

# South Carolina Intermediate Computer Programming Syllabus

High School (120 Contact Hours)

## Course Overview and Goals

The South Carolina Intermediate Computer Programming course is designed to expand upon fundamental programming skills in Python. Topics include intermediate program design and development techniques, security and ethics, and practical experience in programming using a modern, text-based programming language. This course is fully aligned with the South Carolina Intermediate Computer Programming CTE course.

## 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 are made up of short video tutorials, example programs, quizzes, simulations, programming exercises, and free-response prompts. Each unit ends with a comprehensive unit quiz that assesses students' mastery of that unit's material.

## Programming Environment

Students write and run programs in the browser using the CodeHS editor. Students work with Python in the console for programming modules. They learn control structures, functions and parameters, data structures, classes, and objects. Students then expand their use of Python with a data science focus.

## More Information

Browse the content of this course at <https://codehs.com/course/23841/explore>

## Prerequisites

Students should complete the South Carolina Computer Programming 1 with Python course before enrolling in this course.

## Course Breakdown

### Module 1: Programming with Python (2 weeks/10 hours)

In this module, students build on the basics of Python and learn user input, arithmetic expressions, mouse events, and how to create graphics.

Topics Covered	<ul style="list-style-type: none"><li>● Variable Names</li><li>● Data Types</li><li>● Arithmetic Expressions</li><li>● Order of Operations</li><li>● Modulus</li><li>● String Concatenation</li><li>● User Input</li><li>● Program Output</li><li>● Using Python Graphics</li><li>● Mouse Events</li></ul>
Example Assignments	<ul style="list-style-type: none"><li>● Printing<ul style="list-style-type: none"><li>○ Print messages to the console</li></ul></li><li>● Variables<ul style="list-style-type: none"><li>○ Create variables of different types, and print them to the console.</li></ul></li><li>● Types<ul style="list-style-type: none"><li>○ Investigate the types of different variables</li><li>○ Convert between types</li></ul></li><li>● Graphics<ul style="list-style-type: none"><li>○ Students will create a snowman by creating circles of different sizes.</li></ul></li><li>● Mouse Events<ul style="list-style-type: none"><li>○ Students will create a program that draws a rectangle whenever they click the mouse.</li></ul></li></ul>

### Module 2: Python Control Structures (2 weeks/10 hours)

In this module, students learn how to use booleans and logical operators with control structures to create more advanced programs in Python.

Topics Covered	<ul style="list-style-type: none"><li>● Booleans</li><li>● Relational Operators</li><li>● Selection</li><li>● Conditional Statements</li><li>● Nested Conditionals</li><li>● Random Numbers</li><li>● Iteration</li><li>● Loops</li></ul>
Example Assignments	<ul style="list-style-type: none"><li>● Comparison Operators<ul style="list-style-type: none"><li>○ Students will write a program that stores three variables from the user: their points per game, their rebounds per game, and their assists per game, in that order.</li></ul></li></ul>

	<ul style="list-style-type: none"> <li>● While Loops <ul style="list-style-type: none"> <li>○ <i>Better Password Prompt</i>: Students write a program that uses a while loop to prompt a user for a password. They keep prompting the user for the password, and if they get it correct, they then break out of the loop. If they don't get it correct, they should give the user an error message. This activity requires that students use multiple program statements in a specific order to solve a problem.</li> </ul> </li> <li>● Challenge Activity <ul style="list-style-type: none"> <li>○ The computer picks a number between 1 and 100, and you have to guess it. The computer will tell you whether your guess was too high, too low, or correct. Your assignment is to generate a random number and let the user guess numbers until they guess the correct number.</li> </ul> </li> </ul>
--	---

### Module 3: Functions and Parameters (2 weeks/10 hours)

In this module, students will build on their knowledge of functions and learn to write reusable code using functions and parameters.

Topics Covered	<ul style="list-style-type: none"> <li>● Program Output</li> <li>● Procedures</li> <li>● Parameters</li> <li>● Return Values</li> <li>● Using Existing Algorithms</li> </ul>
Example Assignments	<ul style="list-style-type: none"> <li>● Functions and Return Values <ul style="list-style-type: none"> <li>○ <i>Pool Table</i>: Students write a program with a function that draws a pool ball. This function should take as parameters, the color, the number that should go on the pool ball, and the location of the center of the pool ball. Students need to consider the function abstractly as a means for taking specific data via the parameters and creating a unique graphical output based on those inputs.</li> </ul> </li> <li>● Challenge Activity <ul style="list-style-type: none"> <li>○ Write a program to draw ghosts on the screen. You must do this by writing a function called <code>draw_ghost`</code>, which takes three parameters, the x location of the ghost, the y location of the ghost, and the color of the ghost. x and y for the ghost define where the center of the head should go.</li> </ul> </li> </ul>

### Module 4: Basic Data Structures (2 weeks/10 hours)

In this module, students learn the basics of lists in Python while exploring tuples and lists. Students will explore how to create each of these data structures and the various methods to access or alter them.

