generating random numbers. You can also create your own modules by creating a new Python file with functions or classes and then importing that file into your main program.

To import a module, you use the import keyword followed by the module name. Here's an example:

import math

print(math.pi)

This will import the math module and print the value of pi to the console.

You can also import specific functions or variables from a module using the from keyword. Here's an example:

from math import sqrt

print(sqrt(4))

This will import the sqrt function from the math module and print the square root of 4 to the console.

Assignments 1:

1. Write a function that takes in a string as input and returns the string reversed. For example, if the input is "hello", the output should be "olleh".
2. Write a function that takes in a list of numbers and returns the sum of all the even numbers in the list.
3. Create a module with at least three functions that perform different tasks, such as sorting a list or generating a random number.

Learning resources:

1. Read the Python documentation on functions and modules to gain a deeper understanding of how they work: <https://docs.python.org/3/tutorial/controlflow.html#defining-functions> and <https://docs.python.org/3/tutorial/modules.html>
2. Complete the functions exercises on the Practice Python website: <https://www.practicepython.org/exercises/>
3. Watch some tutorial videos on functions and modules in Python, such as this one on YouTube: <https://www.youtube.com/watch?v=9Os0o3wzS_I>

Some basic operations on lists:

1. Creating a list: You can create a list by enclosing a comma-separated sequence of elements within square brackets. For example, to create a list of numbers, you can write:

numbers = [1, 2, 3, 4, 5]

1. Accessing elements in a list: You can access elements in a list by their index, which starts at 0. For example, to access the first element in the numbers list above, you can write:

first\_number = numbers[0]

1. Updating elements in a list: You can update an element in a list by assigning a new value to it. For example, to update the third element in the numbers list above, you can write:

numbers[2] = 10

1. Adding elements to a list: You can add elements to a list using the append() method. For example, to add the number 6 to the end of the numbers list above, you can write:

numbers.append(6)

1. Removing elements from a list: You can remove an element from a list using the remove() method. For example, to remove the number 3 from the numbers list above, you can write:

numbers.remove(3)

Now that we've covered the basics of lists, let's move on to some more advanced topics, such as nested lists and list comprehensions. But before we do that, I want you to practice working with lists by completing the following exercises:

1. Write a function that takes in a list of numbers and returns the sum of all the numbers in the list.
2. Write a function that takes in a list of strings and returns a new list containing only the strings that have length greater than 5.
3. Write a function that takes in a list of numbers and returns a new list containing only the even numbers in the list.

And for further reading and practice, here are some resources:

1. The Python documentation on lists: <https://docs.python.org/3/tutorial/datastructures.html#more-on-lists>
2. The Python documentation on list comprehensions: <https://docs.python.org/3/tutorial/datastructures.html#list-comprehensions>
3. The Python documentation on nested lists: <https://docs.python.org/3/tutorial/datastructures.html#nested-list-comprehensions>

# Year Two

Semester 1:

Week 1:

* Database Systems: In this week, we will start our study of database systems. We will cover data modeling, SQL, and database design. I recommend the book "Database System Concepts" by Abraham Silberschatz, Henry F. Korth, and S. Sudarshan.
* Computer Security: We will also begin our study of computer security. We will learn about authentication, access control, and cryptography. I recommend the book "Computer Security: Principles and Practice" by William Stallings and Lawrie Brown.
* Programming Lab: In the programming lab, we will work on implementing SQL queries and exploring database design.
* Assignment: You will be given a programming assignment to create a database schema and implement a set of queries to manipulate the data.

Week 2:

* Software Engineering: This week, we will begin our study of software engineering. We will cover the software development lifecycle, software requirements, and software design. I recommend the book "Software Engineering: A Practitioner's Approach" by Roger S. Pressman.
* Computer Vision: We will also begin our study of computer vision. We will learn about image processing, feature detection, and object recognition. I recommend the book "Computer Vision: Algorithms and Applications" by Richard Szeliski.
* Programming Lab: In the programming lab, we will work on implementing a software design using object-oriented programming principles.
* Assignment: You will be given a programming assignment to implement an object-oriented design for a software application.

