CodeHS

Introduction to Python Programming Course Syllabus
One Year for High School, 120 Hours

Course Overview and Goals

Introduction to Python Programming introduces students to the fundamentals of computer programming, with an emphasis on helping students develop logical thinking and problem-solving skills. Students begin by learning to design, code, and test their programs while applying mathematical concepts. Students then move to more advanced programming concepts and learn to create more powerful programs using functions, strings, data structures, file i/o operations, and objects.

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 120 hours of hands-on programming practice in total. Each unit ends with a comprehensive unit test that assesses students' mastery of the material from that unit as well as unit projects where students can display their understanding of the material.

Programming Environment: Students write and run Python programs in the browser using the CodeHS editor.

Technology Requirements: This course will require access to specific sites and tools listed in the linked google doc, each essential for course activities and assignments.

More information: Browse the content of this course at https://codehs.com/course/21076/overview

Prerequisites: Introduction to Python Programming is designed for complete beginners with no previous background in computer science. The course is highly visual, dynamic, and interactive, making it engaging for new coders.

Course Breakdown

Unit 1: Welcome (1 day/1 hour)
Students gain an understanding of what the course is about and reflect on what they hope to learn.

Unit 2: Karel in Python (3 weeks/15 hours)
Students learn the basics of programming by giving Karel the Dog commands in a grid world.

Browse the full content of this unit at https://codehs.com/course/21076/explore/module/29312

Objectives / Topics Covered	- Commands - Defining vs. Calling Functions - Designing Functions - Control Flow - Looping - Conditionals

	● Commenting Code ● Top Down Design

Assignments / Labs	● Program-specific tasks for Karel the Dog
	   ○ Example Exercise: Pyramid of Karel
	     Write a program to have Karel build a pyramid. There should be three balls on the first row, two in the second row, and one in the third row.
	
	● Teach Karel new commands like `turnRight()` or `makePancakes()`
	   ○ Example Exercise: Pancakes
	     Karel is the waiter. He needs to deliver a stack of pancakes to the guests on the 2nd, 4th, and 6th columns. Each stack of pancakes should have three pancakes. Create a function called `makePancakes()` to help Karel solve this problem. The world should end up exactly as shown here.
	
	● Solve large Karel problems by breaking them down into smaller, more manageable problems using Top Down Design
	   ○ Example Exercise: The Two Towers
	     In this program, Karel should build two towers of tennis balls. Each tower should be 3 tennis balls high. In the end, Karel should end up on top of the second tower, facing East.
	
	● Using control structures and conditionals to solve general problems
	   ○ Example Exercise: Random Hurdles
	     Write a program that has Karel run to the other side of the first row, jumping over all of the hurdles. Karel should only jump if there is a hurdle blocking the way. However, the hurdles can be in random locations. The world is fourteen columns long. You must write a function named `jumpHurdle()` as part of your solution.

Unit 3: Basic Python and Console Interaction (3 weeks/15 hours)
Students learn the basics of programming by writing programs that interact with users through the keyboard.

Browse the full content of this unit at https://codehs.com/course/21076/explore/module/29313

Objectives / Topics Covered	● Printing ● Variables ● Types ● User Input ● Converting Input Types ● Arithmetic Expressions ● String Operators ● Comments

Example Assignments / Labs	● Printing
	   ○ Print messages to the console
	● Variables
	   ○ Create variables of different types, and print them to the console.
	● Types
	   ○ Investigate the types of different variables
	   ○ Convert between types

	● Arithmetic Expressions & Converting Input Types
	   ○ 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: Project: Mad Libs (2 day/2 hours)

Students write a program that allows users to create their own Mad Lib stories using variables and user input.

Browse the full content of this unit at https://codehs.com/course/21076/explore/module/29314

Objectives / Topics Covered	● Variables ● User Input ● Print Statements
Example Assignments / Labs	● Create a Mad Libs game where users can input different types of words to complete a story. The program should prompt the user for various types of words (e.g., noun, adjective, verb) and then construct a story using the provided words.