Topics Covered	<ul style="list-style-type: none"> <li>● Data Values</li> <li>● Lists and Elements</li> <li>● Indices</li> <li>● List Procedures</li> <li>● Traversing a List</li> <li>● Iteration Statements</li> <li>● Using Existing Algorithms</li> </ul>
----------------	---

Example Assignments	<ul style="list-style-type: none"> <li>● For Loops and Lists <ul style="list-style-type: none"> <li>○ <i>Owls</i>: Your friend really likes talking about owls. Write a function <code>owl_count</code> that takes a block of text and counts how many words they say have word "owl" in them. Any word with "owl" in it should count, so "owls," "owlette," and "howl" should all count.</li> </ul> </li> <li>● List Methods <ul style="list-style-type: none"> <li>○ <i>Librarian, Part 2</i>: Students write a program to ask the user for an author's full name. Students will then use list procedures to split the full name into individual names and then slice the list to add the last name to a new list. Once the student has collected all of the last names, they will sort them and then print the results. This program development requires students to use user input data that can contain a variable number of names. The students must then use various list techniques to manipulate the data.</li> </ul> </li> </ul>
---------------------	---

### Module 5: Creative Development (3 weeks/15 hours)

In this module, students will learn the theory and practice of user interface design. They will create a program that will tell a story using Python graphics. This project allows students to think creatively about the applications of the concepts covered in the course, and create something of personal value.

Topics Covered	<ul style="list-style-type: none"> <li>● Design Thinking</li> <li>● Prototyping</li> <li>● Testing</li> <li>● Project Prep and Development</li> </ul>
Example Assignments	<ul style="list-style-type: none"> <li>● Development Process <ul style="list-style-type: none"> <li>○ How to Make Accessible Programs</li> <li>○ Iterative Development</li> <li>○ User Research <ul style="list-style-type: none"> <li>■ Diverse Perspectives</li> </ul> </li> <li>○ Online Collaboration Tools</li> </ul> </li> <li>● User Testing <ul style="list-style-type: none"> <li>○ Wizard of Oz Prototyping</li> <li>○ How to User Test Responses</li> </ul> </li> </ul>

### Module 6: Classes and Objects (4 weeks/20 hours)

In this module, students learn the principles of object-oriented design. They will use classes and objects and learn how to break code down in these structures.

Topics Covered	<ul style="list-style-type: none"> <li>● Classes and Objects</li> <li>● Methods</li> <li>● Overloading</li> <li>● Class vs. Instance Variables</li> <li>● Inheritance</li> <li>● Namespaces</li> <li>● Modules</li> </ul>
----------------	---

Example Assignments	<ul style="list-style-type: none"> <li>● Built-in Methods <ul style="list-style-type: none"> <li>○ Add a method that will allow instances of the Rectangle class to be considered equal if they have the same length and width. Create three different Rectangle objects. Two of the rectangles should have the same length and width. Compare each Rectangle object and print out the result.</li> </ul> </li> <li>● Class vs. Instance Variables <ul style="list-style-type: none"> <li>○ Most cars have four doors. Create a class variable, and set it to 4. Cars that have a different number of doors can set their doors variable as though it were an instance variable, without affecting the number of doors every other car has. Test this by modifying the code outside the class definition. Instead of changing and printing the variable, you should change and print each object's doors variable.</li> </ul> </li> </ul>
---------------------	---

### Module 7: Exploring Data Science with Python (4 weeks/20 hours)

Students will learn and apply the process of the data science life cycle. This includes asking statistical questions, collecting or obtaining reliable raw data, analyzing the data using measures of central tendency and spread and interpreting, and summarizing the results.

