

SmashCode #2

Course Description: The CodeHS introduction to Computer Science in Python 3 curriculum teaches the foundations of computer science and basic programming, with an emphasis on helping students develop logical thinking and problem solving skills. Once students complete the CodeHS Introduction to Computer Science in Python course, they will have learned material equivalent to a semester college introductory course in Computer Science and be able to program using Python 3.

Representation across all Rings of Culture in programming is essential. Diversity in this field allows for effectively built programs that solve problems for a larger more diverse group, which in turn increases quality of life and profitability.

Grades: 9-12

Course Expectations & Rationale: This introductory computer science course will introduce students to the foundations of computer science and programming, emphasizing logical thinking and problem solving skills.

Course Resources & Materials: [CodeHS Intro to Computer Science in Python 3](#)

Course Essential Questions (5-10):

- How can we use computer programming to model and solve real world situations and problems?
- What makes a good program?
- How can computing and the use of computational tools foster creative expression?
- What does it mean to be literate in the 21st century?

Course Priority Standards (5-10):

DESE

9-10.AP.V.01 Create problem solutions that utilize primitive variables (e.g., strings, ints, booleans, doubles)

9-10.AP.C.01 Apply the concepts of specific control structures (e.g., sequence, conditionals, repetition, procedures) considering program efficiencies such as readability, performance and memory usage.

11-12.AP.M.01 Construct solutions to problems using student-created components (e.g., procedures, modules, objects).

11-12.AP.PD.05 Develop and use a series of test cases to verify that a program performs according to its design specifications.

CSTA

2-AP-11 Create clearly named variables that represent different data types and perform operations on their values.

3B-AP-12 Compare and contrast fundamental data structures and their uses.

3A-AP-23 Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.

2-AP-13 Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.

Course Enduring Understandings, Goals, & Objectives (5-10):

Enduring Understandings

- Computing enables people to use creative development processes to create computational artifacts for creative expressions or to solve a problem.
- Programs can be developed for creative expression, to satisfy personal curiosity, to create new knowledge, or to solve problems (to help people, organizations, or society).
- People write programs to execute algorithms.
- Programming is facilitated by appropriate abstractions.
- Programs are developed, maintained, and used by people for different purposes.
- Programming uses mathematical and logical concepts.

Course Essential Vocabulary: boolean, comparison operator, if statement, variables, logical operator, floating point numbers, loop, infinite loops, break, continue, control structure, function, exception, parameters, namespace, error, strings, array, slice, immutability, tuples, list, for loop, 2d list, comprehensions, pack (lists), dictionaries

Course “I Can...” Statements (5-10):

- I can create Booleans to test whether a condition is true or false.
- I can modularize my programs with functions.
- I can use indexing in order to find a specific character in a string.
- I can use various string methods to alter string values.
- I can create and store information in tuples.
- I can use 2d lists to store information in rows and columns.
- I can use dictionaries to structure data.
- I can use all concepts learned in this course to create a word guessing game in Python.

Unit 1	Conditionals (2 Weeks) In this unit, students teach their programs to make decisions based on the information it receives.
Unit 1 Big Ideas	<p>Conditional statements are also called decision-making statements. We use those statements while we want to execute a block of code when the given condition is true or false. Students teach their programs to make decisions based on the information it receives.</p> <p>Booleans refer to a value that is either true or false. Using comparison operators allows programs to make decisions. If/else statements allow programmers to use conditions to determine how their code should run. Logical operators give the ability to connect or modify Boolean expressions. When using a comparison operator with floating point numbers, you should use round(x) to avoid strange rounding behavior.</p>
Unit 1 Guiding & Essential Questions	<p>How can we use computer programming to model and solve real world situations and problems?</p> <p>What values can a boolean variable hold?</p> <p>What is an if/else statement?</p> <p>What is the difference between a single = and a double ==?</p> <p>What are the different logical operators?</p>
Unit 1 Student Learning Goal(s)	<ul style="list-style-type: none"> ● Students will create boolean variables to represent meaningful yes/no values ● Students will print out the value of a boolean variable ● Students will use if statements for control flow in their programs ● Students will explain the meaning of each of the comparison operators (<, <=, >, >=, ==, !=) ● Students will create programs using the comparison operators to compare values ● Students will predict the boolean result of comparing two values ● Students will describe the meaning and usage of each logical operator: or, and, and NOT (!) ● Students will construct logical statements using boolean variables and logical operators ● Students will print out the boolean result of comparing values ● Students will use floating point numbers ● Students will round values in their programs

