# Scope and Sequencing - Text

## Standards Alignment

# Table of Contents

| Course Name | Lesson/Activity Count | Grade | Access | Difficulty | K-2 | Elementary School | Middle School | High School | Page |
|---|---|---|---|---|---|---|---|---|---|
| **Intro to Programming and Art** | 59 Lessons 304 Activities | 9+ | 🌐 | Advanced | | | | ✔ | **61** |
| **AP Computer Science Principles** | 110 Lessons 499 Activities | 9+ | 🌐 | Advanced | | | | ✔ | **73** |
| **AP Computer Science A** | 123 Lessons 588 Activities | 9+ | 🌐 | Advanced | | | | ✔ | **88** |

# JavaScript 101

## Grades 7+

### Scope and Sequence

Each lesson takes about 45-60 minutes to complete.

JavaScript 101 is a course for students in grade 7 or above who are already familiar with the basics of programming. The stories, games, puzzles, and projects engage students in developing computational thinking skills in JavaScript, as listed below from the CSTA Level 3 Computer Science standards and UK Computing standards. The Common Core State Standards for Mathematics and English Language Arts that students develop are also listed here.

| | Lesson 1 - The Basics | Lesson 2 - Loops and Patterns | Lesson 3 - Conditional Logic | Lesson 4 - Conditional Loops | Lesson 5 - Variables | Lesson 6 - Expressions | Lesson 7 - Using the Canvas | Lesson 8 - User Interaction |
|---|---|---|---|---|---|---|---|---|
| **Key Skills and Concepts** | • Create custom sequences to solve puzzles<br>• Use function calls<br>• Use JavaScript syntax, naming conventions, and comments | • Use simple, nested, and "for" loops<br>• Use arithmetic operators<br>• Recognize patterns | • Distinguish between assignment, comparison, and logical operators<br>• Use conditional logic to program algorithms | • Use "while" and "do-while" loops<br>• Use conditional loops to solve puzzles<br>• Distinguish between "for" and "while" loops. | • Use variable declarations and assignments<br>• Use and define strings<br>• Use variables to solve puzzles | • Write and understand JavaScript expressions<br>• Use operator precedence to evaluate an expression | • Use canvas elements, sprites, and layering<br>• Create game using HTML and JavaScript<br>• Create timed loops that execute code with a delay | • Set up and handle keyboard and mouse events<br>• Find the correct key code to handle a specific keyboard input |
| **CCSS-Math Standards** | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3<br>MP.1 | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3<br>MP.1<br>MP.2 | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3<br>MP.1 | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3<br>MP.1 | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3<br>MP.1 | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3<br>HSA.CED.1<br>MP.1 | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3<br>HSA.CED.A.1<br>HSA.CED.A.3<br>MP.1, MP.2, MP.4 | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3<br>HSA.CED.A.1<br>HSA.CED.A.3<br>MP.1, MP.2, MP.4 |
| **CCSS-ELA Standards** | SL.7.1<br>SL.8.1<br>RI.9-10.3<br>RI.9-10.6<br>L.9-10.3<br>L.9-10.6 | SL.7.1<br>SL.8.1<br>RI.9-10.3<br>RI.9-10.6<br>L.9-10.3<br>L.9-10.6 | SL.7.1<br>SL.8.1<br>RI.9-10.3<br>RI.9-10.6<br>L.9-10.3<br>L.9-10.6 | SL.7.1<br>SL.8.1<br>RI.9-10.3<br>RI.9-10.6<br>L.9-10.3<br>L.9-10.6 | SL.7.1<br>SL.8.1<br>RI.9-10.3<br>RI.9-10.6<br>L.9-10.3<br>L.9-10.6 | SL.7.1<br>SL.8.1<br>RI.9-10.3<br>RI.9-10.6<br>L.9-10.3<br>L.9-10.6 | SL.7.1<br>SL.8.1<br>RI.9-10.3<br>RI.9-10.6<br>L.9-10.3<br>L.9-10.6 | SL.7.1<br>SL.8.1<br>RI.9-10.3<br>RI.9-10.6<br>L.9-10.3<br>L.9-10.6 |
| **CSTA Computer Science Standards** | 2-AP-10<br>2-AP-13<br>2-AP-17<br>3A-AP-17<br>3A-AP-19<br>3B-AP-11<br>3B-AP-12 | 2-AP-11<br>2-AP-13<br>2-AP-15<br>2-AP-17<br>3A-AP-17<br>3A-AP-19<br>3B-AP-11 | 2-AP-11<br>2-AP-12<br>2-AP-13<br>2-AP-15<br>2-AP-17<br>3A-AP-17<br>3A-AP-19 | 2-AP-11<br>2-AP-13<br>2-AP-15<br>2-AP-17<br>3A-AP-17<br>3A-AP-19<br>3B-AP-11 | 2-AP-11<br>2-AP-13<br>2-AP-15<br>2-AP-17<br>3A-AP-17<br>3A-AP-19<br>3B-AP-11 | 2-AP-11<br>2-AP-13<br>2-AP-15<br>2-AP-17<br>3A-AP-17<br>3A-AP-19<br>3B-AP-11 | 2-AP-11<br>2-AP-13<br>2-AP-16<br>2-AP-17<br>3A-AP-17<br>3B-AP-11<br>3B-AP-12 | 2-AP-11<br>2-AP-13<br>2-AP-16<br>2-AP-17<br>3A-AP-17<br>3B-AP-11<br>3B-AP-12 |

| | | 3B-AP-12 | 3B-AP-11 3B-AP-12 | 3B-AP-12 | 3B-AP-12 | 3B-AP-12 | 3B-AP-22 | 3B-AP-22 |
|---|---|---|---|---|---|---|---|---|
| **ISTE Standards** | 1.c, 1.d, 4.d, 5.c, 5.d | 1.c, 1.d, 4.d, 5.c, 5.d | 1.c, 1.d, 4.d, 5.c, 5.d | 1.c, 1.d, 4.d, 5.c, 5.d | 1.c, 1.d, 4.d, 5.c, 5.d | 1.c, 1.d, 4.d, 5.c, 5.d | 1.c, 1.d, 4.d, 5.c, 5.d | 1.c, 1.d, 4.d, 5.c, 5.d |
| **UK National Curriculum** | Keystages 3 & 4 Computing* | Keystages 3 & 4 Computing* | Keystages 3 & 4 Computing* | Keystages 3 & 4 Computing* | Keystages 3 & 4 Computing* | Keystages 3 & 4 Computing* | Keystages 3 & 4 Computing* | Keystages 3 & 4 Computing* |
| **Sample Application of Skills** | Use function calls, naming conventions, and syntax to solve puzzles. | Use "for" loops, arithmetic operators, and sequencing to solve puzzles. | Use sequencing, operators, and conditional logic to solve puzzles. | Use conditional loops and pattern recognition to solve puzzles. | Use variables to store and manipulate information to solve puzzles. | Use expressions to solve puzzles. | Create a slideshow using images and delays | Create a scene that responds to keyboard and mouse inputs. |

*See individual lesson guides for details on UK Computer standards

# JavaScript 101

**Scope and Sequence**

**Grades 7+**

Each lesson takes about 45-60 minutes to complete.

JavaScript 101 is a course for students in grade 7 or above who are already familiar with the basics of programming. The stories, games, puzzles, and projects engage students in developing computational thinking skills in JavaScript, as listed below from the CSTA Level 3 Computer Science standards and UK Computing standards. The Common Core State Standards for Mathematics and English Language Arts that students develop are also listed here.

| | Lesson 9 - Game Design | Lesson 10 - Snake | Lesson 11 - Breakout | Lesson 12 - Pong | Lesson 13 - Final Game | | | |
|---|---|---|---|---|---|---|---|---|
| **Key Skills and Concepts** | • Use game loops, win/loss conditions, and keeping score functionality<br>• Use collision detection | • Use HTML and JavaScript syntax and functions<br>• Design and create a game on the canvas | • Use HTML and JavaScript syntax and functions<br>• Design and create a game on the canvas | • Use HTML and JavaScript syntax and functions<br>• Design and create a game on the canvas | • Use HTML and JavaScript syntax and functions<br>• Design and create a game on the canvas | | | |
| **CCSS-Math Standards** | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3<br>HSA.CED.A.1<br>HSA.CED.A.3<br>MP.1, MP.2, MP.4 | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3<br>HSA.CED.A.1<br>HSA.CED.A.3<br>MP.1, MP.2, MP.4 | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3<br>HSA.CED.A.1<br>HSA.CED.A.3<br>MP.1, MP.2, MP.4 | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3<br>HSA.CED.A.1<br>HSA.CED.A.3<br>MP.1, MP.2, MP.4 | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3<br>HSA.CED.A.1<br>HSA.CED.A.3<br>MP.1, MP.2, MP.4 | | | |
| **CCSS-ELA Standards** | SL.7.1<br>SL.8.1<br>RI.9-10.3<br>RI.9-10.6<br>L.9-10.3<br>L.9-10.6 | SL.7.1<br>SL.8.1<br>RI.9-10.3<br>RI.9-10.6<br>L.9-10.3<br>L.9-10.6 | SL.7.1<br>SL.8.1<br>RI.9-10.3<br>RI.9-10.6<br>L.9-10.3<br>L.9-10.6 | SL.7.1<br>SL.8.1<br>RI.9-10.3<br>RI.9-10.6<br>L.9-10.3<br>L.9-10.6 | SL.7.1<br>SL.8.1<br>RI.9-10.3<br>RI.9-10.6<br>L.9-10.3<br>L.9-10.6 | | | |
| **CSTA Computer Science Standards** | 2-AP-11<br>2-AP-12<br>2-AP-13<br>2-AP-16<br>2-AP-17<br>2-AP-19<br>3A-AP-13<br>3A-AP-17<br>3A-AP-19<br>3B-AP-11<br>3B-AP-12<br>3B-AP-22 | 2-AP-11<br>2-AP-12<br>2-AP-13<br>2-AP-16<br>2-AP-17<br>2-AP-19<br>3A-AP-13<br>3A-AP-17<br>3A-AP-19<br>3B-AP-09<br>3B-AP-11<br>3B-AP-12 | 2-AP-11<br>2-AP-12<br>2-AP-13<br>2-AP-16<br>2-AP-17<br>2-AP-19<br>3A-AP-13<br>3A-AP-17<br>3A-AP-19<br>3B-AP-11<br>3B-AP-12<br>3B-AP-22 | 2-AP-11<br>2-AP-12<br>2-AP-13<br>2-AP-16<br>2-AP-17<br>2-AP-19<br>3A-AP-13<br>3A-AP-17<br>3A-AP-19<br>3B-AP-11<br>3B-AP-12<br>3B-AP-22 | 2-AP-11<br>2-AP-12<br>2-AP-13<br>2-AP-16<br>2-AP-17<br>2-AP-19<br>3A-AP-13<br>3A-AP-17<br>3A-AP-19<br>3B-AP-11<br>3B-AP-12<br>3B-AP-22 | | | |