Unit 5: Conditionals (2 weeks/10 hours)

Students teach their programs to make decisions based on the information it receives.

Browse the full content of this unit at https://codehs.com/course/21076/explore/module/29315

Objectives / Topics Covered	● If Statements ● Boolean Values ● Logical Operators ● Comparison Operators ● Floating Point Numbers and "Equality"

Example Assignments / Labs	● If statements and boolean values
	   ○ Is it raining? - Write a program that uses a boolean variable to determine whether or not it is raining
	● Boolean operators, and expressions
	   ○ Boolean variable - Take a variable and use it in an if statement
	   ○ Legally allowed to vote - User reports age and the program tells them whether or not they can vote in the US
	   ○ Transaction - The user reports balance and deposit/withdrawal, and the program prints a 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 6: Project: Quiz Game (2 days/2 hours)

Students write a program that quizzes users on a series of multiple-choice questions and checks their answers.

Browse the full content of this unit at https://codehs.com/course/21076/explore/module/29316

Objectives / Topics	● Variables

Covered	- Boolean Logic - Arithmetic Operators - Conditionals - Comparison Operators
Project Description	Create a simple multiple-choice quiz game using Python. The program should ask the user a series of questions and check their answers. After all the questions have been answered, the program should display the final score.

Unit 7: Looping (2 weeks/10 hours)

Students learn how to write more efficient code by using loops as shortcuts.

Browse the full content of this unit at https://codehs.com/course/21076/explore/module/29326

Objectives / Topics Covered	- While Loops - For Loops - Break and Continue - Nested Control Structures
Example Assignments / Labs	- While Loops - Divisibility - Ask the user to enter a numerator and denominator, and re-prompt until the denominator is non-zero - For Loops - Average test score - Compute the average of several test scores - Break and Continue - 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 - Rolling Dice - Print out all combinations that can be made when 2 dice are rolled

Unit 8: Project: Password Authenticator (2 day/2 hours)

Students write a program to provide feedback on whether the entered password is correct or incorrect.

Browse the full content of this unit at https://codehs.com/course/21076/explore/module/29317

Objectives / Topics Covered	- Variables - User Input - Boolean Operations - Loops - Control Structures
Project Description	Create a password authentication program that prompts the user to enter a password. The program should compare the entered password with a predefined correct password and provide feedback on whether the entered password is correct or incorrect. The program should allow the user to try again a limited number of times before locking them out.

Unit 9: Functions and Exceptions (1-2 week/5-8 hours)

Students learn how to decompose problems into smaller pieces that work together to solve a problem.

Objectives / Topics Covered	● Functions ● Namespaces ● Parameters ● Return Values ● Exceptions
Example Assignments / Labs	● Example exercises: 　○ Functions 　　■ 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 　　■ Temperature converter, part 2 - Add exception handling to your temperature conversion program 　○ Putting it all together 　　■ Enter a positive number - Make a function to repeatedly ask the user to enter a number until they enter a positive number

Unit 10: Strings (1-2 weeks/5-8 hours)

Students learn more sophisticated strategies for manipulating text in their programs.

Objectives / Topics Covered	● Indexing and Slicing ● Math Operators on Strings ● For Loops Over a String ● String Methods
Example Assignments / Labs	● Example exercises: 　○ Indexing 　　■ 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 　　■ 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 　　■ 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 　　■ Add 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 11: Project: Game of Pig (1 week/ 4 hours)

Students program a classic two-player game played with a six-sided die.

Browse the full content of this unit at https://codehs.com/course/21076/explore/module/29321

Objectives / Topics Covered	- Allow students to combine a variety of topics in a single program - Introduce students to incremental development - Strengthen debugging skills by having students develop a larger project - Team project development skills

Unit 12: Creating and Altering Data Structures (1-2 weeks/5-8 hours)

Students learn how tuples and lists are formed and the various methods that can alter them.