Unit 1 Vocabulary	boolean, comparison operator, if statement, variables, logical operator, floating point numbers
Unit 1 DESE Missouri Learning Standards Computer Science Standards K-12	9-10.AP.V.01 create problem solutions that utilize primitive variables (e.g., strings, ints, booleans, doubles)
Unit 1 Computer Science Teacher Association K-12 Standards	2-AP-11 create clearly named variables that represent different data types and perform operations on their values.
	2-AP-13 decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.
	2-AP-14 create procedures with parameters to organize code and make it easier to reuse.
	2-AP-18 distribute tasks and maintain a project timeline when collaboratively developing computational artifacts.
	3A-AP-23 document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.
	3B-AP-12 compare and contrast fundamental data structures and their uses.
Unit 1 Assessments <i>(Minimum 1 Summative, 1 Formative)</i>	Each individual unit subcategory has one summative Check for Understanding as well as multiple formative options to demonstrate skills. Conditionals Quiz
Curricular Resources Utilized in Unit 1	CodeHS Intro to Computer Science in Python 3

Unit 2	Looping (2 Weeks) In this unit, students learn how to write more efficient code by using loops as shortcuts.
Unit 2 Big Ideas	Looping means repeating something over and over until a particular condition is satisfied. Students learn how to write more efficient code by using loops as shortcuts.
Unit 2 Guiding & Essential Questions	How can we use computer programming to model and solve real world situations and problems? What is a while loop? What is a for loop? What is an infinite loop?
Unit 2 Student Learning Goal(s)	<ul style="list-style-type: none"> • Students will effectively use while loops in their programs • Students will detect and resolve infinite loops • Students will implement for loops • Students will use the variable i as a counter • Students will control the values of i in a for loop • Students will explain the critical difference between break and continue • Students will describe why a break or continue statement would be needed in a coding scenario • Students will identify the different control structures that can be used to modify the flow of control through a program • Students will combine control structures to solve complicated problems • Students will choose the proper control structure for a given problem
Unit 2 Vocabulary	Loop, infinite loops, break, continue, control structure
Unit 2 DESE Missouri Learning Standards Computer Science Standards K-12	9-10.APC.01 apply the concepts of specific control structures (e.g., sequence, conditionals, repetition, procedures) considering program efficiencies such as readability, performance and memory usage.
	11-12.APC.01 trace the execution of iteration (e.g., loops, recursion), illustrating output and changes in values of named variables.
Unit 2 Computer Science Teacher Association	2-AP-11 create clearly named variables that represent different data types and perform operations on their values.

K-12 Standards	2-AP-12 design and iteratively develop programs that combine control structures, including nested loops and compound conditionals
	2-AP-13 decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.
	2-AP-18 distribute tasks and maintain a project timeline when collaboratively developing computational artifacts.
	3A-AP-23 document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.
	3B-AP-12 compare and contrast fundamental data structures and their uses.
Unit 2 Assessments <i>(Minimum 1 Summative, 1 Formative)</i>	Each individual unit subcategory has one summative Check for Understanding as well as multiple formative options to demonstrate skills. Looping Quiz
Curricular Resources Utilized in Unit 2	CodeHS Intro to Computer Science in Python 3

Unit 3	Functions and Exceptions (3 Weeks) In this unit, students learn how their programs can be decomposed into smaller pieces that work together to solve a problem.
Unit 3 Big Ideas	A function is a block of code that only runs when it is called. An exception is an event, which occurs during the execution of a program that disrupts the normal flow of the program's instructions. Students learn how their programs can be decomposed into smaller pieces that work together to solve a problem.
Unit 3 Guiding & Essential Questions	How can we use computer programming to model and solve real world situations and problems? Why is it important to break a program down into functions? What is the difference between a function with parameters, and a function without parameters? Why is it important to know if a variable is global or local?