Week 3:

* Artificial Intelligence: In this week, we will start our study of artificial intelligence. We will cover search algorithms, game playing, and logic. I recommend the book "Artificial Intelligence: A Modern Approach" by Stuart Russell and Peter Norvig.
* Computer Graphics: We will also begin our study of computer graphics. We will learn about rasterization, rendering, and 3D modeling. I recommend the book "Computer Graphics: Principles and Practice" by James D. Foley, Andries van Dam, Steven K. Feiner, and John F. Hughes.
* Programming Lab: In the programming lab, we will work on implementing search algorithms and game playing strategies.
* Assignment: You will be given a programming assignment to implement a search algorithm for a game.

Week 4:

* Operating Systems: This week, we will continue our study of operating systems, focusing on virtual memory and file systems. I recommend the book "Operating System Concepts" by Abraham Silberschatz, Peter B. Galvin, and Greg Gagne.
* Distributed Systems: We will also begin our study of distributed systems. We will learn about network protocols, distributed file systems, and distributed consensus algorithms. I recommend the book "Distributed Systems: Concepts and Design" by George Coulouris, Jean Dollimore, and Tim Kindberg.
* Programming Lab: In the programming lab, we will work on implementing virtual memory and file systems in practice.
* Assignment: You will be given a programming assignment to implement a basic distributed file system.

Week 4

* Operating Systems: This week, we will continue our study of operating systems, focusing on virtual memory and file systems. I recommend the book "Operating System Concepts" by Abraham Silberschatz, Peter B. Galvin, and Greg Gagne.
* Distributed Systems: We will also begin our study of distributed systems. We will learn about network protocols, distributed file systems, and distributed consensus algorithms. I recommend the book "Distributed Systems: Concepts and Design" by George Coulouris, Jean Dollimore, and Tim Kindberg.
* Programming Lab: In the programming lab, we will work on implementing virtual memory and file systems in practice.
* Assignment: You will be given a programming assignment to implement a basic distributed file system.

Week 5:

* Programming Languages: This week, we will begin our study of programming languages. We will cover language syntax, semantics, and type systems. I recommend the book "Programming Language Pragmatics" by Michael L. Scott.
* Machine Learning: We will also begin our study of machine learning. We will learn about supervised and unsupervised learning, and explore some common machine learning algorithms. I recommend the book "Pattern Recognition and Machine Learning" by Christopher M. Bishop.
* Programming Lab: In the programming lab, we will work on implementing a programming language interpreter.
* Assignment: You will be given a programming assignment to implement a simple machine learning algorithm, such as linear regression or k-nearest neighbors.

Week 6:

* Computer Networks: In this week, we will start our study of computer networks. We will cover network architecture, protocols, and security. I recommend the book "Computer Networks" by Andrew S. Tanenbaum and David J. Wetherall.
* Natural Language Processing: We will also begin our study of natural language processing. We will learn about language models, parsing, and sentiment analysis. I recommend the book "Speech and Language Processing" by Daniel Jurafsky and James H. Martin.
* Programming Lab: In the programming lab, we will work on implementing a basic network protocol.
* Assignment: You will be given a programming assignment to implement a natural language processing algorithm, such as a part-of-speech tagger or sentiment analyzer.

Week 7:

* Theory of Computation: This week, we will start our study of the theory of computation. We will cover automata theory, formal languages, and computability. I recommend the book "Introduction to the Theory of Computation" by Michael Sipser.
* Distributed Databases: We will also begin our study of distributed databases. We will learn about distributed query processing, data replication, and fault tolerance. I recommend the book "Distributed Databases" by Stefano Ceri and Giuseppe Pelagatti.
* Programming Lab: In the programming lab, we will work on implementing a formal language recognizer.
* Assignment: You will be given a programming assignment to implement a basic distributed database system.

