



C# Programming

For Introductory Computer Science

Course Syllabus and Planner

Course Overview

CompuScholar's **C# Programming** curriculum is a one-year (two-semester) course that teaches students to code in the C# language. It is aligned to numerous state and national standards for courses such as "Computer Programming I" or similar titles.

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

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 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 C# 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).**

Two optional chapters at the end of the course introduce Visual Studio as an external IDE. Teachers may use those optional chapters, if desired, to cover graphical projects. Teachers may also ask students to use Visual Studio for other course coding exercises instead of using our online IDE. Visual Studio requires a **Windows** or **Mac OS computer** for local installation.

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 the external Visual Studio IDE. The teacher is responsible for grading those optional projects.

Course Navigation

Chapters 1 - 23 are considered “core” curriculum that covers the required topics in most state 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 (e.g., recursion or sorting), then teachers may choose to skip those chapters.

Chapter 24 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 25 - 26 introduce Visual Studio as an **optional**, external IDE.



Supplemental Chapters 1 – 4 contain a variety of 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.

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* Computer Hardware* Computer Software* Computer Ethics* Computer Security	Common curricular requirements
6	Chapter 2: Fundamentals of C# <ul style="list-style-type: none">* Common Programming Languages* Introduction to C#* Writing Your First Program* Help and Reference Documentation	ACTIVITY: School Supplies Online, auto-graded



5	Chapter 3: Data Types and Variables <ul style="list-style-type: none">* Value Data Types* Variables* Printing Data	ACTIVITY: Pizza Bot Online, auto-graded
5	Chapter 4: Working with Numbers <ul style="list-style-type: none">* Simple Math Operations* Compound Assignments and Shortcuts* Type Casting and Truncation	ACTIVITY: Math Professor Online, auto-graded
6	Chapter 5: Introducing Strings <ul style="list-style-type: none">* Introducing Strings* Getting Console Input* Common String Operations* Formatting Strings	ACTIVITY: Language Jumbler Online, auto-graded
5	Chapter 6: Numbering Systems and Math Functions <ul style="list-style-type: none">* Converting Between Strings and Numbers* .NET Math and Random Classes* Binary and Hexadecimal Numbers	ACTIVITY: Math Machine Online, auto-graded
6	Chapter 7: Logic and Decision-Making <ul style="list-style-type: none">* Logical Expressions* Making Decisions with if()* Using "else-if()" and "else"* The "switch()" Statement	ACTIVITY: Coffee Shop Online, auto-graded
5	Chapter 8: More Complex Logic <ul style="list-style-type: none">* Compound Expressions* Operator Precedence* Boolean Algebra and Truth Tables	ACTIVITY: Dog Trainer Online, auto-graded
5	Chapter 9: Handling Exceptions <ul style="list-style-type: none">* Understanding Exceptions* Catching Exceptions* Validating User Input	ACTIVITY: String Slicer Online, auto-graded
3	Chapter 10: Debugging <ul style="list-style-type: none">* Finding Run-time Errors* Debugger Concepts	ACTIVITY: Menu Madness Online, auto-graded



6	Chapter 11: Iteration <ul style="list-style-type: none">* For() Loops* While() Loops* Break and Continue* Nested Loops	ACTIVITY: Population Growth Online, auto-graded
5	Chapter 12: Algorithms <ul style="list-style-type: none">* Designing with Flowcharts* Writing Pseudocode* Common Algorithms	ACTIVITY: Delivery Driver Online, auto-graded
7	Chapter 13: Creating Classes <ul style="list-style-type: none">* Object-Oriented Concepts and History* Defining a Class* Defining and Calling Methods* Method Parameters and Return Data* Method Overloading	ACTIVITY: Duck Trainer Online, auto-graded
6	Chapter 14: Class Data <ul style="list-style-type: none">* Class Fields* Constructors* Class Properties and Encapsulation* Variable Scope	ACTIVITY: Video Player Online, auto-graded
4	Chapter 15: Static Concepts <ul style="list-style-type: none">* Static Fields* Static Methods	ACTIVITY: Gas Station Online, auto-graded
4	Chapter 16: Mid-Term Project <ul style="list-style-type: none">* Introducing the "Game Machine " Project	ACTIVITY 1: Guess Number Game ACTIVITY 2: Word Challenge Game ACTIVITY 3: Game Machine Online, auto-graded
84	Approximate Days, Semester 1	

Semester 2 Timeline

Days	CompuScholar Chapter	Activity Notes
6	Chapter 17: 1D Arrays <ul style="list-style-type: none">* Array Concepts* Array Traversals* The foreach() Loop* Array Algorithms	ACTIVITY: 1D Bingo Online, auto-graded



6	Chapter 18: Lists in C# <ul style="list-style-type: none">* Linked Lists* The List Class* Traversing Collections* List Algorithms	ACTIVITY: Card Dealer Online, auto-graded
6	Chapter 19: Searching and Sorting <ul style="list-style-type: none">* Bubble Sort* Selection Sort* Insertion Sort* Sequential and Binary Searches	ACTIVITY: Weird Echo Online, auto-graded
5	Chapter 20: 2D Arrays <ul style="list-style-type: none">* 2D Arrays* Traversal and Ordering* 2D Array Algorithms	ACTIVITY: Super Tic-Tac-Toe Online, auto-graded
5	Chapter 21: Inheritance <ul style="list-style-type: none">* Superclass and Subclass Concepts* Subclass Constructors* Using Superclass and Subclass References	ACTIVITY: Modern Art Online, auto-graded
5	Chapter 22: Polymorphism <ul style="list-style-type: none">* Overriding Superclass Methods* The Object Base Class* Using Superclass Features with "base"	ACTIVITY: Chess Challenge Online, auto-graded
5	Chapter 23: Recursion <ul style="list-style-type: none">* Recursion* Recursive Binary Search* Merge Sort	ACTIVITY: Fibonacci Numbers Online, auto-graded
37	Approximate Days, Semester 2 (all "core" chapters complete at this point)	

Classes who complete the first 23 chapters at this point have spent approximately 122 days and finished "core" requirements typical of most states. The remaining class time should be spent on any teacher-selected topics from Chapters 24 – 26 or the Supplemental Chapters.

Please see below 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 24: Team Project <ul style="list-style-type: none">* Design Processes and Teamwork* Requirements and Design Documents* Testing Your Code	ACTIVITY 1: Team Project Requirements ACTIVITY 2: Project Design ACTIVITY 3: Team Project Implementation ACTIVITY 4: Team Project Testing CompuScholar online environment or external IDE, teacher-graded project
3	Chapter 25: Introducing Visual Studio <ul style="list-style-type: none">* Installing Visual Studio* Local Source Files* Creating Console Projects	“How-to” chapter to enable the use of Visual Studio (requires Windows computer)
5	Chapter 26: Graphical Desktop Apps <ul style="list-style-type: none">* Creating Graphical Projects* Event-Driven Programming* Input Controls	ACTIVITY: Tall Tales Requires locally installed Visual Studio and a Windows Computer. 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	Offline work, teacher-graded projects
4	Supplemental Chapter 3: Computers and Modern Society	Offline work, teacher-graded projects
6	Supplemental Chapter 4: Computer Networking	Offline work, teacher-graded projects