



CodeHS

Introduction to Computer Science with Python Course Syllabus

Course Overview and Goals

The CodeHS Introduction to Computer Science in Python course teaches the fundamentals of computer programming as well as some advanced features of the Python language. Students will develop an appreciation for how computers store and manipulate information by building simple console-based games. This course is equivalent to a semester-long introductory Python course at the college level.

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, adding up to over 100 hours of hands-on programming practice in total. Each unit ends with a comprehensive unit test that assesses student's mastery of the material from that unit.

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/626>

Prerequisites

This course does not assume any prior programming experience. Learn the basics of programming, and then gradually harness the power of some of Python's more advanced features to make games and solve real-world problems.

Students can take this course as their first introduction to computer science or as a secondary introductory course that explores a new language, after our Introduction to Computer Science in JavaScript.

Course Breakdown

Unit 1: Introduction to Programming in Python with Tracy the Turtle (3 weeks)

Objectives / Topics Covered	<ul style="list-style-type: none">● Commands● Colors● Loops● Functions
Assignments / Labs	<ul style="list-style-type: none">● 17 exercises in total● Example exercises:<ul style="list-style-type: none">○ Draw simple shapes with different colors○ Reduce redundancy in your program using loops and functions

Unit 2: Basic Python and Console Interaction (3 weeks)

Objectives / Topics Covered	<ul style="list-style-type: none">● Printing● Debugging● Variables● Types● Arithmetic expressions● User input● Converting input from a string
Assignments / Labs	<ul style="list-style-type: none">● 12 exercises in total● Example exercises:<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○ Input/output, arithmetic expressions<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■ Quadratic formula - ask the user for the

	coefficients of a quadratic equation, and compute the solutions
--	---

Unit 3: Control Flow (10 weeks)

Objectives / Topics Covered	<ul style="list-style-type: none"> ● Comments ● If statements ● Boolean values ● Logical operators ● Comparison operators ● Truth tables ● De Morgan’s Laws ● Short-circuit evaluation ● Floating point numbers and “equality” ● While loops ● For loops ● Nested control structures ● Functions ● Exceptions
Assignments / Labs	<ul style="list-style-type: none"> ● 30 exercises in total ● Example exercises: <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 ■ Old Enough 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 ○ Loops <ul style="list-style-type: none"> ■ Average test score - compute the average of several test scores ■ Divisibility - ask the user to enter a numerator and denominator, and re-prompt

	<ul style="list-style-type: none"> ○ until the denominator is non-zero ○ 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 4: Strings (3 weeks)

Objectives / Topics Covered	<ul style="list-style-type: none"> ● Indexing and slicing ● Math operators on strings ● For loops over a string ● String methods
Assignments / Labs	<ul style="list-style-type: none"> ● 11 exercises in total ● Example exercises: <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

	<ul style="list-style-type: none"> ○ String methods <ul style="list-style-type: none"> ■ Enthusiasm - write a function that takes a string and returns that string in all upper case ■ 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 5: Data Structures (6 weeks)

Objectives / Topics Covered	<ul style="list-style-type: none"> ● Tuples ● Lists ● Dictionaries ● 2D lists ● List comprehensions ● Packing and unpacking ● Mutable vs. immutable ● Equivalence vs. identity
Assignments / Labs	<ul style="list-style-type: none"> ● 15 exercises in total ● Example exercises: <ul style="list-style-type: none"> ○ Tuples <ul style="list-style-type: none"> ■ Point distance - represent two points as tuples, and make a function to compute the distance between them ○ Lists <ul style="list-style-type: none"> ■ Max in List- find the maximum value in a list ■ Slopes - ask for ten coordinate pairs, store the pairs as a list of tuples, and report the slope between each adjacent pair ○ Dictionaries <ul style="list-style-type: none"> ■ Phone book - user repeatedly enters name, and program either asks for the person's phone number or reports the phone number already provided ○ 2d lists <ul style="list-style-type: none"> ■ Checkerboard - write a program that prints the initial setup of a checkerboard, with a 1 where a piece would be and a 0 where a blank square would be

Unit 6: Hangman Project (3 weeks)

Objectives / Topics Covered	<ul style="list-style-type: none">● Allow students to combine a variety of topics (strings, loops, booleans, user input, etc.) in a single program● Introduce students to incremental development● Strengthen debugging skills by having students develop a larger project● Testing
Assignments / Labs	<ul style="list-style-type: none">● Part 1 - store a secret word in a variable, ask the user for a guess, and report whether or not it is correct● Part 2 - refine fetching guesses to check for invalid guesses and repeatedly ask until guess is valid● Part 3 - allow the user to guess 10 times, and print a combination of dashes and correct guesses before each guess● Part 4 - only penalize the user for incorrect guesses

Unit 7: Classes and Objects (6 weeks)

Objectives / Topics Covered	<ul style="list-style-type: none">● Classes● Attributes● Class variables vs. instance variables● Methods● Built-in methods● Composition, inheritance, and polymorphism● Private attributes
Assignments / Labs	<ul style="list-style-type: none">● 12 exercises in total● Example exercises:<ul style="list-style-type: none">○ Rectangle class<ul style="list-style-type: none">■ Capable of computing its area and perimeter○ Inheritance<ul style="list-style-type: none">■ Examine a program that has a Food class, a Vegetable class that inherits from Food, and a Broccoli class that inherits from Vegetable■ Predict what happens when instance and class variables are accessed in different ways

Unit 8: Who Said It? Project (3 weeks)

Objectives / Topics Covered	<ul style="list-style-type: none">● File reading● Allow students to combine a variety of topics in a single program● Re-emphasize incremental development● Strengthen debugging skills by having students develop a larger project● Testing
Assignments / Labs	<ul style="list-style-type: none">● Part 1 - Make a dictionary of word counts for a shortened version of <i>Pride and Prejudice</i> and <i>Hamlet</i>.● Part 2 - Test your code from Part 1 on the full texts of <i>Pride and Prejudice</i> and <i>Hamlet</i>.● Part 3 - Use your dictionary of word counts to predict whether a small text sample was written by Jane Austen or William Shakespeare. Test this prediction program on shortened versions of the texts.● Part 4 - Test your code from Part 3 on the full texts.