Topics Covered	<ul style="list-style-type: none"> <li>● What is Data Science?</li> <li>● Gathering Data <ul style="list-style-type: none"> <li>○ Quantitative/Qualitative</li> </ul> </li> <li>● Exploring Data Using Python</li> <li>● Modules and Libraries</li> <li>● Using the Pandas Library <ul style="list-style-type: none"> <li>○ Series <ul style="list-style-type: none"> <li>■ Measures of Central Tendency</li> <li>■ Measures of Spread</li> </ul> </li> <li>○ DataFrames <ul style="list-style-type: none"> <li>■ Selecting Columns</li> <li>■ Using Functions</li> </ul> </li> </ul> </li> </ul>
Example Assignments	<ul style="list-style-type: none"> <li>● Mini-Project: <ul style="list-style-type: none"> <li>○ Students will go through the first two steps of the data cycle using data of their choosing.</li> </ul> </li> <li>● Hot Dog Plots: <ul style="list-style-type: none"> <li>○ Use the correct Python functions to create a boxplot of the data. Using the graph, determine the summary statistics and the spread.</li> </ul> </li> <li>● Roller Coaster Rankings: <ul style="list-style-type: none"> <li>○ Define a function that will compute a score for each roller coaster. Use this function to store the results in a new column.</li> </ul> </li> <li>● Student Test Scores: <ul style="list-style-type: none"> <li>○ Create a function that finds the maximum test score between test one and test two for each student. Create a function that finds the maximum test score between all three tests for each student. Decide which calculations, along with these two new columns, can help you answer the original statistical question. Explore and further analyze your data until you come to a conclusion.</li> </ul> </li> </ul>

## Module 8: Security and Ethics (1 week/5 hours)

In this module, students learn about common cyber attacks, how coding and security are related, and how to cite quality resources online while following copyright laws.

Topics Covered	<ul style="list-style-type: none"><li>● Common Cyber Attacks and Prevention</li><li>● Databases</li><li>● Security in Coding</li><li>● Creative Credit and Copyright</li><li>● Ethics and Legal Considerations</li><li>● Responsibly Using Someone Else's Code</li><li>● Digital Footprint and Responsibility</li><li>● Can All Information Be Trusted?</li></ul>
Example Assignments	<ul style="list-style-type: none"><li>● Cyber Impacts<ul style="list-style-type: none"><li>○ Create a fictional cyber attack story. Feel free to be creative! You can create a story, a poem, or a comic book. End the story with a summary of consequences that the organization faced (if they were unprotected) or that they avoided (if they were protected).</li></ul></li><li>● XSS Tutorial<ul style="list-style-type: none"><li>○ In this activity, you will perform simulated Cross Site Scripting (XSS) attacks on vulnerable websites using unprotected input fields. As you run through this tutorial, think about how these sites may be able to better protect themselves against these types of attacks.</li></ul></li><li>● Digital Footprint and Responsibility<ul style="list-style-type: none"><li>○ Students explore the impact of social media and technology on teenagers, covering topics like digital footprints, the rise of social media screenings, cyberbullying, and the importance of updating privacy settings.</li></ul></li></ul>

## Module 9: Project: Harnessing Technology to Solve Problems (1 week/5 hours)

In this module, students will explore the systematic selection and integration of hardware and software components that represent a powerful approach to problem-solving in today's technology-driven world. Whether it's optimizing crop yields, enhancing athletic performance, or creating intelligent home systems, the synergy between hardware and software enables us to improve efficiency and address challenges.

Topics Covered	<ul style="list-style-type: none"><li>● Technological innovations</li><li>● Integrating hardware and software</li><li>● Researching and defining a problem</li><li>● Developing a system/device prototype</li></ul>
Example Assignments	<ul style="list-style-type: none"><li>● Choose a topic and research the problem.<ul style="list-style-type: none"><li>○ What is the root of the problem?</li><li>○ What are the needs of those experiencing the problem?</li><li>○ What factors contribute to the problem?</li></ul></li><li>● Plan out what hardware and software your device will need<ul style="list-style-type: none"><li>○ Will the hardware need to communicate with software or another device?</li><li>○ If your system will need data analytics, explain what type of analysis you need and why.</li></ul></li></ul>

	<ul style="list-style-type: none"> <li>● Draw a sketch of your prototype or create a physical prototype.</li> <li>● Pitch your innovation <ul style="list-style-type: none"> <li>○ A pitch refers to a persuasive presentation that attempts to convince others about the value, benefits, and feasibility of a product, service, idea, or project. The goal of a pitch is to capture the attention of the audience, generate interest, and ultimately secure buy-in, funding, or support.</li> </ul> </li> </ul>
--	---

### Module 10: The 21st Century Workplace (1 week/5 hours)

In this module, students learn basic safety inside the classroom and how this applies to the workplace. They also explore future career options and the skills needed to obtain these jobs.

Topics Covered	<ul style="list-style-type: none"> <li>● School and Classroom Safety</li> <li>● Workplace Safety</li> <li>● Skills for the 21st Century</li> <li>● Computer Science Careers</li> </ul>
Example Assignments	<ul style="list-style-type: none"> <li>● Students explore the workplace and the skills needed to succeed in computer science careers</li> <li>● Example exercises: <ul style="list-style-type: none"> <li>○ One Page Project on workplace safety</li> <li>○ Review 21st Century workplace skills and reflect on using them in the classroom</li> <li>○ Explore computer science careers and how computer science is used in other fields</li> </ul> </li> </ul>