



Teaching Tomorrow's Technology.

CompuScholar

Python Programming

Course Syllabus and Planner

For classrooms starting in the Fall of 2026 or later

Course Overview

CompuScholar's **Python Programming** curriculum is a one-year (two-semester) course that teaches students to code in the Python language. It is aligned to numerous state and national standards for courses such as "Computer Programming I" or similar titles. For details, please see our State Alignments and course description pages:

<https://www.compuscholar.com/schools/standards/states/>

<https://www.compuscholar.com/schools/courses/python/>

Other introductory programming courses are not required. Students merely need to have typical computer usage skills prior to starting this course.

First-semester topics are appropriate for middle school students seeking an introductory coding experience. Both semesters are appropriate for high school students.

Course Material

The course consists of the following student-facing elements:

- **Instructional Videos** – optional (not required), but enjoyed by many students as an audio-visual introduction and reinforcement of the lesson topics.
- **Lesson Text** – required reading, contains full topic details and live coding exercises
- **Quizzes and Exams** – multiple-choice and automatically graded by our system
- **Chapter Homework Exercises** – hands-on practice problems
- **Chapter Activities** – hands-on projects, automatically graded by our system

Teachers additionally have access to:

- **Teacher's Guides** – for each lesson, with suggested classroom discussion questions
- **Quiz and Exam Answer Keys** – PDFs for quick reference
- **Activity Solution Guides** – fully coded activity solutions for each chapter activity



Programming Environment and Device Requirements

CompuScholar provides an in-browser Python coding environment. This online feature may be used by students to complete all exercises and activities in all required chapters. When using our online coding environment:

- **No local software installation is needed.**
- **All activities can be completed from any web browser on any device (including Chromebooks and tablets).**

Optional chapters at the end of the course introduce the use of locally installed Python environments and Visual Studio Code as an external IDE. Visual Studio Code requires a **Windows** or **Mac OS computer** for installation. Teachers may use those optional chapters, if desired, to cover graphical projects. Students may also opt to use Visual Studio Code or other preferred 3rd party IDEs for all course coding exercises instead of using our online IDE.

Project Grading

Each chapter normally contains one or more hands-on, graded activities. The activities in **all required chapters are fully auto-graded by our system**. Teachers have complete control over the auto-graded results.

Some activities in later, optional chapters are free-form (creative) or completed in an external IDE. The teacher is responsible for grading those optional projects.

Course Navigation

Chapters 1 - 14 form the first semester and end in a mid-term project. These chapters are suitable as an introductory coding experience for middle school students. They are also a prerequisite for high school students to learn basic coding skills before continuing into the second semester.

Chapters 15 - 26 form the second semester and are generally required for high school students to meet the standards in most state computer science courses. Advanced topics include algorithms, sorting, Object-Oriented Programming (OOP), File I/O, data structures, and recursion. If a specific topic (e.g., recursion) is not required by your state or appropriate for your classroom, then teachers may choose to skip those chapters.



Teaching Tomorrow's Technology.

CompuScholar

Chapter 27 contains an **optional** team project that allows students to create unique work in teams or individually. Teachers may assign this project as desired or to meet state requirements.

Chapters 28 - 29 are **optional** and introduce locally installed Python environments, the Visual Studio Code IDE, and graphical applications.

Supplemental Chapters 1 - 4 contain a variety of **optional** topics that may be required by individual states to satisfy requirements for career exploration, computing in modern society, computer networking, and other enrichment topics. Teachers may **optionally** select any of these topics for students, time permitting.

Please refer to the specific computer science requirements for your state when selecting optional or supplemental topics. Our [State Alignments page](#) contains guidance for many states, or you can contact CompuScholar for additional help.

Supplemental Chapter 5 contains a variety of **optional** topics that are recommended for students preparing to take the **PCEP certification exam**. For details, please see our [Industry Certifications page](#).

Course Planner

The following pages contain a suggested timeline for completing course content over two semesters. A typical school year consists of 36 calendar weeks or 180 days of school. After completing the “core” content, most classes will have time left in the school year to explore optional and supplemental topics. Teachers may direct students to any appropriate topics, time permitting.