Week 8:

* Human-Computer Interaction: This week, we will begin our study of human-computer interaction. We will learn about user interfaces, usability testing, and interaction design. I recommend the book "Designing Interactive Systems" by David Benyon.
* Cloud Computing: We will also begin our study of cloud computing. We will learn about virtualization, infrastructure as a service, and platform as a service. I recommend the book "Cloud Computing: Concepts, Technology & Architecture" by Thomas Erl, Ricardo Puttini, and Zaigham Mahmood.
* Programming Lab: In the programming lab, we will work on designing and implementing a user interface.
* Assignment: You will be given a programming assignment to implement a simple cloud

Week 8-9: Operating Systems and Computer Networks

* In these two weeks, we will focus on understanding the basics of operating systems and computer networks.
* Topics covered will include: Processes and threads, CPU scheduling, Memory management, File systems, Introduction to computer networks, Network architecture and protocols, Network applications and services, Network security.
* Recommended resources:
  + "Operating System Concepts" by Silberschatz, Galvin, and Gagne
  + "Computer Networking: A Top-Down Approach" by Kurose and Ross

Week 10-11: Software Engineering

* In these two weeks, we will explore the software engineering process, from requirements gathering to deployment.
* Topics covered will include: Software development methodologies, Requirements gathering and analysis, Software design principles, Coding and testing, Deployment and maintenance, Agile software development.
* Recommended resources:
  + "Software Engineering: A Practitioner's Approach" by Pressman and Maxim
  + "Agile Estimating and Planning" by Mike Cohn

Week 12-13: Database Systems

* In these two weeks, we will delve into the fundamentals of database systems.
* Topics covered will include: Database models, Relational algebra, SQL, Data normalization, Transaction management, Indexing and query optimization.
* Recommended resources:
  + "Database Systems: The Complete Book" by Garcia-Molina, Ullman, and Widom
  + "SQL Cookbook" by Anthony Molinaro

Week 14-15: Ethics in Computing

* In these two weeks, we will examine ethical issues that arise in computing and technology.
* Topics covered will include: Privacy and security, Intellectual property, Social media and online behavior, Autonomous systems, Bias and discrimination.
* Recommended resources:
  + "Ethics for the Information Age" by Michael J. Quinn
  + "Weapons of Math Destruction" by Cathy O'Neil

Week 16: Project Presentations and Reflection

* In this final week of the academic year, students will present their projects from the software engineering course and reflect on their learning experiences throughout the year.

# Year Three

Week 1-2: Algorithms and Complexity

* In these two weeks, we will study the fundamentals of algorithms and complexity analysis.
* Topics covered will include: Algorithm design techniques, Complexity analysis, Sorting and searching, Graph algorithms, Dynamic programming, NP-complete problems.
* Recommended resources:
  + "Introduction to Algorithms" by Cormen, Leiserson, Rivest, and Stein
  + "Algorithm Design" by Kleinberg and Tardos

Week 3-4: Artificial Intelligence and Machine Learning

* In these two weeks, we will explore the basics of artificial intelligence and machine learning.
* Topics covered will include: Machine learning basics, Supervised learning, Unsupervised learning, Neural networks, Deep learning, Natural language processing.
* Recommended resources:
  + "Artificial Intelligence: A Modern Approach" by Russell and Norvig
  + "Hands-On Machine Learning with Scikit-Learn and TensorFlow" by Aurélien Géron

Week 5-6: Distributed Systems

* In these two weeks, we will study distributed systems and their design principles.
* Topics covered will include: Distributed system models, Communication protocols, Naming and synchronization, Consistency and replication, Fault tolerance, Cloud computing.
* Recommended resources:
  + "Distributed Systems: Principles and Paradigms" by Andrew S. Tanenbaum and Maarten van Steen
  + "Cloud Computing: Concepts, Technology & Architecture" by Thomas Erl, Zaigham Mahmood, and Ricardo Puttini

Week 7-8: Computer Graphics and Visualization

* In these two weeks, we will explore the basics of computer graphics and visualization.
* Topics covered will include: Graphics primitives and APIs, 2D and 3D transformations, Viewing and projection, Color models and shading, Texture mapping, Data visualization techniques.
* Recommended resources:
  + "Computer Graphics: Principles and Practice" by Foley, van Dam, Feiner, and Hughes
  + "Interactive Data Visualization for the Web" by Scott Murray

Week 9-10: Web Application Development

* In these two weeks, we will study the development of web applications.
* Topics covered will include: Web application architecture, Client-side and server-side scripting, Database connectivity, Web security, Scalability, Web application testing and deployment.
* Recommended resources:
  + "Web Development with Node and Express" by Ethan Brown
  + "Learning Web App Development" by Semmy Purewal