| | | 3B-AP-22 | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **ISTE Standards** | 1.c, 1.d, 4.d, 5.c, 5.d, 6.b | 1.c, 1.d, 4.d, 5.c, 5.d, 6.b | 1.c, 1.d, 4.d, 5.c, 5.d, 6.b | 1.c, 1.d, 4.d, 5.c, 5.d, 6.b | 1.c, 1.d, 4.d, 5.c, 5.d, 6.b | | | |
| **UK National Curriculum** | Keystages 3 & 4 Computing* | Keystages 3 & 4 Computing* | Keystages 3 & 4 Computing* | Keystages 3 & 4 Computing* | Keystages 3 & 4 Computing* | | | |
| **Sample Application of Skills** | Program sprites to represent the player and the opponent. | Create a "Snake" game. | Create a "Breakout" game. | Create a "Pong" game. | Create and customize three different games – "Flappy Bird", "Alien Invaders", and "Geometry Dash". | | | |

*See individual lesson guides for details on UK Computer standards

# Python 101

## Scope and Sequence

### Grades 7+

Each lesson takes about 45-60 minutes to complete.

Python 101 is a course for students in grade 7 or above who are already familiar with the basics of programming. The stories, games, puzzles, and projects engage students in developing computational thinking skills in Python, as listed below from the CSTA Level 3 Computer Science standards and UK Computing standards. The Common Core State Standards for Mathematics and English Language Arts that students develop are also listed here.

| | Lesson 1 - The Basics | Lesson 2 - Loops and Patterns | Lesson 3 - Conditional Logic | Lesson 4 - Conditional Loops | Lesson 5 - Variables | Lesson 6 - Expressions | Lesson 7 - Turtle Tool | Lesson 8 - User Interaction |
|---|---|---|---|---|---|---|---|---|
| **Key Skills and Concepts** | • Create custom sequences to solve puzzles<br>• Use function calls<br>• Use Python syntax, naming conventions, and comments | • Use simple, nested, and "for" loops<br>• Use arithmetic operators<br>• Recognize patterns | • Distinguish between assignment, comparison, and logical operators<br>• Use conditional logic to program algorithms | • Use "while" and "do-while" loops<br>• Use conditional loops to solve puzzles<br>• Distinguishing between "for" and "while" loops | • Use variable declarations and assignments<br>• Use and define strings<br>• Use variables to solve puzzles | • Write and understand Python expressions<br>• Use operator precedence to evaluate an expression | • Use turtle graphics<br>• Create animations using turtle graphics and Python | • Set up and handle keyboard and mouse events<br>• Find the correct key code to handle a specific keyboard input |
| **CCSS-Math Standards** | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3<br>MP.1 | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3<br>MP.1<br>MP.2 | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3<br>MP.1 | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3<br>MP.1 | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3<br>MP.1 | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3<br>HSA.CED.A.1<br>HSA.CED.A.3<br>MP.1, MP.2 | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3<br>HSA.CED.A.1<br>HSA.CED.A.3<br>MP.1, MP.2, MP.4 | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3<br>HSA.CED.A.1<br>HSA.CED.A.3<br>MP.1, MP.2, MP.4 |
| **CCSS-ELA Standards** | SL.7.1<br>SL.8.1<br>RI.9-10.3<br>RI.9-10.6<br>L.9-10.3<br>L.9-10.6 | SL.7.1<br>SL.8.1<br>RI.9-10.3<br>RI.9-10.6<br>L.9-10.3<br>L.9-10.6 | SL.7.1<br>SL.8.1<br>RI.9-10.3<br>RI.9-10.6<br>L.9-10.3<br>L.9-10.6 | SL.7.1<br>SL.8.1<br>RI.9-10.3<br>RI.9-10.6<br>L.9-10.3<br>L.9-10.6 | SL.7.1<br>SL.8.1<br>RI.9-10.3<br>RI.9-10.6<br>L.9-10.3<br>L.9-10.6 | SL.7.1<br>SL.8.1<br>RI.9-10.3<br>RI.9-10.6<br>L.9-10.3<br>L.9-10.6 | SL.7.1<br>SL.8.1<br>RI.9-10.3<br>RI.9-10.6<br>L.9-10.3<br>L.9-10.6 | SL.7.1<br>SL.8.1<br>RI.9-10.3<br>RI.9-10.6<br>L.9-10.3<br>L.9-10.6 |
| **CSTA Computer Science Standards** | 2-AP-13<br>2-AP-17<br>3A-AP-17<br>3A-AP-19<br>3B-AP-11<br>3B-AP-12 | 2-AP-11<br>2-AP-13<br>2-AP-15<br>2-AP-17<br>3A-AP-17<br>3A-AP-19 | 2-AP-11<br>2-AP-12<br>2-AP-13<br>2-AP-15<br>2-AP-17<br>3A-AP-17 | 2-AP-11<br>2-AP-13<br>2-AP-15<br>2-AP-17<br>3A-AP-17<br>3A-AP-19 | 2-AP-11<br>2-AP-13<br>2-AP-15<br>2-AP-17<br>3A-AP-17<br>3A-AP-19 | 2-AP-11<br>2-AP-13<br>2-AP-15<br>2-AP-17<br>3A-AP-17<br>3A-AP-19 | 2-AP-11<br>2-AP-13<br>2-AP-16<br>2-AP-17<br>3A-AP-17<br>3B-AP-11 | 2-AP-11<br>2-AP-13<br>2-AP-16<br>2-AP-17<br>3A-AP-17<br>3B-AP-11 |

|  | 3B-AP-11 3B-AP-12 | 3A-AP-19 3B-AP-11 3B-AP-12 | 3B-AP-11 3B-AP-12 | 3B-AP-11 3B-AP-12 | 3B-AP-11 3B-AP-12 | 3B-AP-12 3B-AP-22 | 3B-AP-12 3B-AP-22 |
|---|---|---|---|---|---|---|---|
| **ISTE Standards** | 1.c, 1.d, 4.d, 5.c, 5.d | 1.c, 1.d, 4.d, 5.c, 5.d | 1.c, 1.d, 4.d, 5.c, 5.d | 1.c, 1.d, 4.d, 5.c, 5.d | 1.c, 1.d, 4.d, 5.c, 5.d | 1.c, 1.d, 4.d, 5.c, 5.d | 1.c, 1.d, 4.d, 5.c, 5.d | 1.c, 1.d, 4.d, 5.c, 5.d |
| **UK National Curriculum** | Keystages 3 & 4 Computing* | Keystages 3 & 4 Computing* | Keystages 3 & 4 Computing* | Keystages 3 & 4 Computing* | Keystages 3 & 4 Computing* | Keystages 3 & 4 Computing* | Keystages 3 & 4 Computing* | Keystages 3 & 4 Computing* |
| **Sample Application of Skills** | Use function calls, naming conventions, and syntax to solve puzzles. | Use "for" loops, arithmetic operators, and sequencing to solve puzzles. | Use sequencing, operators, and conditional logic to solve puzzles. | Use conditional loops and pattern recognition to solve puzzles. | Use variables to store and manipulate information to solve puzzles. | Use expressions to solve puzzles. | Create an animation using turtle graphics and Python. | Create a scene that responds to keyboard and mouse inputs. |

*See individual lesson guides for details on UK Computer standards

# Python 101

## Scope and Sequence

**Grades 7+**

Each lesson takes about 45-60 minutes to complete.