Each “day” listed below represents one typical day or class period of 45 – 60 minutes. In most cases, we anticipate students will complete one lesson per day (including the quiz), 1 day per lab, and 1 day per chapter test. Some classes may move faster or slower than the suggested pace.



Semester 1 Timeline

Days	CompuScholar Chapter	Activity Notes
6	Chapter 1: Computing Concepts <ul style="list-style-type: none">* Evolution of Computers* Survey of Computer Hardware* Introduction to Computer Software* Computer Ethics* Computer Security	Common curricular requirements
7	Chapter 2: Fundamentals of Python <ul style="list-style-type: none">* Common Programming Languages* Introduction to Python* Running Python Programs* Writing Python Code* Help and Reference Documentation	ACTIVITY: Study Planner Online, auto-graded
5	Chapter 3: Storing Data <ul style="list-style-type: none">* Variables* Data Storage and References* Combining and Printing Text	ACTIVITY: Gamer Profile Online, auto-graded
6	Chapter 4: Working with Strings <ul style="list-style-type: none">* Getting Console Input* String Basics* String Methods* Formatting Strings	ACTIVITY: Superhero Signal Online, auto-graded
5	Chapter 5: Working with Numbers <ul style="list-style-type: none">* Simple Math Operations* Operator Precedence* Type Casting and Truncation	ACTIVITY: Amusement Park Budget Online, auto-graded
5	Chapter 6: Math Concepts <ul style="list-style-type: none">* Random Numbers* The Math Library* Binary and Hexadecimal Numbers	ACTIVITY: Space Mission Online, auto-graded
5	Chapter 7: Debugging <ul style="list-style-type: none">* Types of Errors* Troubleshooting Tools* Using the Python Debugger	ACTIVITY: Snack Receipt Online, auto-graded



Days	CompuScholar Chapter	Activity Notes
6	Chapter 8: Making Decisions <ul style="list-style-type: none">* Logical Expressions* The "if" Statement* Using "elif" and "else"* The "match-case" Statement	ACTIVITY: Festival Stage Planner Online, auto-graded
5	Chapter 9: More Complex Logic <ul style="list-style-type: none">* Compound Expressions* Operator Precedence* Boolean Algebra and Truth Tables	ACTIVITY: Talent Show Coordinator Online, auto-graded
6	Chapter 10: Creating Functions <ul style="list-style-type: none">* Writing and Calling Functions* Function Parameters* Function Return Values* Local and Global Scope	ACTIVITY: School Spirit Online, auto-graded
6	Chapter 11: Loops <ul style="list-style-type: none">* "For" Loops* "While" Loops* Break, Continue, and Else* Nested Loops	ACTIVITY: Escape Room Online, auto-graded
5	Chapter 12: Handling Exceptions <ul style="list-style-type: none">* Understanding Exceptions* Catching Exceptions* Validating User Input	ACTIVITY: Ticket Booth Activity Online, auto-graded
6	Chapter 13: Lists <ul style="list-style-type: none">* Lists and Tuples* List Functions* List Traversal* Sequential and Binary Searches	ACTIVITY: Track Roster Online, auto-graded
4	Chapter 14: Mid-Term Project <ul style="list-style-type: none">* Introducing Bulls & Bears	ACTIVITY 1: Bulls & Bears – Part 1 ACTIVITY 2: Bulls & Bears – Part 2 ACTIVITY 3: Bulls & Bears – Analysis Online, auto-graded (Parts 1 and 2)
77	Approximate Days, Semester 1	



