



Computer Science Foundations For Introductory Computer Science

Course Syllabus and Planner

Course Overview

CompuScholar's **Computer Science Foundations** curriculum is a one-year (two-semester) course that teaches introductory computer science concepts using the Python language.. It is aligned to numerous state and national standards for courses such as "Fundamentals of Computer Science", "Computer Science Principles", "Information Technology Foundations", 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/csfoundations/>

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

Computer Science Foundations can also be used as an **Advanced Placement (AP) "Computer Science Principles"** curriculum. For details, please see our Syllabus and Planner guide for AP CSP teachers.

Course Material

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 multimedia format. All content is delivered through an online system that allows students to work seamlessly both in the classroom and at home.



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
- **Homework Exercises** - coding and other challenges recommended (but not required) for additional practice
- **Chapter Activities** – hands-on projects, submitted for a grade

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 the Python coding exercises and activities. When using the online coding environment:

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

Later, optional chapters contain a mixture of activities. Teachers may select any of these topics for students as desired to meet specific state requirements. Some optional activities can be done in CompuScholar's online environment, while others are completed offline on a local computer.

Project Grading

Each chapter normally contains one or more hands-on, graded activities. The Python coding activities **are fully auto-graded by the CompuScholar system**. Teachers have complete control over the auto-graded results.

Some activities in other chapters, especially those not involving a programming project, result in creative student work or work that is completed offline. The teacher is responsible for grading those creative or external projects.



Course Navigation

Chapters 1 – 20 are considered the “core” curriculum that cover the required topics in most state introductory computer science courses. Students are generally advised to complete these chapters, in sequence. If a specific topic is not required by your state or appropriate for your classroom then teachers may choose to skip those chapters.

Chapters 21 – 22 contain topics specific to AP Computer Science Principles (CSP) students and may be skipped in non-AP settings.

Chapters 21 – 22 contain topics specific to AP Computer Science Principles (CSP) students and may be skipped in non-AP settings.

Chapters 23 – 25 contain an optional **Web Design** unit, which may be used in cases where students should learn basic HTML/CSS and web design skills.

Chapters 26 – 28 contain an optional **Computer Skills** unit, which may be used in cases where students should learn more about operating systems, computer files, and search engines.

Chapter 29 contains an optional **Computer Careers** unit, which may be used in cases where students should learn about career opportunities, workplace behavior, and safety.

Supplemental Chapters contain a variety of enrichment topics that may be required by individual states to satisfy requirements for other coding or digital literacy courses.

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. In some cases, state requirements can be met in a single-semester using an abbreviated set of chapters, lessons, and hands-on activities.



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 in chapters 1 - 20, most classes will have approximately 6-8 weeks left in the school year to explore optional units and supplemental topics. Teachers may direct students to any appropriate topics, time permitting.

Each "day" listed below represents one typical class period of 45 – 60 minutes. In most cases, 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.

Schools will normally award 1.0 credits for completion of this two-semester course.

Semester 1 Timeline

Days	CompuScholar Chapter and Lab	Notes
5	Chapter 1: Computing Concepts * Evolution of Computers * Computer Hardware * Computer Software LAB: Using Peripherals	Online, teacher-graded
6	Chapter 2: Networking * Network Hardware * How the Internet Works * Internet Scalability & Fault Tolerance * Parallel and Distributed Computing LAB: Network Analysis	Online, auto-graded
5	Chapter 3: Fundamentals of Python * Introduction to Python * Running Python Programs * Writing Python Code LAB: Class Schedule	Online, auto-graded



Days	CompuScholar Chapter and Lab	Notes
6	Chapter 4: Working with Data <ul style="list-style-type: none">* Introduction to Abstraction* Data Types and Variables* Using Numeric Variables* Using String Variables LAB: Cash Register	Online, auto-graded
5	Chapter 5: Input and Output <ul style="list-style-type: none">* Printing with Parameters* Getting Input from a User* String Formatting LAB: Character Art	Online, auto-graded
6	Chapter 6: Making Decisions <ul style="list-style-type: none">* Logical Expressions* The "if" Statement* Logical Operators* More Complex Expressions LAB: Blue Moon	Online, auto-graded
5	Chapter 7: Finding and Fixing Problems <ul style="list-style-type: none">* Types of Errors* Troubleshooting Tools* Using the Python Debugger LAB: Chat-Bot	Online, auto-graded
5	Chapter 8: Loops <ul style="list-style-type: none">* "For" Loops* "While" Loops* Break, Continue, and Else LAB: Vowel Eraser	Online, auto-graded
6	Chapter 9: Lists <ul style="list-style-type: none">* Lists and Tuples* List Functions* List Traversal* Sequential and Binary Searches LAB: Burger Castle	Online, auto-graded