Python 101 is a course for students in grade 7 or above who are already familiar with the basics of programming. The stories, games, puzzles, and projects engage students in developing computational thinking skills in Python, as listed below from the CSTA Level 3 Computer Science standards and UK Computing standards. The Common Core State Standards for Mathematics and English Language Arts that students develop are also listed here.

| | Lesson 9 - Game Design | Lesson 10 - Snake | Lesson 11 - Connect 4 | Lesson 12 - Tetris | Lesson 13 - Final Game | | | |
|---|---|---|---|---|---|---|---|---|
| **Key Skills and Concepts** | • Use game loops, win/loss conditions, and keeping score functionality<br>• Use collision detection | • Use turtle graphics, Python syntax, and functions<br>• Design and implement a game | • Use turtle graphics, Python syntax, and functions<br>• Design and implement a game | • Use turtle graphics, Python syntax, and functions<br>• Design and implement a game | • Use turtle graphics, Python syntax, and functions<br>• Design and implement a game | | | |
| **CCSS-Math Standards** | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3<br>HSA.CED.1<br>HSA.CED.2<br>HSA.CED.3<br>MP.1<br>MP.2<br>MP.4 | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3<br>HSA.CED.1<br>HSA.CED.2<br>HSA.CED.3<br>MP.1<br>MP.2<br>MP.4 | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3<br>HSA.CED.1<br>HSA.CED.2<br>HSA.CED.3<br>MP.1<br>MP.2<br>MP.4 | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3<br>HSA.CED.1<br>HSA.CED.2<br>HSA.CED.3<br>MP.1<br>MP.2<br>MP.4 | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3<br>HSA.CED.1<br>HSA.CED.2<br>HSA.CED.3<br>MP.1<br>MP.2<br>MP.4 | | | |
| **CCSS-ELA Standards** | SL.7.1<br>SL.8.1<br>RI.9-10.3<br>RI.9-10.6<br>L.9-10.3<br>L.9-10.6 | SL.7.1<br>SL.8.1<br>RI.9-10.3<br>RI.9-10.6<br>L.9-10.3<br>L.9-10.6 | SL.7.1<br>SL.8.1<br>RI.9-10.3<br>RI.9-10.6<br>L.9-10.3<br>L.9-10.6 | SL.7.1<br>SL.8.1<br>RI.9-10.3<br>RI.9-10.6<br>L.9-10.3<br>L.9-10.6 | SL.7.1<br>SL.8.1<br>RI.9-10.3<br>RI.9-10.6<br>L.9-10.3<br>L.9-10.6 | | | |
| **CSTA Computer Science Standards** | 2-AP-11<br>2-AP-12<br>2-AP-13<br>2-AP-15<br>2-AP-16<br>2-AP-17<br>3A-AP-17 | 2-AP-11<br>2-AP-12<br>2-AP-13<br>2-AP-16<br>2-AP-17<br>2-AP-19<br>3A-AP-17 | 2-AP-11<br>2-AP-12<br>2-AP-13<br>2-AP-16<br>2-AP-17<br>2-AP-19<br>3A-AP-17 | 2-AP-11<br>2-AP-12<br>2-AP-13<br>2-AP-16<br>2-AP-17<br>2-AP-19<br>3A-AP-17 | 2-AP-11<br>2-AP-12<br>2-AP-13<br>2-AP-16<br>2-AP-17<br>2-AP-19<br>3A-AP-17 | | | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | 3B-AP-11<br>3B-AP-12<br>3B-AP-22 | 3B-AP-09<br>3B-AP-10<br>3B-AP-11<br>3B-AP-12<br>3B-AP-22 | 3B-AP-11<br>3B-AP-12<br>3B-AP-22 | 3B-AP-09<br>3B-AP-10<br>3B-AP-11<br>3B-AP-12<br>3B-AP-22 | 3B-AP-11<br>3B-AP-12<br>3B-AP-21<br>3B-AP-22 | | |
| **ISTE Standards** | 1.c, 1.d, 4.d, 5.c, 5.d, 6.b | 1.c, 1.d, 4.d, 5.c, 5.d, 6.b | 1.c, 1.d, 4.d, 5.c, 5.d, 6.b | 1.c, 1.d, 4.d, 5.c, 5.d, 6.b | 1.c, 1.d, 4.d, 5.c, 5.d, 6.b | | |
| **UK National Curriculum** | Keystages 3 & 4 Computing* | Keystages 3 & 4 Computing* | Keystages 3 & 4 Computing* | Keystages 3 & 4 Computing* | Keystages 3 & 4 Computing* | | |
| **Sample Application of Skills** | Program sprites to represent the player and the opponent. | Create a "Snake" game. | Create a "Connect 4" game. | Create a "Tetris" game. | Create and customize two games – "Frogga" and "Pong." | | |

*See individual lesson guides for details on UK Computer standards

# MicroPython 101
## Scope and Sequence

**Grades 6+**

Each lesson takes about 45-60 minutes to complete.

The MicroPython 101 course introduces students to physical computing using micro:bit and MicroPython. A micro:bit is a tiny microcomputer with programmable LEDs, sensors, and more. Students will learn about coding by using a hands-on combination of interactive lessons, concept explanations, videos, puzzles, and DIY projects. This course engages students in developing computational thinking skills, as listed below from the CSTA Computer Science standards and UK Computing standards. The Common Core State Standards for Mathematics and English Language Arts that students develop are also listed here.

|  | Lesson 1 - Introduction | Lesson 2 - Emoji Maker | Lesson 3 - Track and Field | Lesson 4 - Prime Time | Lesson 5 - Temperature Search | Lesson 6 - Reflex Master |
|---|---|---|---|---|---|---|
| **Key Skills and Concepts** | • Learn how to program the micro:bit using MicroPython<br>• Use loops | • Apply infinite loops<br>• Use variables<br>• Apply knowledge of strings<br>• Program the micro:bit's LED to display an emoji | • Track footsteps using the micro:bit's sensors<br>• Track time using the micro:bit's buttons | • Demonstrate an understanding of prime numbers and square roots<br>• Use Python's math functions | • Create and call functions<br>• Program the micro:bit to return temperature values | • Create and call functions<br>• Program the micro:bit to display a message<br>• Use delays |
| **CCSS-Math Standards** | MP.1 | MP.1 | MP.1 | MP.1 | MP.1 | MP.1 |
| **CCSS-ELA Standards** | 6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7<br>RI.9-10.3<br>RI.11-12.3 | 6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7<br>RI.9-10.3<br>RI.11-12.3 | 6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7<br>RI.9-10.3<br>RI.11-12.3 | 6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7<br>RI.9-10.3<br>RI.11-12.3 | 6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7<br>RI.9-10.3<br>RI.11-12.3 | 6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7<br>RI.9-10.3<br>RI.11-12.3 |
| **CSTA Computer Science Standards** | 2-CS-02<br>2-CS-03<br>3A-CS-03<br>3A-DA-09<br>3B-CS-02<br>2-AP-11<br>2-AP-13<br>2-AP-15<br>2-AP-17<br>3A-AP-17<br>3A-AP-22<br>3B-AP-11<br>3B-AP-21<br>3B-AP-22 | 2-CS-02<br>2-CS-03<br>3A-CS-03<br>3A-DA-09<br>3B-CS-02<br>2-AP-11<br>2-AP-13<br>2-AP-15<br>2-AP-17<br>3A-AP-17<br>3A-AP-22<br>3B-AP-11<br>3B-AP-21<br>3B-AP-22 | 2-CS-02<br>2-CS-03<br>3A-CS-03<br>3A-DA-09<br>3B-CS-02<br>2-AP-11<br>2-AP-13<br>2-AP-15<br>2-AP-17<br>3A-AP-17<br>3A-AP-22<br>3B-AP-11<br>3B-AP-21<br>3B-AP-22 | 2-CS-02<br>2-CS-03<br>3A-CS-03<br>3A-DA-09<br>3B-CS-02<br>2-AP-11<br>2-AP-13<br>2-AP-15<br>2-AP-17<br>3A-AP-17<br>3A-AP-22<br>3B-AP-11<br>3B-AP-21<br>3B-AP-22 | 2-CS-02<br>2-CS-03<br>3A-CS-03<br>3A-DA-09<br>3B-CS-02<br>2-AP-11<br>2-AP-13<br>2-AP-15<br>2-AP-17<br>3A-AP-17<br>3A-AP-22<br>3B-AP-11<br>3B-AP-21<br>3B-AP-22 | 2-CS-01<br>2-CS-02<br>2-CS-03<br>3A-CS-03<br>3A-DA-09<br>3B-CS-02<br>2-AP-11<br>2-AP-13<br>2-AP-15<br>2-AP-17<br>3A-AP-17<br>3A-AP-22<br>3B-AP-11<br>3B-AP-21 |

|  | 3B-AP-23 | 3B-AP-23 | 3B-AP-23 | 3B-AP-23 | 3B-AP-23 | 3B-AP-22 3B-AP-23 |
|---|---|---|---|---|---|---|
| **ISTE Standards** | 1.c, 1.d, 4.d, 5.c, 5.d, 6.b | 1.c, 1.d, 4.d, 5.c, 5.d, 6.b | 1.c, 1.d, 4.d, 5.c, 5.d, 6.b | 1.c, 1.d, 4.d, 5.c, 5.d, 6.b | 1.c, 1.d, 4.d, 5.c, 5.d, 6.b | 1.c, 1.d, 4.d, 5.c, 5.d, 6.b |
| **UK National Curriculum** | Keystage 2* | Keystage 2* | Keystage 2* | Keystage 2* | Keystage 2* | Keystage 2* |
| **Sample Application of Skills** | Create a text scrolling project. | Create animated projects using the micro:bit. | Create interactive pedometer and lap timer projects. | Create math-themed projects. | Program the micro:bit to sense temperature changes. | Create a game on the micro:bit that tests the user's reaction time. |

*See individual lesson guides for details on UK Computer standards

# MicroPython 101
## Scope and Sequence

**Grades 6+**

Each lesson takes about 45-60 minutes to complete.