Week 11-12: Mobile Application Development

* In these two weeks, we will explore the development of mobile applications.
* Topics covered will include: Mobile application architecture, User interface design, Platform-specific development, Mobile database connectivity, Mobile application testing and deployment.
* Recommended resources:
  + "Android Programming: The Big Nerd Ranch Guide" by Bill Phillips, Chris Stewart, and Kristin Marsicano
  + "iOS Programming: The Big Nerd Ranch Guide" by Joe Conway and Aaron Hillegass

Spring Semester:

Week 1-2: Computer Architecture and Organization

* In these two weeks, we will study the design principles of computer architecture and organization.
* Topics covered will include: Computer organization, Instruction set architecture, Processor design, Memory hierarchy, I/O systems, Multiprocessor systems.
* Recommended resources:
  + "Computer Organization and Design" by Patterson and Hennessy
  + "Structured Computer Organization" by Andrew S. Tanenbaum

Week 3-4:

* Topics: Artificial Intelligence and Machine Learning
* Readings: "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow" by Aurélien Géron; "Artificial Intelligence: A Modern Approach" by Stuart Russell and Peter Norvig
* Assignments: Implement a machine learning algorithm using Scikit-Learn; write a research paper on the ethical considerations of artificial intelligence

Week 5-6:

* Topics: Robotics and Control Systems
* Readings: "Introduction to Robotics: Mechanics and Control" by John J. Craig; "Feedback Control of Dynamic Systems" by Gene F. Franklin, J. Da Powell, and Michael L. Workman
* Assignments: Build a robot using Arduino or Raspberry Pi and program it to perform a task; design and implement a feedback control system for a mechanical system

Week 7-8:

* Topics: Computer Networks and Security
* Readings: "Computer Networks" by Andrew S. Tanenbaum; "Network Security Essentials: Applications and Standards" by William Stallings
* Assignments: Build a simple network topology using Cisco Packet Tracer and configure its routing protocols; design and implement a secure communication protocol using encryption techniques

Week 9-10:

* Topics: Human-Computer Interaction and User Experience
* Readings: "The Design of Everyday Things" by Don Norman; "Don't Make Me Think: A Common Sense Approach to Web Usability" by Steve Krug
* Assignments: Design a user interface for a mobile or web application using Adobe XD or Figma; conduct a user study to evaluate the usability of the designed interface

Week 11-12:

* Topics: Cloud Computing and Big Data
* Readings: "Cloud Computing: Principles and Paradigms" by Rajkumar Buyya, James Broberg, and Andrzej Goscinski; "Big Data: Principles and Best Practices of Scalable Realtime Data Systems" by Nathan Marz and James Warren
* Assignments: Set up a cloud environment using Amazon Web Services or Microsoft Azure and deploy a web application; develop a data processing pipeline using Hadoop and Apache Spark

Learning Curve: Year Three is designed to deepen the student's knowledge in various fields of computer science and engineering, with a focus on advanced topics such as machine learning, robotics, computer networks, and cloud computing. The assignments are more complex and require significant effort and time investment from the students. Therefore, it is recommended that students start early on their assignments and seek help from their peers or instructors when needed.

# Year Four

Semester One:

Week 1-2:

* Topics: Distributed Systems and Parallel Computing
* Readings: "Distributed Systems: Concepts and Design" by George Coulouris, Jean Dollimore, and Tim Kindberg; "Programming Massively Parallel Processors: A Hands-on Approach" by David B. Kirk and Wen-mei W. Hwu
* Assignments: Implement a distributed system using Apache Hadoop or Apache Spark; develop a parallel program using OpenMP or CUDA

Week 3-4:

* Topics: Cybersecurity and Cryptography
* Readings: "Security Engineering: A Guide to Building Dependable Distributed Systems" by Ross Anderson; "Cryptography Engineering: Design Principles and Practical Applications" by Niels Ferguson, Bruce Schneier, and Tadayoshi Kohno
* Assignments: Design and implement a secure web application using TLS and HTTPS; develop a secure communication protocol using cryptographic techniques