Unit 3 Student Learning Goal(s)	<ul style="list-style-type: none"> • Students will modularize their programs with functions • Students will effectively use parameters to customize functions in their programs • Students will describe the different namespaces with regards to variables and functions • Students will remove complexity from their programs by abstracting with functions • Students will generalize their functions with parameters • Students will chain functions together using return values • Students will create programs that can gracefully handle exceptions • Students will continue to function when an error is raised
Unit 3 Vocabulary	function, exception, parameters, namespace, error
Unit 3 DESE Missouri Learning Standards Computer Science Standards K-12	9-10.AP.C.01 apply the concepts of specific control structures (e.g., sequence, conditionals, repetition, procedures) considering program efficiencies such as readability, performance and memory usage.
	9-10.A.P.M.01 break down a solution into procedures using systematic analysis and design utilizing functional abstraction.
	11-12.A.P.M.01 construct solutions to problems using student-created components (e.g., procedures, modules, objects).
Unit 3 Computer Science Teacher Association K-12 Standards	2-AP-11 create clearly named variables that represent different data types and perform operations on their values.
	2-AP-13 decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.
	2-AP-18 distribute tasks and maintain a project timeline when collaboratively developing computational artifacts.
	3A-AP-23 document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.
	3B-AP-12 compare and contrast fundamental data structures and their uses.
Unit 3 Assessments <i>(Minimum 1 Summative,</i>	Each individual unit subcategory has one summative Check for Understanding as well as multiple formative options to demonstrate skills.

1 Formative)	Functions and Exceptions Quiz
Curricular Resources Utilized in Unit 3	CodeHS Intro to Computer Science in Python 3

Unit 4	Strings (3 Weeks) In this unit, students use more sophisticated strategies for manipulating text in their programs - slicing, concatenating, and formatting.
Unit 4 Big Ideas	Strings are arrays of bytes representing Unicode characters. Students use more sophisticated strategies for manipulating text in their programs - slicing, concatenating, and formatting.
Unit 4 Guiding & Essential Questions	How can we use computer programming to model and solve real world situations and problems? What is indexing? What is slicing? How would you explain immutability? What is a substring? How can we search for all occurrences of the letter a in the string?
Unit 4 Student Learning Goal(s)	<ul style="list-style-type: none"> • Students will use indexing in order to find a specific character in a string • Students will use slicing to select a set of values from a string • Students will explain what immutability is and how this applies to strings in Python • Students will iterate over characters in a string using for loops • Students will use the in keyword to check if a character is in a string • Students will use various string methods to alter string values
Unit 4 Vocabulary	strings, array, slice, immutability
Unit 4 DESE Missouri Learning Standards Computer Science	9-10.AP.V.01 create problem solutions that utilize primitive variables (e.g., strings, ints, booleans, doubles)

Standards K-12	9-10.AP.C.01 apply the concepts of specific control structures (e.g., sequence, conditionals, repetition, procedures) considering program efficiencies such as readability, performance and memory usage.
Unit 4 Computer Science Teacher Association K-12 Standards	2-AP-11 create clearly named variables that represent different data types and perform operations on their values.
	2-AP-13 decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.
	2-AP-18 distribute tasks and maintain a project timeline when collaboratively developing computational artifacts.
	3A-AP-23 document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.
	3B-AP-12 compare and contrast fundamental data structures and their uses.
Unit 4 Assessments <i>(Minimum 1 Summative, 1 Formative)</i>	Each individual unit subcategory has one summative Check for Understanding as well as multiple formative options to demonstrate skills. Strings Quiz
Curricular Resources Utilized in Unit 4	CodeHS Intro to Computer Science in Python 3

Unit 5	Creating and Altering Data Structures (2 Weeks) In this unit, students learn how tuples and lists are formed and the various methods that can alter them.
Unit 5 Big Ideas	Data structures are “containers” that organize and group data according to type. Students learn how tuples and lists are formed and the various methods that can alter them.
Unit 5 Guiding & Essential Questions	How can we use computer programming to model and solve real world situations and problems? What does it mean that a tuple is heterogeneous? What are the characteristics of a list?

	<p>How can we use a loop to go through the items in a list?</p> <p>What is the find() method?</p>
Unit 5 Student Learning Goal(s)	<ul style="list-style-type: none"> ● Students will create and store information in tuples ● Students will explain the characteristics of a tuple ● Students will understand and explain the characteristics of a list ● Students will use lists to store and recall information ● Students will understand and explain the characteristics of a list ● Students will use for loops to go through items in a list ● Students will apply useful list methods to alter and access information about a list
Unit 5 Vocabulary	tuples, list, for loop
Unit 5 DESE Missouri Learning Standards Computer Science Standards K-12	9-10.AP.V.02 demonstrate the use of advanced variables (e.g., lists, arrays, objects) to simplify solutions, generalizing computational problems instead of repeatedly using primitive variables.
Unit 5 Computer Science Teacher Association K-12 Standards	2-AP-11 create clearly named variables that represent different data types and perform operations on their values.
	2-AP-12 design and iteratively develop programs that combine control structures, including nested loops and compound conditionals
	2-AP-13 decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.
	2-AP-18 distribute tasks and maintain a project timeline when collaboratively developing computational artifacts.
	3A-AP-14 use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables.
	3A-AP-23 document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.
	3B-AP-12 compare and contrast fundamental data structures and their uses.