The MicroPython 101 course introduces students to physical computing using micro:bit and MicroPython. A micro:bit is a tiny microcomputer with programmable LEDs, sensors, and more. Students will learn about coding by using a hands-on combination of interactive lessons, concept explanations, videos, puzzles, and DIY projects. This course engages students in developing computational thinking skills, as listed below from the CSTA Computer Science standards and UK Computing standards. The Common Core State Standards for Mathematics and English Language Arts that students develop are also listed here.

|  | Lesson 7 - Maze Madness | Lesson 8 - High Rollers | Lesson 9 - Soundboard | Lesson 10 - Button Masher | Lesson 11 - Dodgeball | Lesson 12 - Guess the Word |
|---|---|---|---|---|---|---|
| **Key Skills and Concepts** | • Attach pins to the micro:bit<br>• Demonstrate an understanding of input/output | • Create and use variables<br>• Set the value of a variable<br>• Use random numbers | • Make the micro:bit play sounds<br>• Create and code a tune | • Program the micro:bit to detect how many times the user presses each button<br>• Use loops | • Use arrays<br>• Create and call functions | • Create and call functions<br>• Use lists<br>• Determine the length of a string |
| **CCSS-Math Standards** | MP.1 | MP.1 | MP.1 | MP.1 | MP.1 | MP.1 |
| **CCSS-ELA Standards** | 6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7<br>RI.9-10.3<br>RI.11-12.3 | 6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7<br>RI.9-10.3<br>RI.11-12.3 | 6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7<br>RI.9-10.3<br>RI.11-12.3 | 6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7<br>RI.9-10.3<br>RI.11-12.3 | 6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7<br>RI.9-10.3<br>RI.11-12.3 | 6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7<br>RI.9-10.3<br>RI.11-12.3 |
| **CSTA Computer Science Standards** | 2-CS-02<br>2-CS-03<br>3A-CS-03<br>3A-DA-09<br>3B-CS-02<br>2-AP-11<br>2-AP-13<br>2-AP-15<br>2-AP-17<br>3A-AP-17<br>3A-AP-22<br>3B-AP-11<br>3B-AP-21<br>3B-AP-22 | 2-CS-02<br>2-CS-03<br>3A-CS-03<br>3A-DA-09<br>3B-CS-02<br>2-AP-11<br>2-AP-13<br>2-AP-15<br>2-AP-17<br>3A-AP-17<br>3A-AP-22<br>3B-AP-11<br>3B-AP-21<br>3B-AP-22 | 2-CS-02<br>2-CS-03<br>3A-CS-03<br>3A-DA-09<br>3B-CS-02<br>2-AP-11<br>2-AP-13<br>2-AP-15<br>2-AP-17<br>3A-AP-17<br>3A-AP-22<br>3B-AP-11<br>3B-AP-21<br>3B-AP-22 | 2-CS-02<br>2-CS-03<br>3A-CS-03<br>3A-DA-09<br>3B-CS-02<br>2-AP-11<br>2-AP-13<br>2-AP-15<br>2-AP-17<br>3A-AP-17<br>3A-AP-22<br>3B-AP-11<br>3B-AP-21<br>3B-AP-22 | 2-CS-02<br>2-CS-03<br>3A-CS-03<br>3A-DA-09<br>3B-CS-02<br>2-AP-11<br>2-AP-13<br>2-AP-15<br>2-AP-17<br>3A-AP-17<br>3A-AP-22<br>3B-AP-09<br>3B-AP-11<br>3B-AP-21 | 2-CS-02<br>2-CS-03<br>3A-CS-03<br>3A-DA-09<br>3B-CS-02<br>2-AP-11<br>2-AP-13<br>2-AP-15<br>2-AP-17<br>3A-AP-17<br>3A-AP-22<br>3B-AP-09<br>3B-AP-11<br>3B-AP-21 |

|  | 3B-AP-23 | 3B-AP-23 | 3B-AP-23 | 3B-AP-23 | 3B-AP-22 3B-AP-23 | 3B-AP-22 3B-AP-23 |
|---|---|---|---|---|---|---|
| **ISTE Standards** | 1.c, 1.d, 4.d, 5.c, 5.d, 6.b | 1.c, 1.d, 4.d, 5.c, 5.d, 6.b | 1.c, 1.d, 4.d, 5.c, 5.d, 6.b | 1.c, 1.d, 4.d, 5.c, 5.d, 6.b | 1.c, 1.d, 4.d, 5.c, 5.d, 6.b | 1.c, 1.d, 4.d, 5.c, 5.d, 6.b |
| **UK National Curriculum** | Keystage 2* | Keystage 2* | Keystage 2* | Keystage 2* | Keystage 2* | Keystage 2* |
| **Sample Application of Skills** | Create a maze game. | Create a virtual dice roller using variables and random numbers. | Make voice effects using the micro:bit | Create a 2-player game where the plate rwho clicks the button the fastest wins. | Create a dodgeball game on the micro:bit. | Create a game where the user has to guess the mystery word. |

*See individual lesson guides for details on UK Computer standard

# MicroPython 101

# Grades 6+

## Scope and Sequence

Each lesson takes about 45-60 minutes to complete.

The MicroPython 101 course introduces students to physical computing using micro:bit and MicroPython. A micro:bit is a tiny microcomputer with programmable LEDs, sensors, and more. Students will learn about coding by using a hands-on combination of interactive lessons, concept explanations, videos, puzzles, and DIY projects. This course engages students in developing computational thinking skills, as listed below from the CSTA Computer Science standards and UK Computing standards. The Common Core State Standards for Mathematics and English Language Arts that students develop are also listed here.

| | Lesson 13 - Morse Code | Lesson 14 - Treasure Hunt | | | | |
|---|---|---|---|---|---|---|
| Key Skills and Concepts | • Import the radio library<br>• Program the micro:bit to receive/return messages | • Create multiplayer games using the micro:bit<br>• Program the micro:bits to interact with one another<br>• Display an image on the micro:bit | | | | |
| CCSS-Math Standards | MP.1 | MP.1 | | | | |
| CCSS-ELA Standards | 6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7<br>RI.9-10.3<br>RI.11-12.3 | 6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7<br>RI.9-10.3<br>RI.11-12.3 | | | | |
| CSTA Computer Science Standards | 2-CS-02<br>2-CS-03<br>3A-CS-03<br>3A-DA-09<br>3B-CS-02<br>2-AP-11<br>2-AP-13<br>2-AP-15<br>2-AP-17<br>3A-AP-17<br>3A-AP-22<br>3B-AP-09<br>3B-AP-11 | 2-CS-01<br>2-CS-02<br>2-CS-03<br>3A-CS-03<br>3A-DA-09<br>3B-CS-02<br>2-AP-11<br>2-AP-13<br>2-AP-15<br>2-AP-17<br>3A-AP-17<br>3A-AP-22<br>3B-AP-09 | | | | |

| | | | | | | |
|---|---|---|---|---|---|---|
| | 3B-AP-21<br>3B-AP-22<br>3B-AP-23 | 3B-AP-11<br>3B-AP-21<br>3B-AP-22<br>3B-AP-23 | | | | |
| **ISTE Standards** | 1.c, 1.d, 4.d, 5.c, 5.d, 6.b | 1.c, 1.d, 4.d, 5.c, 5.d, 6.b | | | | |
| **UK National Curriculum** | Keystage 2* | Keystage 2* | | | | |
| **Sample Application of Skills** | Use radio functions to create a project that sends and receives morse code messages. | Use multiple micro:bits to create an interactive treasure hunt game. | | | | |

*See individual lesson guides for details on UK Computer standards

# Web Development 101

## Scope and Sequence

**Grades 7+**

Each lesson takes about 45-60 minutes to complete.