Week 5-6:

* Topics: Advanced Algorithms and Data Structures
* Readings: "Introduction to Algorithms" by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein; "The Algorithm Design Manual" by Steven S. Skiena
* Assignments: Solve a complex algorithmic problem using dynamic programming or divide-and-conquer techniques; implement a data structure such as a balanced tree or a priority queue

Week 7-8:

* Topics: Computer Vision and Image Processing
* Readings: "Computer Vision: Algorithms and Applications" by Richard Szeliski; "Digital Image Processing" by Rafael C. Gonzalez and Richard E. Woods
* Assignments: Develop an image processing pipeline using OpenCV or MATLAB; implement a computer vision algorithm such as object detection or feature extraction

Week 9-10:

* Topics: Natural Language Processing and Text Mining
* Readings: "Speech and Language Processing" by Daniel Jurafsky and James H. Martin; "Text Mining: Classification, Clustering, and Applications" by Ashok Srivastava and Mehran Sahami
* Assignments: Develop a text classification system using Naive Bayes or Support Vector Machines; implement a sentiment analysis algorithm using natural language processing techniques

Week 11-12:

* Topics: Software Engineering and Project Management
* Readings: "Software Engineering: A Practitioner's Approach" by Roger S. Pressman; "Agile Estimating and Planning" by Mike Cohn
* Assignments: Design and develop a software system using agile methodologies; manage a project using project management tools and techniques

Semester Two:

Week 1-2:

* Topics: Advanced Operating Systems and Virtualization
* Readings: "Modern Operating Systems" by Andrew S. Tanenbaum and Herbert Bos; "Virtualization: A Manager's Guide" by Dan Kusnetzky
* Assignments: Implement a virtual machine using VMware or VirtualBox; design and implement a kernel module for a Unix-like operating system

Week 3-4:

* Topics: Computer Graphics and Game Development
* Readings: "Computer Graphics: Principles and Practice" by John F. Hughes, Andries van Dam, Morgan McGuire, David F. Sklar, James D. Foley, Steven K. Feiner, and Kurt Akeley; "Game Engine Architecture" by Jason Gregory
* Assignments: Develop a real-time rendering engine using OpenGL or DirectX; design and implement a game using a game engine such as Unity or Unreal Engine

Week 5:

* In Computer Vision class, we will dive deeper into 3D computer vision, covering topics such as 3D reconstruction, stereo vision, and depth estimation. We will read and discuss research papers on these topics and work on a project to apply what we've learned to a real-world problem. Recommended reading: "Multiple View Geometry in Computer Vision" by Richard Hartley and Andrew Zisserman.
* In Software Engineering class, we will focus on software design patterns and their implementation in real-world projects. We will also learn about agile software development methodologies and how to apply them effectively. Recommended reading: "Design Patterns: Elements of Reusable Object-Oriented Software" by Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides.
* In Robotics class, we will explore advanced topics in robotics, such as robot control, motion planning, and manipulation. We will also learn about state-of-the-art research in the field and work on a project to develop a new robot control algorithm. Recommended reading: "Robot Modeling and Control" by Mark W. Spong, Seth Hutchinson, and M. Vidyasagar.

Week 6:

* In Natural Language Processing class, we will focus on advanced techniques for text processing, including neural network models such as recurrent neural networks and transformers. We will also learn about language models, sentiment analysis, and machine translation. Recommended reading: "Speech and Language Processing" by Dan Jurafsky and James H. Martin.
* In Distributed Systems class, we will explore the challenges and opportunities presented by distributed computing environments. We will learn about distributed algorithms, consistency models, and fault tolerance techniques. We will also work on a project to develop a distributed system that solves a real-world problem. Recommended reading: "Distributed Systems: Principles and Paradigms" by Andrew S. Tanenbaum and Maarten Van Steen.
* In Machine Learning Applications class, we will examine real-world applications of machine learning, such as image and speech recognition, recommendation systems, and fraud detection. We will also learn about ethical considerations in machine learning and how to design systems that are fair and unbiased. Recommended reading: "Python Machine Learning" by Sebastian Raschka and Vahid Mirjalili.