Days	CompuScholar Chapter and Lab	Notes
6	Chapter 10: Math Concepts * Libraries and Documentation * Random Numbers * The Math Library * Computer Number Systems LAB: Math Contest	Online, auto-graded
5	Chapter 11: Working with Strings * Character Data * String Functions * Input Validation (try/except) LAB: Pig Latin Translator	Online, auto-graded
6	Chapter 12: Creating Functions * Writing and Calling Functions * Function Inputs and Outputs * Local and Global Scope * Procedural Abstraction LAB: Verification Function	Online, auto-graded
7	Chapter 13: Mid-Term Project * Introducing "GOPS" LAB 1: "GOPS" Part 1 LAB 2: "GOPS" Part 2 LAB 3: "GOPS" Analysis	Online, auto-graded (Part 1 and 2) Online, teacher-graded (Part 3)
73	Approximate Days, Semester 1, "core" chapters	

Semester 2 Timeline

Days	CompuScholar Chapter and Lab	Notes
7	Chapter 14: Collaborative Design and Development * Collaboration * Design Processes and Teamwork * Requirements and Design Documents * Testing Your Code LAB 1: Project Requirements and Design LAB 2: Project Implementation	Online, teacher-graded



	LAB 3: Project Testing	
5	Chapter 15: Designing Algorithms * Designing with Flowcharts * Writing Pseudocode * Common Math Algorithms LAB: Cat and Mouse	Online, auto-graded
5	Chapter 16: Real-World Algorithms * Algorithm Efficiency * Undecidable Problems * Simulations LAB: Knapsack Problem	Online, auto-graded
6	Chapter 17: Understanding Data * Extracting Info from Data * Challenges with Processing Data * Using Data in Programs * Data Compression LAB: Census Analysis	Online, teacher-graded
5	Chapter 18: Impact of Computing * Digital Divide and Bias * Effects of Computing Innovations * Crowdsourcing and Citizen Science LAB: Impact Study	Online, teacher-graded
5	Chapter 19: Legal and Ethical Concerns * Computing Ethics * Computing Laws * Intellectual Property LAB: Ethics Illustration	Online, teacher-graded
5	Chapter 20: Cybersecurity * Protect Yourself Online * Computer Security * Social Engineering LAB: Stuxnet Worm	Online, auto-graded
38	Approximate Days, Semester 2, "core" chapters	

Classes who complete the first 20 chapters at this point have spent approximately 110 days on standard "core" content. **The remaining class time should be spent in exploration of any desired optional units and supplemental (enrichment) topics (see below).**



The following table suggests the timeline needed for each **optional** or **supplemental chapter**, along with notes as to the programming environment and grading approach.

Optional WEB DESIGN Unit

Days	CompuScholar Chapter and Lab	Notes
5	Chapter 23: Creating Web Pages * Getting Started with HTML * Creating HTML Files * HTML File Layout LAB: Beginning Web Page	WEB DESIGN UNIT Offline, teacher-graded
5	Chapter 24: Web Page Design * Body Elements * Using Colors * Design Rules LAB: Formatted Web Page	WEB DESIGN UNIT Offline, teacher-graded
6	Chapter 25: Links, Images and Animation * Adding Hyperlinks * Using Images * Adding Animation * Simple JavaScript LAB: Final Website	WEB DESIGN UNIT Offline, teacher-graded
16	Days in the Web Design Unit	



Optional COMPUTER SKILLS Unit

Days	CompuScholar Chapter and Lab	Notes
5	Chapter 26: Operating Systems * Popular Operating Systems * Managing Your OS * Managing Your Applications LAB: OS Report	COMPUTER SKILLS UNIT Online, teacher-graded
5	Chapter 27: Computer Files * Understanding Files and Folders * Managing Files on Your Computer * File Associations LAB: Mystery Files	COMPUTER SKILLS UNIT Online, teacher-graded
5	Chapter 28: Search Engines * Using Search Engines * Search Results * Verifying and Citing Sources LAB: Search Report	COMPUTER SKILLS UNIT Online, teacher-graded
15	Days in the Computer Skills Unit	

Optional COMPUTER CAREERS Unit

Days	CompuScholar Chapter and Lab	Notes
6	Chapter 29: Computer Careers * Computer Career Opportunities * Professionalism in the Workplace * Workplace Safety * Student Organizations LAB: Exploring Computing Careers	COMPUTER CAREERS UNIT Offline, teacher-graded
6	Days in the Computer Careers Unit	



Optional Supplemental Chapters and Topics

Days	CompuScholar Chapter and Lab	Notes
3	Supplemental Chapter 1: Python on Your Computer <ul style="list-style-type: none">* Installing Python* Managing Project Directories* Python IDLE Development Environment	Offline
3	Supplemental Chapter 2: File I/O with Python <ul style="list-style-type: none">* Text File Reading and Writing* Managing Files with the OS Module* File Resource and Error Handling	Online
6	Days in the Supplemental Chapters	