Web Development 101 is a course for students in grades 7 or above. The activities, puzzles, and projects engage students in developing computational thinking skills, as listed below from the CSTA Computer Science standards and UK Computing standards. The Common Core State Standards for Mathematics and English Language Arts that students develop are also listed here.

|  | Lesson 1 - Introduction | Lesson 2 - Headings and Images | Lesson 3 - All About Lists | Lesson 4 - Adding Hyperlinks | Lesson 5 - Using Containers | Lesson 6 - Tables and Media | Lesson 7 - More on Styling | Lesson 8 - Pixel Art |
|---|---|---|---|---|---|---|---|---|
| **Key Skills and Concepts** | • Apply basic HTML and CSS concepts | • Use images and font styles | • Use ordered and unordered lists to organize content | • Use hyperlinks to link to other pages | • Use container elements such as div and section | • Apply tables to format content | • Use forms and CSS features for advanced layouts and designs | • Apply CSS and divs |
| **CCSS-Math Standards** | MP.1 MP.7 | MP.1 MP.2 MP.4 MP.7 | MP.1 MP.2 MP.4 MP.7 | MP.1 MP.4 | MP.1 MP.2 MP.4 MP.7 | MP.1 MP.2 MP.4 MP.7 | MP.1 MP.2 MP.4 MP.7 | MP.1 MP.2 MP.4 MP.7 |
| **CCSS-ELA Standards** | 7.RI.4 8.RI.4 6-8.RST.3 6-8.RST.4 6-8.RST.7 RI.9-10.3 RI.11-12.3 | 7.RI.4 8.RI.4 6-8.RST.3 6-8.RST.4 6-8.RST.7 RI.9-10.3 RI.11-12.3 | 7.RI.4 8.RI.4 6-8.RST.3 6-8.RST.4 6-8.RST.7 RI.9-10.3 RI.11-12.3 | 7.RI.4 8.RI.4 6-8.RST.3 6-8.RST.4 6-8.RST.7 RI.9-10.3 RI.11-12.3 | 7.RI.4 8.RI.4 6-8.RST.3 6-8.RST.4 6-8.RST.7 RI.9-10.3 RI.11-12.3 | 7.RI.4 8.RI.4 6-8.RST.3 6-8.RST.4 6-8.RST.7 RI.9-10.3 RI.11-12.3 | 7.RI.4 8.RI.4 6-8.RST.3 6-8.RST.4 6-8.RST.7 RI.9-10.3 RI.11-12.3 | 7.RI.4 8.RI.4 6-8.RST.3 6-8.RST.4 6-8.RST.7 RI.9-10.3 RI.11-12.3 |
| **CSTA Computer Science Standards** | 2-AP-10 2-AP-13 2-AP-16 2-AP-17 3A-AP-17 3A-AP-22 | 2-AP-13 2-AP-16 2-AP-17 3A-AP-17 3A-AP-22 | 2-AP-13 2-AP-16 2-AP-17 3A-AP-17 3A-AP-22 | 2-AP-10 2-AP-13 2-AP-16 2-AP-17 3A-AP-17 3A-AP-22 | 2-AP-10 2-AP-13 2-AP-16 2-AP-17 2-AP-19 3A-AP-17 3A-AP-22 | 2-AP-10 2-AP-13 2-AP-17 3A-AP-22 | 2-AP-13 2-AP-14 2-AP-16 2-AP-17 3A-AP-17 3A-AP-22 | 2-AP-13 2-AP-16 2-AP-17 3A-AP-17 3A-AP-22 3B-AP-11 3B-AP-12 3B-AP-21 3B-AP-22 |
| **ISTE Standards** | 1.c, 1.d, 4.d, 5.c, 5.d, 6.b | 1.c, 1.d, 4.d, 5.c, 5.d, 6.b | 1.c, 1.d, 4.d, 5.c, 5.d, 6.b | 1.c, 1.d, 4.d, 5.c, 5.d, 6.b | 1.c, 1.d, 4.d, 5.c, 5.d, 6.b | 1.c, 1.d, 4.d, 5.c, 5.d, 6.b | 1.c, 1.d, 4.d, 5.c, 5.d, 6.b | 1.c, 1.d, 4.d, 5.c, 5.d, 6.b |
| **UK National** | Keystage 3 Computing* | Keystage 3 Computing* | Keystage 3 Computing* | Keystage 3 Computing* | Keystage 3 Computing* | Keystages 3 Computing* | Keystage 3 Computing* | Keystage 3 Computing* |

| Curriculum | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Sample Application of Skills** | Create a simple web page. | Create a poem, poster, and photo gallery. | Create a My Pets web page with lists and images. | Create a Home Page Navigation project. | Create a Hero Unit project. | Create a My Channels Page project. | Create a profile card and Responsive Page project. | Create a Space Invaders Art project and Game World Creator project. |

*See individual lesson guides for details on UK Computer standards

# Web Development 101
## Scope and Sequence

# Grades 7+

Each lesson takes about 45-60 minutes to complete.

Web Development 101 is a course for students in grades 7 or above. The activities, puzzles, and projects engage students in developing computational thinking skills, as listed below from the CSTA Computer Science standards and UK Computing standards. The Common Core State Standards for Mathematics and English Language Arts that students develop are also listed here.

| | Lesson 9 - Photo Gallery | Lesson 10 - Pictogram | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Key Skills and Concepts** | • Use divs, spans, sections, hyperlinks, headings, and images | • Use divs, spans, sections, hyperlinks, headings, and images | | | | | | | |
| **CCSS-Math Standards** | MP.1 MP.2 MP.4 MP.7 | MP.1 MP.2 MP.4 MP.7 | | | | | | | |
| **CCSS-ELA Standards** | 7.RI.4 8.RI.4 6-8.RST.3 6-8.RST.4 6-8.RST.7 RI.9-10.3 RI.11-12.3 | 7.RI.4 8.RI.4 6-8.RST.3 6-8.RST.4 6-8.RST.7 RI.9-10.3 RI.11-12.3 | | | | | | | |
| **CSTA Computer Science Standards** | 2-AP-13 2-AP-16 2-AP-17 3A-AP-17 3A-AP-22 3A-DA-09 3A-IC-29 3B-AP-10 3B-AP-11 3B-AP-12 3B-AP-21 3B-AP-22 | 2-AP-13 2-AP-16 2-AP-17 3A-AP-17 3A-AP-22 3A-NI-06 3B-AP-10 3B-AP-11 3B-AP-12 3B-AP-21 3B-AP-22 | | | | | | | |
| **ISTE Standards** | 1.c, 1.d, 4.d, 5.c, | 1.c, 1.d, 4.d, 5.c, | | | | | | | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 5.d, 6.b | 5.d, 6.b | | | | | | |
| **UK National Curriculum** | Keystages 2 & 3 Computing* | Keystage 3 Computing* | | | | | | |
| **Sample Application of Skills** | Create a photo gallery web page | Create an Instagram-influenced photo gallery with clickable pictures that users can vote on. | | | | | | |

*See individual lesson guides for details on UK Computer standards

# Artificial Intelligence 401

## Scope and Sequence

**Grades 7+**

Each lesson takes about 45-60 minutes to complete.

Artificial Intelligence 401 is a JavaScript and p5.js based course that's perfect for experienced students looking to implement AI in their programs. They learn how to use MediaPipe, an AI solution from Google. Additionally, the activities and projects engage students in developing computational thinking skills, as listed below from the CSTA Computer Science standards and UK Computing standards. The Common Core State Standards for Mathematics and English Language Arts that students develop are also listed here.

| | Lesson 1.1 - What is AI? | Lesson 1.2 - Intro to JavaScript | Lesson 1.3 - Intro to p5.js | Lesson 1.4 - Objects in JavaScript | Lesson 1.5 - Design an Emoji | Lesson 1.6 - Review | Lesson 1.7 - Quiz | Lesson 2.1 - Training AI Models |
|---|---|---|---|---|---|---|---|---|
| **Key Skills and Concepts** | • Artificial intelligence<br>• Machine learning | • JavaScript commands<br>• Comments<br>• Syntax<br>• Functions | • p5.js<br>• Functions<br>• Create shapes<br>• Draw on the canvas | • Classes<br>• Objects | • Draw shapes<br>• Colors, strokes, stroke weight | • Artificial intelligence<br>• Machine learning<br>• Syntax<br>• Variables<br>• Functions<br>• Objects | • Artificial intelligence<br>• Machine learning<br>• Syntax<br>• Variables<br>• Functions<br>• Objects | • Supervised learning AI models<br>• Arrays |
| **CCSS-Math Standards** | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3<br>MP.1<br>MP.2<br>MP.4 | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3<br>MP.1<br>MP.2<br>MP.4 | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3<br>MP.1<br>MP.2<br>MP.4 | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3<br>MP.1<br>MP.2<br>MP.4 | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3<br>MP.1<br>MP.2<br>MP.4 | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3<br>MP.1<br>MP.2<br>MP.4 | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3<br>MP.1<br>MP.2<br>MP.4 | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3<br>MP.1<br>MP.2<br>MP.4 |
| **CCSS-ELA Standards** | SL.8.1<br>RI.9-10.3<br>RI.9-10.6<br>RI.11-12.3<br>RI.11-12.6<br>L.9-10.3<br>L.9-10.6<br>L.11-12.3<br>L.11-12.6 | SL.8.1<br>RI.9-10.3<br>RI.9-10.6<br>RI.11-12.3<br>RI.11-12.6<br>L.9-10.3<br>L.9-10.6<br>L.11-12.3<br>L.11-12.6 | SL.8.1<br>RI.9-10.3<br>RI.9-10.6<br>RI.11-12.3<br>RI.11-12.6<br>L.9-10.3<br>L.9-10.6<br>L.11-12.3<br>L.11-12.6 | SL.8.1<br>RI.9-10.3<br>RI.9-10.6<br>RI.11-12.3<br>RI.11-12.6<br>L.9-10.3<br>L.9-10.6<br>L.11-12.3<br>L.11-12.6 | SL.8.1<br>RI.9-10.3<br>RI.9-10.6<br>RI.11-12.3<br>RI.11-12.6<br>L.9-10.3<br>L.9-10.6<br>L.11-12.3<br>L.11-12.6 | SL.8.1<br>RI.9-10.3<br>RI.9-10.6<br>RI.11-12.3<br>RI.11-12.6<br>L.9-10.3<br>L.9-10.6<br>L.11-12.3<br>L.11-12.6 | SL.8.1<br>RI.9-10.3<br>RI.9-10.6<br>RI.11-12.3<br>RI.11-12.6<br>L.9-10.3<br>L.9-10.6<br>L.11-12.3<br>L.11-12.6 | SL.8.1<br>RI.9-10.3<br>RI.9-10.6<br>RI.11-12.3<br>RI.11-12.6<br>L.9-10.3<br>L.9-10.6<br>L.11-12.3<br>L.11-12.6 |
| **CSTA Computer Science Standards** | 2-AP-11<br>2-AP-13<br>2-AP-17<br>3A-AP-17<br>3B-AP-11 | 2-AP-11<br>2-AP-13<br>2-AP-17<br>3A-AP-17<br>3B-AP-11 | 2-AP-11<br>2-AP-13<br>2-AP-17<br>3A-AP-17<br>3B-AP-11 | 2-AP-11<br>2-AP-13<br>2-AP-17<br>3A-AP-17<br>3B-AP-11 | 2-AP-11<br>2-AP-13<br>2-AP-17<br>3A-AP-17<br>3B-AP-11 | 2-AP-11<br>2-AP-13<br>2-AP-17<br>3A-AP-17<br>3B-AP-11 | 2-AP-11<br>2-AP-13<br>2-AP-17<br>3A-AP-17<br>3B-AP-11 | 2-AP-11<br>2-AP-13<br>2-AP-17<br>3A-AP-17<br>3B-AP-11 |
| **ISTE Standards** | 1.c, 1.d, 4.d, 5.c, | 1.c, 1.d, 4.d, 5.c, | 1.c, 1.d, 4.d, 5.c, | 1.c, 1.d, 4.d, 5.c, | 1.c, 1.d, 4.d, 5.c, | 1.c, 1.d, 4.d, 5.c, | 1.c, 1.d, 4.d, 5.c, | 1.c, 1.d, 4.d, 5.c, |