Unit 5 Assessments <i>(Minimum 1 Summative, 1 Formative)</i>	Each individual unit subcategory has one summative Check for Understanding as well as multiple formative options to demonstrate skills. Creating and Altering Data Structures Quiz
Curricular Resources Utilized in Unit 5	CodeHS Intro to Computer Science in Python 3

Unit 6	Extending Data Structures (3 Weeks) In this unit, students learn to build more complex programs that make use of grids and dictionaries.
Unit 6 Big Ideas	The basic data structures in Python include list, set, tuples, and dictionary. Data structures are “containers” that organize and group data according to type. The data structures differ based on mutability and order. Mutability refers to the ability to change an object after its creation. Students learn to build more complex programs that make use of grids and dictionaries.
Unit 6 Guiding & Essential Questions	How can we use computer programming to model and solve real world situations and problems? Why are lists helpful? What are the benefits of using list comprehensions? What are the benefits of using a dictionary rather than a tuple or list? Why would it be helpful to know if values are equivalent?
Unit 6 Student Learning Goal(s)	<ul style="list-style-type: none"> ● Students will use 2d lists to store information in rows and columns ● Students will perform list comprehensions in order to alter all items in a list at once ● Students will pack and unpack lists in order to quickly and efficiently assign variables to list items ● Students will use dictionaries to structure data ● Students will predict if two values are identical and/or equivalent
Unit 6 Vocabulary	2d list, comprehensions, pack (lists), dictionaries
Unit 6 DESE Missouri Learning Standards Computer Science	9-10.AP.V.02 demonstrate the use of advanced variables (e.g., lists, arrays, objects) to simplify solutions, generalizing computational problems instead of repeatedly using primitive variables.

Standards K-12	
Unit 6 Computer Science Teacher Association K-12 Standards	2-AP-11 create clearly named variables that represent different data types and perform operations on their values.
	2-AP-12 design and iteratively develop programs that combine control structures, including nested loops and compound conditionals
	2-AP-13 decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.
	2-AP-18 distribute tasks and maintain a project timeline when collaboratively developing computational artifacts.
	3A-AP-23 document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.
	3A-AP-14 use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables.
	3B-AP-12 compare and contrast fundamental data structures and their uses.
Unit 6 Assessments <i>(Minimum 1 Summative, 1 Formative)</i>	Each individual unit subcategory has one summative Check for Understanding as well as multiple formative options to demonstrate skills. Extending Data Structures Quiz
Curricular Resources Utilized in Unit 6	CodeHS Intro to Computer Science in Python 3

Unit 7	Project Guess the Word (3 Weeks) In this unit, students use the skills they've learned throughout the course to build a word guessing game.
Unit 7 Big Ideas	This project is broken up into several pieces to guide students through the game development. Students use the skills they've learned throughout the course to build a word guessing game.

Unit 7 Guiding & Essential Questions	How can we use computer programming to model and solve real world situations and problems?
Unit 7 Student Learning Goal(s)	<ul style="list-style-type: none"> • Students will use all concepts learned in this course to create a word guessing game in Python.
Unit 7 Vocabulary	boolean, comparison operator, if statement, variables, logical operator, floating point numbers, loop, infinite loops, break, continue, control structure, function, exception, parameters, namespace, error, strings, array, slice, immutability, tuples, list, for loop, 2d list, comprehensions, pack (lists), dictionaries
Unit 7 DESE Missouri Learning Standards Computer Science Standards K-12	11-12AP.PD.05 develop and use a series of test cases to verify that a program performs according to its design specifications.
Unit 7 Computer Science Teacher Association K-12 Standards	2-AP-11 create clearly named variables that represent different data types and perform operations on their values.
	2-AP-12 design and iteratively develop programs that combine control structures, including nested loops and compound conditionals
	2-AP-13 decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.
	2-AP-18 distribute tasks and maintain a project timeline when collaboratively developing computational artifacts.
	3A-AP-14 use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables.
	3A-AP-16 design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions.
	3A-AP-17 decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.
	3A-AP-23 document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.

	3B-AP-12 compare and contrast fundamental data structures and their uses.
Unit 7 Assessments <i>(Minimum 1 Summative, 1 Formative)</i>	SmashCode #2 Final Exam
Curricular Resources Utilized in Unit 7	CodeHS Intro to Computer Science in Python 3