Semester 2 Timeline

Days	CompuScholar Chapter	Activity Notes
5	Chapter 15: Designing Algorithms <ul style="list-style-type: none">* Designing with Flowcharts* Writing Pseudocode* Common Algorithms	ACTIVITY: Quick Drop Delivery Online, auto-graded
5	Chapter 16: Sorting Algorithms <ul style="list-style-type: none">* Bubble Sort* Selection Sort* Insertion Sort	ACTIVITY: Sprint Meet Online, auto-graded
6	Chapter 17: Creating Classes <ul style="list-style-type: none">* OOP Concepts and History* Defining a Class and Creating Objects* Defining and Calling Methods* Returning Objects and Tuples	ACTIVITY: Animal Shelter Online, auto-graded
6	Chapter 18: Object Data <ul style="list-style-type: none">* Object Attributes* Constructors* Properties and Encapsulation* Variable Scope	ACTIVITY: Deep Space Observatory Online, auto-graded
5	Chapter 19: Static Concepts <ul style="list-style-type: none">* Class Attributes* Static Methods* Class Methods	ACTIVITY: Iron Shield Online, auto-graded
5	Chapter 20: File I/O <ul style="list-style-type: none">* Text File Reading and Writing* Managing Files with the OS Module* File Resource and Error Handling	ACTIVITY: City Desk Article Submission Online, auto-graded
5	Chapter 21: Dates and Times <ul style="list-style-type: none">* Dates and Times* Date and Time Calculations* Time Zones and Date Inputs	ACTIVITY: Vacation Planner Online, auto-graded
5	Chapter 22: Lists of Lists <ul style="list-style-type: none">* Matrix and Cube Structures* Traversal and Ordering* Advanced List Algorithms	ACTIVITY: Arcade Score Tracker Online, auto-graded



Teaching Tomorrow's Technology.

CompuScholar

Days	CompuScholar Chapter	Activity Notes
5	Chapter 23: Dictionaries and Sets <ul style="list-style-type: none">* Dictionaries* Dictionary Methods* Sets	ACTIVITY: Coffee Shop Orders Online, auto-graded
5	Chapter 24: Inheritance <ul style="list-style-type: none">* Superclass and Subclass Concepts* Subclass Constructors* Using Superclass and Subclass References	ACTIVITY: Fleet Manager Online, auto-graded
5	Chapter 25: Polymorphism <ul style="list-style-type: none">* Overriding Superclass Methods* The "object" Superclass* Using Superclass Features with "super()"	ACTIVITY: Stage Management System Online, auto-graded
5	Chapter 26: Recursion <ul style="list-style-type: none">* Introducing Recursion* Recursive Binary Search* Merge Sort	ACTIVITY: The Lost Vault Online, auto-graded
62	Approximate Days, Semester 2 (all "core" chapters complete at this point)	

Classes that complete the first 26 chapters at this point have spent approximately 139 days and finished "core" requirements typical of most states. The remaining class time should be spent on any teacher-selected topics from Chapters 27 - 29 or the Supplemental Chapters.

Please see the next page for information on the **optional chapters and Supplemental topics**.



The following table suggests the timeline needed for each **optional or supplemental chapter**, along with notes as to the programming environment and grading approach. There are more “optional” chapters available than students can complete in a single year, so teachers can pick topics as time permits!

Days	CompuScholar Chapter	Notes
10-15	Chapter 27: Team Project <ul style="list-style-type: none">* Design Processes and Teamwork* Requirements and Design Documents* Testing Your Code	ACTIVITY 1: Project Requirements and Design ACTIVITY 2: Project Implementation ACTIVITY 3: Project Testing CompuScholar online environment or external IDE, teacher-graded project
3	Chapter 28: Python On Your Computer <ul style="list-style-type: none">* Installing Python* Managing Project Directories* Visual Studio Code	“How-to” chapter / no activity
5	Chapter 29: Graphical Applications <ul style="list-style-type: none">* Drawing with Code* Events and Interaction* Simple Graphical Interfaces	ACTIVITY: Etch-A-Sketch Requires locally installed Python or an external IDE. Offline work, teacher-graded.
12	Supplemental Chapter 1: Enrichment Topics	See individual lessons and activities for the programming environment and grading approach.
8	Supplemental Chapter 2: Software and Industry	See individual lessons and activities for the programming environment and grading approach.
4	Supplemental Chapter 3: Computers and Modern Society	See individual lessons and activities for the programming environment and grading approach.
6	Supplemental Chapter 4: Computer Networking	See individual lessons and activities for the programming environment and grading approach.
4	Supplemental Chapter 5: PCSP Exam Prep	See individual lessons and activities for the programming environment and grading approach.