| | 5.d | 5.d | 5.d | 5.d | 5.d | 5.d | 5.d | 5.d |
|---|---|---|---|---|---|---|---|---|
| **UK National Curriculum** | Key Stage 3 & 4 Computing* | Key Stage 3 & 4 Computing* | Key Stage 3 & 4 Computing* | Key Stage 3 & 4 Computing* | Key Stage 3 & 4 Computing* | Key Stage 3 & 4 Computing* | Key Stage 3 & 4 Computing* | Key Stage 3 & 4 Computing* |
| **Sample Application of Skills** | Explore what AI is, the history of AI, and its applications. | Solve coding puzzles in JavaScript. | Draw shapes and create animations using p5.js | Create and manipulate objects in JavaScript. | Create an emoji using the p5.js library. | Review Unit 1 concepts. | Take a quiz on Unit 1 concepts. | Train and test a supervised learning classification model. |

*See individual lesson guides for details on UK Computer standards

# Artificial Intelligence 401

## Scope and Sequence

## Grades 7+

Each lesson takes about 45-60 minutes to complete.

Artificial Intelligence 401 is a JavaScript and p5.js based course that's perfect for experienced students looking to implement AI in their programs. They learn how to use MediaPipe, an AI solution from Google. Additionally, the activities and projects engage students in developing computational thinking skills, as listed below from the CSTA Computer Science standards and UK Computing standards. The Common Core State Standards for Mathematics and English Language Arts that students develop are also listed here.

| | Lesson 2.2 - Detecting Objects with Computers | Lesson 2.3 - Peek-a-Boo | Lesson 2.4 - Handwriting Recognition | Lesson 2.5 - Neural Networks | Lesson 2.6 - Review | Lesson 2.7 - Quiz | Lesson 3.1 - Making Models | Lesson 3.2 - What Makes a Model Good? |
|---|---|---|---|---|---|---|---|---|
| **Key Skills and Concepts** | • Hand gestures<br>• Webcam input<br>• Data labels<br>• Training data | • Functions<br>• AI models | • Handwriting recognition AI<br>• Backpropagation<br>• AI model training | • Neural networks<br>• Backpropaga-tion | • Training and testing models<br>• Neural networks<br>• Matrices<br>• Backpropagation | • Training and testing models<br>• Neural networks<br>• Matrices<br>• Backpropagation | • Pretrained models<br>• Biased training data | • Accuracy<br>• Quality of an AI model<br>• True/false positive<br>• True/false negative<br>• Bias |
| **CCSS-Math Standards** | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3<br>MP.1<br>MP.2<br>MP.4 | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3<br>MP.1<br>MP.2<br>MP.4 | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3<br>MP.1<br>MP.2<br>MP.4 | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3<br>MP.1<br>MP.2<br>MP.4 | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3<br>MP.1<br>MP.2<br>MP.4 | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3<br>MP.1<br>MP.2<br>MP.4 | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3<br>MP.1<br>MP.2<br>MP.4 | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3<br>MP.1<br>MP.2<br>MP.4 |
| **CCSS-ELA Standards** | SL.8.1<br>RI.9-10.3<br>RI.9-10.6<br>RI.11-12.3<br>RI.11-12.6<br>L.9-10.3<br>L.9-10.6<br>L.11-12.3<br>L.11-12.6 | SL.8.1<br>RI.9-10.3<br>RI.9-10.6<br>RI.11-12.3<br>RI.11-12.6<br>L.9-10.3<br>L.9-10.6<br>L.11-12.3<br>L.11-12.6 | SL.8.1<br>RI.9-10.3<br>RI.9-10.6<br>RI.11-12.3<br>RI.11-12.6<br>L.9-10.3<br>L.9-10.6<br>L.11-12.3<br>L.11-12.6 | SL.8.1<br>RI.9-10.3<br>RI.9-10.6<br>RI.11-12.3<br>RI.11-12.6<br>L.9-10.3<br>L.9-10.6<br>L.11-12.3<br>L.11-12.6 | SL.8.1<br>RI.9-10.3<br>RI.9-10.6<br>RI.11-12.3<br>RI.11-12.6<br>L.9-10.3<br>L.9-10.6<br>L.11-12.3<br>L.11-12.6 | SL.8.1<br>RI.9-10.3<br>RI.9-10.6<br>RI.11-12.3<br>RI.11-12.6<br>L.9-10.3<br>L.9-10.6<br>L.11-12.3<br>L.11-12.6 | SL.8.1<br>RI.9-10.3<br>RI.9-10.6<br>RI.11-12.3<br>RI.11-12.6<br>L.9-10.3<br>L.9-10.6<br>L.11-12.3<br>L.11-12.6 | SL.8.1<br>RI.9-10.3<br>RI.9-10.6<br>RI.11-12.3<br>RI.11-12.6<br>L.9-10.3<br>L.9-10.6<br>L.11-12.3<br>L.11-12.6 |

| CSTA Computer Science Standards | 2-AP-11 2-AP-13 2-AP-17 3A-AP-17 3B-AP-11 | 2-AP-11 2-AP-13 2-AP-17 3A-AP-17 3B-AP-11 | 2-AP-11 2-AP-13 2-AP-17 3A-AP-17 3B-AP-11 | 2-AP-11 2-AP-13 2-AP-17 3A-AP-17 3B-AP-11 | 2-AP-11 2-AP-13 2-AP-17 3A-AP-17 3B-AP-11 | 2-AP-11 2-AP-13 2-AP-17 3A-AP-17 3B-AP-11 | 2-AP-11 2-AP-13 2-AP-17 3A-AP-17 3B-AP-11 | 2-AP-11 2-AP-13 2-AP-17 3A-AP-17 3B-AP-11 |
|---|---|---|---|---|---|---|---|---|
| ISTE Standards | 1.c, 1.d, 4.d, 5.c, 5.d | 1.c, 1.d, 4.d, 5.c, 5.d | 1.c, 1.d, 4.d, 5.c, 5.d | 1.c, 1.d, 4.d, 5.c, 5.d | 1.c, 1.d, 4.d, 5.c, 5.d | 1.c, 1.d, 4.d, 5.c, 5.d | 1.c, 1.d, 4.d, 5.c, 5.d | 1.c, 1.d, 4.d, 5.c, 5.d |
| UK National Curriculum | Key Stage 3 & 4 Computing* | Key Stage 3 & 4 Computing* | Key Stage 3 & 4 Computing* | Key Stage 3 & 4 Computing* | Key Stage 3 & 4 Computing* | Key Stage 3 & 4 Computing* | Key Stage 3 & 4 Computing* | Key Stage 3 & 4 Computing* |
| Sample Application of Skills | Train a model with hand gestures. | Create an interactive game. | Train an AI model to recognize handwritten digits. | Demonstrate an understanding of the basic structure of an artificial neural network. | Review Unit 2 concepts. | Take a quiz on Unit 2 concepts. | Train a model to differentiate between different hand gestures using a webcam. | Evaluate the speed and accuracy of a model's predictions. |

*See individual lesson guides for details on UK Computer standards

# Artificial Intelligence 401

**Grades 7+**

## Scope and Sequence

Each lesson takes about 45-60 minutes to complete.

