

Georgia Foundations of Artificial Intelligence

Syllabus

High School - One Year (120 - 140 Contact Hours)

Course Overview and Goals

The Georgia Foundations of Artificial Intelligence is the introductory course in the Artificial Intelligence pathway. This course explores the foundations of Artificial Intelligence in society and the workplace, including programming, data science, mathematical reasoning, creative problem solving, ethical reasoning, and real-world applications of Artificial Intelligence. Students will learn the basics of programming in Python and will apply their programming skills to fields where artificial intelligence is used.

Learning Environment

The course utilizes a blended classroom approach. The content is fully web-based, with students writing and running code in the browser. Teachers utilize tools and resources provided by CodeHS to leverage time in the classroom and give focused 1-on-1 attention to students. Each unit of the course is broken down into lessons. Lessons consist of video tutorials, short quizzes, example programs to explore, and written programming exercises.

Programming Environment

Students write and run python programs in the browser using the CodeHS editor.

More Information

Browse the content of this course at <https://codehs.com/course/17655/explore>.

Prerequisites

This course is designed for complete beginners with no previous background in computer science, information technology, or artificial intelligence.

Course Breakdown

Unit 1: What is Artificial Intelligence? (2 - 3 weeks/10-15 hours)

Students will learn what defines Artificial Intelligence, how it is used, how it plans to be used, and the social and ethical implications of its use in society. Students will develop a case study exploring an ethical issue in Artificial Intelligence, highlighting the competing arguments on both sides of the issue, and ultimately choosing a side in the debate.

Objectives / Topics Covered	<ul style="list-style-type: none">● What is Artificial Intelligence?● Types of Artificial Intelligence● The Ethics of Artificial Intelligence● Exploring an Ethical Issue in AI
-----------------------------	--

<p>Example Assignments / Labs</p>	<ul style="list-style-type: none"> ● Drawing with AI <ul style="list-style-type: none"> ○ Students play a game of pictictionary against one of their classmates and time how long it takes for their classmate to determine what the image is. Students then play the same game against an AI that predicts what a user is drawing. Students compare the time rates, and explore the value of these predictive systems. ● Ethical Roundtable <ul style="list-style-type: none"> ○ Students participate in a fishbowl activity, where students share their perspectives on a series of ethical questions in AI. Students must prepare answers and questions to these ethical problems, and reflect on their classmates' perspectives. ● Project: The Ethics of AI <ul style="list-style-type: none"> ○ Students pick a topic on a specific ethical issue in AI and explore the different sides of the ethical argument. Students must ultimately pick a side, and present their project to their classmates.
-----------------------------------	--

Unit 2: Data and Artificial Intelligence (1-2 weeks/5-10 hours)

Students learn how data is stored digitally and explore the relationship between AI and big data. Students also learn about spreadsheets and how they can use spreadsheet tools to manipulate and analyze data.

<p>Objectives / Topics Covered</p>	<ul style="list-style-type: none"> ● Intro to Digital Information ● Big Data and AI ● Spreadsheets <ul style="list-style-type: none"> ○ Organizing data by sorting columns ○ Using spreadsheet formulas ○ Creating data visualizations
<p>Example Assignments / Labs</p>	<ul style="list-style-type: none"> ● Fast Food Menu <ul style="list-style-type: none"> ○ Students explore how digital information is used to encode orders at restaurants into simple numbers. ● Spotify's Discover Weekly <ul style="list-style-type: none"> ○ Students learn about how Spotify uses big data to develop users' Discover Weekly playlist. ● Roller Coaster Data Exploration <ul style="list-style-type: none"> ○ Students use spreadsheet tools to analyze a roller coaster dataset and create a visualization that highlights relationships within the data.

Unit 3: Basic Python and Console Interaction (3 weeks/15 hours)

Students learn the basics of programming in Python, including printing, variables, user input, and mathematical operators, and string operators.

Objectives / Topics Covered	<ul style="list-style-type: none">● Printing● Variables● Types● User Input● Converting Input Types● Arithmetic Expressions● String Operators● Comments
Example Assignments / Labs	<ul style="list-style-type: none">● Printing<ul style="list-style-type: none">○ Print messages to the console● Variables<ul style="list-style-type: none">○ Create variables of different types, and print them to the console.● Types<ul style="list-style-type: none">○ Investigate the types of different variables○ Convert between types● Arithmetic Expressions & Converting Input Types<ul style="list-style-type: none">○ Age in One Year - Ask the user how old they are, and tell them how old they will be in one year○ Rectangle, part 1 - Make variables for length and width and compute area and perimeter○ Rectangle, part 2 - Ask the user for length and width and compute area and perimeter

Unit 4: Conditionals (2 weeks/10 hours)

Students learn about booleans, if statements, and comparison operators.

Objectives / Topics Covered	<ul style="list-style-type: none">● If Statements● Boolean Values● Logical Operators● Comparison Operators● Floating Point Numbers and “Equality”
Example Assignments / Labs	<ul style="list-style-type: none">● If statements and boolean values<ul style="list-style-type: none">○ Is it raining? - Write a program that uses a boolean variable to determine whether or not it is raining● Boolean operators, and expressions<ul style="list-style-type: none">○ Boolean variable - Take a variable and use it in an if statement○ Legally allowed to vote - User reports age and program tells them whether or not they can vote in the US○ Transaction - User reports balance and deposit/withdrawal, and program prints new balance or error○ Recipe - Ask the user for ingredients, amounts per serving, and number of servings, and report the total amount of each ingredient needed

Unit 5: Looping (2 weeks/10 hours)

Students learn the basics of while loops and for loops.

