

# Python Programming

## Course Syllabus and Planner

### Course Overview

The CompuScholar **Python Programming** curriculum is a one-semester course covering topics typically found in introductory computer programming (coding) courses. This course has been aligned to specific course standards in many states. Please visit our course description page for a video tour and alignment information.

<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.

### Teaching Strategies

The course material is designed to appeal to a variety of students, from traditional learners who thrive on written text to audio-visual students who enjoy a multi-media format. All content is delivered through an online system that allows students to work seamlessly both in the classroom and at home.

### Labs and Programming Environment

The chosen programming language is Python 3.x (<https://www.python.org/>).

Each lesson contains hands-on coding opportunities (“Work with Me” sections). An online Python engine is made available within the lesson text and web browser interface, so students do not normally need to install any local Python software or manage local Python source files.

Every chapter contains one or more hands-on programming labs where students will design or implement programs to demonstrate understanding of the lesson topics. These labs use the same online Python engine as the lesson exercises and are automatically graded by our system.

## Supplemental Lessons and State Standards

The main chapters concentrate on Python programming and computer science topics. Certain states may require additional topics ranging from computing ethics and security to career exploration to the impact of computers on modern society. The course contains additional supplemental chapters at the end that can optionally be used to meet common state requirements. Teachers may choose to assign Supplemental Lessons as desired to meet state standards or student interest.

The final “Creative Project” is optional and can be completed in small groups or individually. Working in small groups is sometimes required by state standards. This project can be scaled up or down to meet available class times. The Creative Project is not auto-graded by our system, so teachers will be responsible for assigning these project grades.

## Course Planner

A typical school semester consists of approximately 18 calendar weeks or 90 days of school. The course plan covers approximately 65 to 90 school days, depending on which supplemental lessons and creative projects are incorporated into the class plan. Each “day” listed below represents one typical class period of 45 – 60 minutes, so students will typically work 3-5 hours per week. Some classes may move faster or slower than the suggested pace.

The planner assumes students are working daily with approximately 45 to 60 minutes of class time. In most cases, the planner estimates one day per lesson (including the lesson quiz), one day per graded chapter activity (lab), and one day per chapter test. Some larger labs are assigned more time.

Days	Reading and Objectives	Labs
5	<b>Chapter One: Fundamentals of Python</b> <ul style="list-style-type: none"> <li>• Introduction to Python</li> <li>• Running Python Programs</li> <li>• Writing Python Code</li> </ul>	<b>Class Schedule</b>

<b>Days</b>	<b>Reading and Objectives</b>	<b>Labs</b>
<b>5</b>	<b>Chapter Two: Working with Data</b> <ul style="list-style-type: none"> <li>• Data Types and Variables</li> <li>• Using Numeric Variables</li> <li>• Using String Variables</li> </ul>	<b>Cash Register</b>
<b>5</b>	<b>Chapter Three: Input and Output</b> <ul style="list-style-type: none"> <li>• Printing with Parameters</li> <li>• Getting Input from a User</li> <li>• String Formatting</li> </ul>	<b>Character Art</b>
<b>6</b>	<b>Chapter Four: Making Decisions</b> <ul style="list-style-type: none"> <li>• Logical Expressions</li> <li>• The “if” Statement</li> <li>• Logical Operators</li> <li>• More Complex Expressions</li> </ul>	<b>Blue Moon</b>
<b>5</b>	<b>Chapter Five: Finding and Fixing Problems</b> <ul style="list-style-type: none"> <li>• Types of Errors</li> <li>• Troubleshooting Tools</li> <li>• Using the Python Debugger</li> </ul>	<b>Chat-Bot</b>
<b>6</b>	<b>Chapter Six: Lists and Loops</b> <ul style="list-style-type: none"> <li>• Lists and Tuples</li> <li>• List Functions</li> <li>• “For” Loops</li> <li>• “While” Loops</li> </ul>	<b>Burger Castle</b>

Days	Reading and Objectives	Labs
6	<b>Chapter Seven: Numeric and Date Functions</b> <ul style="list-style-type: none"> <li>• Dates and Times</li> <li>• Advanced Data and Time Management</li> <li>• Random Numbers</li> <li>• The Math Library</li> </ul>	<b>Vacation Planner</b>
5	<b>Chapter Eight: Working with Strings</b> <ul style="list-style-type: none"> <li>• Character Data</li> <li>• String Functions</li> <li>• Input Validation with “try / except”</li> </ul>	<b>Pig Latin Translator</b>
5	<b>Chapter Nine: Functions</b> <ul style="list-style-type: none"> <li>• Writing and Calling Functions</li> <li>• Function Inputs and Outputs</li> <li>• Local and Global Scope</li> </ul>	<b>Verification Function</b>
5	<b>Chapter Ten: Python Classes</b> <ul style="list-style-type: none"> <li>• Thinking about Objects</li> <li>• Class Variables and Methods</li> <li>• Managing Class Files</li> </ul>	<b>ATM</b>
5	<b>Chapter Eleven: Class Instances</b> <ul style="list-style-type: none"> <li>• Creating Objects with Instance Data</li> <li>• Instance Methods</li> <li>• Managing Objects</li> </ul>	<b>Mars Rover</b>

<b>Days</b>	<b>Reading and Objectives</b>	<b>Labs</b>
<b>7</b>	<b>Chapter Twelve: Food Fight Project</b> <ul style="list-style-type: none"> <li>• Introducing “Food Fight”</li> </ul>	<b>Food and Defense Items</b>  <b>The Chef Class</b>  <b>Main Game Code</b>
<b>10</b>	<b>Chapter Thirteen: Creative Project</b>  (Scale as desired to meet available time) <ul style="list-style-type: none"> <li>• Project Life-cycles and teams</li> </ul>	<b>Requirements</b>  <b>Design</b>  <b>Coding</b>  <b>Testing</b>
<b>3</b>	<b>Supplemental Chapter One: Python on Your Computer</b> <ul style="list-style-type: none"> <li>• Installing Python</li> <li>• Managing Projects</li> <li>• Using the Python IDE</li> </ul>	N/A
<b>4</b>	<b>Supplemental Chapter Two: Computing Ethics and Security</b> <ul style="list-style-type: none"> <li>• Ethical Use of Computers</li> <li>• Intellectual Property</li> <li>• Digital Security</li> <li>• Social Engineering</li> </ul>	<b>Individual exercises per lesson</b>

<b>Days</b>	<b>Reading and Objectives</b>	<b>Labs</b>
<b>5</b>	<b>Supplemental Chapter Three: Enrichment Topics</b> <ul style="list-style-type: none"> <li>• Computer Number Systems</li> <li>• Encoding Data</li> <li>• Algorithms</li> <li>• Software Development Careers</li> <li>• Student Organizations</li> </ul>	<b>Individual exercises per lesson</b>
<b>4</b>	<b>Supplemental Chapter Four: Computers and Modern Society</b> <ul style="list-style-type: none"> <li>• Global Computing Issues</li> <li>• Managing Your Digital Identity</li> <li>• Impact of Computing</li> <li>• Artificial Intelligence</li> </ul>	<b>Individual exercises per lesson</b>
<b>3</b>	<b>Supplemental Chapter Five: File I/O</b> <ul style="list-style-type: none"> <li>• Text File Reading and Writing</li> <li>• Managing Files with the OS Module</li> <li>• File Resource and Error Handling</li> </ul>	<b>Individual exercises per lesson</b>