Artificial Intelligence 401 is a JavaScript and p5.js based course that's perfect for experienced students looking to implement AI in their programs. They learn how to use MediaPipe, an AI solution from Google. Additionally, the activities and projects engage students in developing computational thinking skills, as listed below from the CSTA Computer Science standards and UK Computing standards. The Common Core State Standards for Mathematics and English Language Arts that students develop are also listed here.

| | Lesson 3.3 - Real Time Hand Tracking | Lesson 3.4 - Finger Calculator | Lesson 3.5 - Hand Landmarks | Lesson 3.6 - Finger Painter | Lesson 3.7 - Maze Game | Lesson 3.8 - Review | Lesson 3.9 - Quiz | Lesson 4.1 - AI in Games |
|---|---|---|---|---|---|---|---|---|
| **Key Skills and Concepts** | • Functions<br>• Video frame rate<br>• Motion blurring<br>• Ghost matching | • AI games and simulators | • Hand gestures in AI<br>• Hand landmarks<br>• Complex models | • Hand landmark locations<br>• Input | • Hand landmark locations<br>• Input | • Pretrained models<br>• Accuracy<br>• Bias<br>• Hand tracking landmarks<br>• Functions | • Pretrained models<br>• Accuracy<br>• Bias<br>• Hand tracking landmarks<br>• Functions | • AI inputs<br>• Game board states<br>• Reinforcement learning |
| **CCSS-Math Standards** | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3<br>MP.1<br>MP.2<br>MP.4 | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3<br>MP.1<br>MP.2<br>MP.4 | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3<br>MP.1<br>MP.2<br>MP.4 | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3<br>MP.1<br>MP.2<br>MP.4 | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3<br>MP.1<br>MP.2<br>MP.4 | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3<br>MP.1<br>MP.2<br>MP.4 | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3<br>MP.1<br>MP.2<br>MP.4 | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3<br>MP.1<br>MP.2<br>MP.4 |
| **CCSS-ELA Standards** | SL.8.1<br>RI.9-10.3<br>RI.9-10.6<br>RI.11-12.3<br>RI.11-12.6<br>L.9-10.3<br>L.9-10.6<br>L.11-12.3<br>L.11-12.6 | SL.8.1<br>RI.9-10.3<br>RI.9-10.6<br>RI.11-12.3<br>RI.11-12.6<br>L.9-10.3<br>L.9-10.6<br>L.11-12.3<br>L.11-12.6 | SL.8.1<br>RI.9-10.3<br>RI.9-10.6<br>RI.11-12.3<br>RI.11-12.6<br>L.9-10.3<br>L.9-10.6<br>L.11-12.3<br>L.11-12.6 | SL.8.1<br>RI.9-10.3<br>RI.9-10.6<br>RI.11-12.3<br>RI.11-12.6<br>L.9-10.3<br>L.9-10.6<br>L.11-12.3<br>L.11-12.6 | SL.8.1<br>RI.9-10.3<br>RI.9-10.6<br>RI.11-12.3<br>RI.11-12.6<br>L.9-10.3<br>L.9-10.6<br>L.11-12.3<br>L.11-12.6 | SL.8.1<br>RI.9-10.3<br>RI.9-10.6<br>RI.11-12.3<br>RI.11-12.6<br>L.9-10.3<br>L.9-10.6<br>L.11-12.3<br>L.11-12.6 | SL.8.1<br>RI.9-10.3<br>RI.9-10.6<br>RI.11-12.3<br>RI.11-12.6<br>L.9-10.3<br>L.9-10.6<br>L.11-12.3<br>L.11-12.6 | SL.8.1<br>RI.9-10.3<br>RI.9-10.6<br>RI.11-12.3<br>RI.11-12.6<br>L.9-10.3<br>L.9-10.6<br>L.11-12.3<br>L.11-12.6 |
| **CSTA Computer Science Standards** | 2-AP-11<br>2-AP-13<br>2-AP-17<br>3A-AP-17<br>3B-AP-11 | 2-AP-11<br>2-AP-13<br>2-AP-17<br>3A-AP-17<br>3B-AP-11 | 2-AP-11<br>2-AP-13<br>2-AP-17<br>3A-AP-17<br>3B-AP-11 | 2-AP-11<br>2-AP-13<br>2-AP-17<br>3A-AP-17<br>3B-AP-11 | 2-AP-11<br>2-AP-13<br>2-AP-17<br>3A-AP-17<br>3B-AP-11 | 2-AP-11<br>2-AP-13<br>2-AP-17<br>3A-AP-17<br>3B-AP-11 | 2-AP-11<br>2-AP-13<br>2-AP-17<br>3A-AP-17<br>3B-AP-11 | 2-AP-11<br>2-AP-13<br>2-AP-17<br>3A-AP-17<br>3B-AP-11 |
| **ISTE Standards** | 1.c, 1.d, 4.d, 5.c, | 1.c, 1.d, 4.d, 5.c, | 1.c, 1.d, 4.d, 5.c, | 1.c, 1.d, 4.d, 5.c, | 1.c, 1.d, 4.d, 5.c, | 1.c, 1.d, 4.d, 5.c, | 1.c, 1.d, 4.d, 5.c, | 1.c, 1.d, 4.d, 5.c, |

|  | 5.d | 5.d | 5.d | 5.d | 5.d | 5.d | 5.d | 5.d |
|---|---|---|---|---|---|---|---|---|
| **UK National Curriculum** | Key Stage 3 & 4 Computing* | Key Stage 3 & 4 Computing* | Key Stage 3 & 4 Computing* | Key Stage 3 & 4 Computing* | Key Stage 3 & 4 Computing* | Key Stage 3 & 4 Computing* | Key Stage 3 & 4 Computing* | Key Stage 3 & 4 Computing* |
| **Sample Application of Skills** | Create and use a model to track hand movements. | Create a finger-counting AI program using real-time hand tracking model. | Learn how landmarks are used in the hand tracking model. | Create a finger painter project. | Create a maze game. | Review Unit 3 concepts. | Take a quiz on Unit 3 concepts. | Interact with an advanced Tic Tac Toe AI. |

*See individual lesson guides for details on UK Computer standards

# Artificial Intelligence 401

**Grades 7+**

## Scope and Sequence

Each lesson takes about 45-60 minutes to complete.

Artificial Intelligence 401 is a JavaScript and p5.js based course that's perfect for experienced students looking to implement AI in their programs. They learn how to use MediaPipe, an AI solution from Google. Additionally, the activities and projects engage students in developing computational thinking skills, as listed below from the CSTA Computer Science standards and UK Computing standards. The Common Core State Standards for Mathematics and English Language Arts that students develop are also listed here.

| | Lesson 4.2 - How Can AI Learn Tasks? | Lesson 4.3 - Intro to Reinforcement Learning | Lesson 4.4 - Flappy Bird | Lesson 4.5 - Review | Lesson 4.6 - Quiz | Lesson 5.1 - AI and The Future | Lesson 5.2 - Impacts of AI | Lesson 5.3 - Conversation with a Chatbot |
|---|---|---|---|---|---|---|---|---|
| **Key Skills and Concepts** | • Velocity<br>• Angular velocity<br>• Input/output | • Reinforcement learning<br>• E-greedy algorithm<br>• Q-learning | • Neural network<br>• AI games<br>• Algorithms | • States<br>• Actions<br>• Rewards<br>• Policies<br>• Neural networks<br>• Q-learning | • States<br>• Actions<br>• Rewards<br>• Policies<br>• Neural networks<br>• Q-learning | • Ethics<br>• Filter bubble<br>• Singularity | • Applications of AI in various fields<br>• Research<br>• Beneficial and harmful effects of AI | • Large Language Models (LLMs)<br>• Parameters<br>• Overfitting<br>• Confirmation bias |
| **CCSS-Math Standards** | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3<br>MP.1<br>MP.2<br>MP.4 | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3<br>MP.1<br>MP.2<br>MP.4 | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3<br>MP.1<br>MP.2<br>MP.4 | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3<br>MP.1<br>MP.2<br>MP.4 | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3<br>MP.1<br>MP.2<br>MP.4 | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3<br>MP.1<br>MP.2<br>MP.4 | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3<br>MP.1<br>MP.2<br>MP.4 | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3<br>MP.1<br>MP.2<br>MP.4 |
| **CCSS-ELA Standards** | SL.8.1<br>RI.9-10.3<br>RI.9-10.6<br>RI.11-12.3<br>RI.11-12.6<br>L.9-10.3<br>L.9-10.6<br>L.11-12.3<br>L.11-12.6 | SL.8.1<br>RI.9-10.3<br>RI.9-10.6<br>RI.11-12.3<br>RI.11-12.6<br>L.9-10.3<br>L.9-10.6<br>L.11-12.3<br>L.11-12.6 | SL.8.1<br>RI.9-10.3<br>RI.9-10.6<br>RI.11-12.3<br>RI.11-12.6<br>L.9-10.3<br>L.9-10.6<br>L.11-12.3<br>L.11-12.6 | SL.8.1<br>RI.9-10.3<br>RI.9-10.6<br>RI.11-12.3<br>RI.11-12.6<br>L.9-10.3<br>L.9-10.6<br>L.11-12.3<br>L.11-12.6 | SL.8.1<br>RI.9-10.3<br>RI.9-10.6<br>RI.11-12.3<br>RI.11-12.6<br>L.9-10.3<br>L.9-10.6<br>L.11-12.3<br>L.11-12.6 | SL.8.1<br>RI.9-10.3<br>RI.9-10.6<br>RI.11-12.3<br>RI.11-12.6<br>L.9-10.3<br>L.9-10.6<br>L.11-12.3<br>L.11-12.6 | SL.8.1<br>RI.9-10.3<br>RI.9-10.6<br>RI.11-12.3<br>RI.11-12.6<br>L.9-10.3<br>L.9-10.6<br>L.11-12.3<br>L.11-12.6 | SL.8.1<br>RI.9-10.3<br>RI.9-10.6<br>RI.11-12.3<br>RI.11-12.6<br>L.9-10.3<br>L.9-10.6<br>L.11-12.3<br>L.11-12.6 |
| **CSTA Computer Science Standards** | 2-AP-11<br>2-AP-13<br>2-AP-17<br>3A-AP-17<br>3B-AP-11 | 2-AP-11<br>2-AP-13<br>2-AP-17<br>3A-AP-17<br>3B-AP-11 | 2-AP-11<br>2-AP-13<br>2-AP-17<br>3A-AP-17<br>3B-AP-11 | 2-AP-11<br>2-AP-13<br>2-AP-17<br>3A-AP-17<br>3B-AP-11 | 2-AP-11<br>2-AP-13<br>2-AP-17<br>3A-AP-17<br>3B-AP-11 | 2-AP-11<br>2-AP-13<br>2-AP-17<br>3A-AP-17<br>3B-AP-11 | 2-AP-11<br>2-AP-13<br>2-AP-17<br>3A-AP-17<br>3B-AP-11 | 2-AP-11<br>2-AP-13<br>2-AP-17<br>3A-AP-17<br>3B-AP-11 |
| **ISTE Standards** | 1.c, 1.d, 4.d, 5.c, | 1.c, 1.d, 4.d, 5.c, | 1.c, 1.d, 4.d, 5.c, | 1.c, 1.d, 4.d, 5.c, | 1.c, 1.d, 4.d, 5.c, | 1.c, 1.d, 4.d, 5.c, | 1.c, 1.d, 4.d, 5.c, | 1.c, 1.d, 4.d, 5.c, |