Objectives / Topics Covered	<ul style="list-style-type: none">• While Loops• For Loops• Break and Continue• Nested Control Structures
Example Assignments / Labs	<ul style="list-style-type: none">• While Loops<ul style="list-style-type: none">◦ Divisibility - Ask the user to enter a numerator and denominator, and re-prompt until the denominator is non-zero• For Loops<ul style="list-style-type: none">◦ Average test score - Compute the average of several test scores• Break and Continue<ul style="list-style-type: none">◦ Higher/ Lower - Ask the user to guess a particular number between 1 and 100. If the user's guess was too high or too low, they should be notified• Nested Control Structures<ul style="list-style-type: none">◦ Rolling Dice - Print out all combinations that can be made when 2 dice are rolled

Unit 6: Functions and Exceptions (3 weeks/15 hours)

Students learn the basics of functions and how to write functions with parameters and return values.

Objectives / Topics Covered	<ul style="list-style-type: none">• Functions• Namespaces• Parameters• Return Values• Exceptions
Example Assignments / Labs	<ul style="list-style-type: none">• Functions<ul style="list-style-type: none">◦ Raining cats and dogs - Write functions to print text art of a cat and a dog◦ Temperature converter - write functions to convert from Fahrenheit to Celsius and vice versa• Exceptions<ul style="list-style-type: none">◦ Temperature converter, part 2 - Add exception handling to your temperature conversion program• Putting it all together<ul style="list-style-type: none">◦ Enter a positive number - Make a function to repeatedly ask the user to enter a number until they enter a positive number

Unit 7: Project: Create a Chatbot (2 weeks/10 hours)

Students apply the Design Thinking Process and develop an informational, rule-based chatbot.

Objectives / Topics Covered	<ul style="list-style-type: none">● What is a chatbot?● Design Thinking Process overview● Empathy● Define● Ideate● Prototype● Test● Finalize
Example Assignments / Labs	<ul style="list-style-type: none">● What is a chatbot?<ul style="list-style-type: none">○ Students test a variety of chatbots to determine if they pass the Turing test.● Empathy<ul style="list-style-type: none">○ Students interview a peer or community member to learn more about their target chatbot user.● Prototype<ul style="list-style-type: none">○ Students create a prototype of an informational, rule-based chatbot on a topic of their choosing.

Unit 8: Strings (3 weeks/15 hours)

Students learn the basics of strings, how to manipulate strings with string methods, and how to loop over a string.

Objectives / Topics Covered	<ul style="list-style-type: none">● Indexing and Slicing● Math Operators on Strings● For Loops Over a String● String Methods
Example Assignments / Labs	<ul style="list-style-type: none">● Indexing<ul style="list-style-type: none">○ First character - write a function that takes a string and returns the first character○ All but the first character - write a function that takes a string and returns everything but the first character● Math operators and strings<ul style="list-style-type: none">○ Full name - write a function that takes two strings (a first name and a last name) and returns a full name as a single string○ Replace a letter - write a function that takes a string and returns a copy with the character at a particular index replaced with a dash● For loops on strings<ul style="list-style-type: none">○ Count occurrences - write a function that takes two strings and returns the number of times the second string appears in the first string● String methods<ul style="list-style-type: none">○ Remove all from string - write a function that takes two strings and returns a string that consists of the first string with all instances of the second string removed

Unit 9: Creating and Altering Data Structures (3 weeks/15 hours)

Students learn about basic data structures and how to add, remove, and manipulate data.

Objectives / Topics Covered	<ul style="list-style-type: none">• Tuples• Lists• For Loops and Lists• List Methods• 2D Lists
Example Assignments / Labs	<ul style="list-style-type: none">• Tuples<ul style="list-style-type: none">○ Cookout Orders - Given a tuple of food orders, add up the number of burgers and number of hotdogs and print the total sums.• Lists<ul style="list-style-type: none">○ Listed Greeting - Ask a user to enter their name, age, and favorite sport, then split their response into list elements and use index values to greet them by name and respond that you enjoy that sport as well!○ Exclamation Points - Ask the user for a string and then print the same string with every lowercase i replaced with an exclamation point.○ Librarian - Ask the user for the last names of the authors of the five books they are returning. Print a list of those names in sorted order.

Unit 10: Project: AI and Gaming (2 weeks/10 hours)

Students learn about how AI is used in gaming and create a Tic Tac Toe console program.

Objectives / Topics Covered	<ul style="list-style-type: none">• AI and Gaming• Nonplayer Character
Example Assignments / Labs	<ul style="list-style-type: none">• Finite States<ul style="list-style-type: none">○ Students explore a simple game with a non-player character game and analyze the finite state machine algorithm that is being used.• Tic Tac Toe Game<ul style="list-style-type: none">○ Create the Board: Students use a 2D list to create a tic tac toe board.○ Take a Turn: Students program each player's turn by adding the player's token to the correct place in the 2D list.○ Random NPC: Students add a random non-player character that places their token in a random location on the board.

Unit 11: Computer Science Careers (1 weeks/5 hours)

Students explore a variety of computer science careers and learn about important work readiness traits.

Objectives / Topics Covered	<ul style="list-style-type: none">● Variety of computer science careers● Student Organizations● Workplace Readiness<ul style="list-style-type: none">○ Integrity○ Professionalism○ Valuing diversity○ Time Management
Example Assignments / Labs	<ul style="list-style-type: none">● Computer science careers<ul style="list-style-type: none">○ Students read the article “5 Technologies That Will Change the Future of the Human Race” and research an emerging technology that interests them.● Student Organizations<ul style="list-style-type: none">○ Students explore a student organization, such as Future Business Leaders of America or Technology Student Association.● Workplace Readiness<ul style="list-style-type: none">○ Students attempt to complete a set of programming tasks within a 10 minute timeframe. The catch - there are too many tasks to complete in the time limit, so students have to prioritize which tasks to complete. Students then reflect on their approach.