Browse the full content of this unit at https://codehs.com/course/21076/explore/module/29322

Objectives / Topics Covered	- Tuples - Lists - For Loops and Lists - List Methods
Example Assignments / Labs	- Example exercises: - Tuples - Cookout Orders - Given a tuple of food orders, add up the number of burgers and hotdogs and print the total sums. - Lists - 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! - Exclamat!on Po!nts - 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 13: Extending Data Structures (1-2 weeks/5-8 hours)

Students learn to build more complex programs that make use of grids and dictionaries.

Browse the full content of this unit at https://codehs.com/course/21076/explore/module/29323

Objectives / Topics Covered	- Dictionaries 2d lists - List comprehensions - Packing and unpacking - Mutable vs. immutable
Example Assignments	Example exercises:

/ Labs	○ Dictionaries 　■ Phone book - user repeatedly enters their name, and the program either asks for the person's phone number or reports the phone number already provided ○ 2d lists 　■ 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 14: Project: Guess the Word (1 week/ 4 hours)

Students write a program for a word guessing game.

Browse the full content of this unit at https://codehs.com/course/21076/explore/module/29324

Objectives / Topics Covered	● 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

Unit 15: File I/O (1-2 weeks/5-8 hours)

Students learn to read, write, and process information from text files.

Browse the full content of this unit at https://codehs.com/course/21076/explore/module/29325

Objectives / Topics Covered	● Reading from Files ● Writing to Files ● Processing File Data
Example Assignments / Labs	● Example exercises: ○ Reading from Files 　■ Validating Tweet Length - Write a function called that reads the contents of a text file tweet.txt and determines whether the text represents a valid tweet. ○ Write to Files 　■ Activity Tracker - Imagine you are building an activity tracker program. Your task is to write a program that logs a list of activities to a file.

Unit 16: Roles in a Software Development Team (1 week/5-6 hours)

Students learn the key roles and responsibilities of members of a software development team.

Browse the full content of this unit at https://codehs.com/course/21076/explore/module/29318

Objectives / Topics Covered	● Software Engineers ● Quality Assurance Engineers ● Designers ● Project Managers

| Example Assignments / Labs | Create a Mood Board In this assignment, you will act as a designer and create a mood board for a store of your choosing. To visually represent the brand and theme of the store, your mood board must include the following: 1. A color palette that best represents the store's brand 2. One or two fonts that align with the store's identity 3. Images related to the store's products or target audience Create a Task Board Imagine that you are a Project Manager. Before assigning work to members of the software development team, you need to create a list of tasks needed to create an application for the store you created a mood board for in the previous lesson. |

Introduction to Python Programming Supplemental Materials

Supplementary Units	Prerequisite/Recommended Unit(s)	Amount of Material
Assessment 1 (Can be used as a Midterm Exam)	Karel in Python, Basic Python and Console Interaction, Conditionals, Looping	1 quiz (offline materials found in resources)
Assessment 2 (Can be used as a Final Exam)	Complete all units in main course	1 quiz (offline materials found in resources)
Intro to Python with Turtle Graphics	Can be used in place of *Karel in Python* module	21 lessons
Python Graphics (Tkinter **or** Brython) Tkinter and Brython versions cover the same material, but Brython may be more accessible for students	Basic Python and Console Interaction, Conditionals, Looping, Functions and Exceptions	30 activities (Tkinter) 32 activities (Brython)
Classes and Objects	Complete all units in main course	10 lessons
Project: Mastermind	Complete all units in main course; can be used in place of *Project: Guess the Word*	7 activities
Additional Topics - Short Circuit Evaluation - DeMorgan's Laws	Short Circuit Evaluation & DeMorgan's Laws: Can be placed into Conditionals module, preferably after *Logical Operators* lesson	5 activities (Short Circuit Evaluation) 5 activities (DeMorgan's Laws)
Python Level 1 Certification Practice	Complete all units in main course	6 lessons