| | 5.d | 5.d | 5.d | 5.d | 5.d | 5.d | 5.d | 5.d |
|---|---|---|---|---|---|---|---|---|
| **UK National Curriculum** | Key Stage 3 & 4 Computing* | Key Stage 3 & 4 Computing* | Key Stage 3 & 4 Computing* | Key Stage 3 & 4 Computing* | Key Stage 3 & 4 Computing* | Key Stage 3 & 4 Computing* | Key Stage 3 & 4 Computing* | Key Stage 3 & 4 Computing* |
| **Sample Application of Skills** | Explore a Cart-Pole game to understand how AI learns tasks. | Discuss the tradeoff between exploration and exploration in learning to play games. | Apply AI concepts to create a Flappy-Bird style game. | Review Unit 4 concepts. | Take a quiz on Unit 4 concepts. | Analyze the impact of AI in different fields. | Conduct an independent research project on an AI topic. | Interact with a Chatbot. |

*See individual lesson guides for details on UK Computer standards

# Artificial Intelligence 401

## Scope and Sequence

### Grades 7+

Each lesson takes about 45-60 minutes to complete.

Artificial Intelligence 401 is a JavaScript and p5.js based course that's perfect for experienced students looking to implement AI in their programs. They learn how to use MediaPipe, an AI solution from Google. Additionally, the activities and projects engage students in developing computational thinking skills, as listed below from the CSTA Computer Science standards and UK Computing standards. The Common Core State Standards for Mathematics and English Language Arts that students develop are also listed here.

| | Lesson 5.4 - Bias | Lesson 5.5 - Review | Lesson 5.6 - Quiz | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Key Skills and Concepts** | • Bias in AI<br>• Trained models | • Benefits and harmful effects of AI<br>• Bias<br>• Impacts of AI<br>• Ethics | • Benefits and harmful effects of AI<br>• Bias<br>• Impacts of AI<br>• Ethics | | | | | |
| **CCSS-Math Standards** | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3<br>MP.1<br>MP.2<br>MP.4 | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3<br>MP.1<br>MP.2<br>MP.4 | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3<br>MP.1<br>MP.2<br>MP.4 | | | | | |
| **CCSS-ELA Standards** | SL.8.1<br>RI.9-10.3<br>RI.9-10.6<br>RI.11-12.3<br>RI.11-12.6<br>L.9-10.3<br>L.9-10.6<br>L.11-12.3<br>L.11-12.6 | SL.8.1<br>RI.9-10.3<br>RI.9-10.6<br>RI.11-12.3<br>RI.11-12.6<br>L.9-10.3<br>L.9-10.6<br>L.11-12.3<br>L.11-12.6 | SL.8.1<br>RI.9-10.3<br>RI.9-10.6<br>RI.11-12.3<br>RI.11-12.6<br>L.9-10.3<br>L.9-10.6<br>L.11-12.3<br>L.11-12.6 | | | | | |
| **CSTA Computer Science Standards** | 2-AP-11<br>2-AP-13<br>2-AP-17<br>3A-AP-17<br>3B-AP-11 | 2-AP-11<br>2-AP-13<br>2-AP-17<br>3A-AP-17<br>3B-AP-11 | 2-AP-11<br>2-AP-13<br>2-AP-17<br>3A-AP-17<br>3B-AP-11 | | | | | |
| **ISTE Standards** | 1.c, 1.d, 4.d, 5.c, 5.d | 1.c, 1.d, 4.d, 5.c, 5.d | 1.c, 1.d, 4.d, 5.c, 5.d | | | | | |
| **UK National** | Key Stage 3 & 4 | Key Stage 3 & 4 | Key Stage 3 & 4 | | | | | |

| Curriculum | Computing* | Computing* | Computing* | | | | | |
|---|---|---|---|---|---|---|---|---|
| Sample Application of Skills | Practice adding new classes to an AI model. | Review Unit 5 concepts. | Take a quiz on Unit 5 concepts. | | | | | |

*See individual lesson guides for details on UK Computer standards

# Python 201
## Scope and Sequence

# Grades 8+
Each lesson takes about 45-60 minutes to complete.

Python 201 is a course for students in grade 8 or above who are already familiar with the basics of programming. As a complement to Python 101, which approaches Python as a tool to build games and solve visual puzzles, Python 201 takes a more traditional computer science approach. The stories, games, puzzles, and projects engage students in developing computational thinking skills in Python, as listed below from the CSTA Level 3 Computer Science standards. The Common Core State Standards for Mathematics and English Language Arts that students develop are also listed here.

| | Lesson 1 - Welcome to Python | Lesson 2 - Variables and I/O | Lesson 3 - Data Types | Lesson 4 - Math Operators | Lesson 5 - Boolean Logic | Lesson 6 - Turtle Graphics | Lesson 7 - Branching | Lesson 8 - While Loops |
|---|---|---|---|---|---|---|---|---|
| **Key Skills and Concepts** | • Use Python instructions to display different values<br>• Document code with comments | • Use variables to store information<br>• Reassign values to variables | • Define and use strings, integers, floats, and booleans<br>• Convert strings to integers<br>• Convert integers to strings | • Use the exponent and modulus operators<br>• Increment and decrement values stored in variables<br>• convert integers to floats | • Write and apply logic/boolean expressions<br>• Use boolean operators to compare values and expressions | • Import and use multiple programming libraries<br>• Use the turtle library like a pen to draw different shapes and images | • Create and use conditional statements<br>• Write programs that react differently based on user input | • Recognize patterns<br>• Use "while" loops with conditional statements<br>• Use variables to control iteration and accumulate values |
| **CCSS-Math Standards** | HSN.Q.A.2<br>HSN.Q.A.3<br>MP.1<br>MP.2 | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3<br>MP.1<br>MP.2<br>MP.4 | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3<br>MP.1<br>MP.2<br>MP.4 | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3<br>MP.1<br>MP.2<br>MP.4 | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3<br>MP.1<br>MP.2<br>MP.4 | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3<br>HSA.CED.A.1<br>HSA.CED.A.3<br>MP.1, MP.2, MP.4 | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3<br>MP.1<br>MP.2<br>MP.4 | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3<br>MP.1<br>MP.2<br>MP.4 |
| **CCSS-ELA Standards** | SL.8.1<br>RI.9-10.3<br>RI.9-10.6<br>RI.11-12.3<br>RI.11-12.6<br>L.9-10.3<br>L.9-10.6<br>L.11-12.3<br>L.11-12.6 | SL.8.1<br>RI.9-10.3 RI.9-10.6<br>RI.11-12.3<br>RI.11-12.6<br>L.9-10.3<br>L.9-10.6<br>L.11-12.3<br>L.11-12.6 | SL.8.1<br>RI.9-10.3<br>RI.9-10.6<br>RI.11-12.3<br>RI.11-12.6<br>L.9-10.3<br>L.9-10.6<br>L.11-12.3<br>L.11-12.6 | SL.8.1<br>RI.9-10.3<br>RI.9-10.6<br>RI.11-12.3<br>RI.11-12.6<br>L.9-10.3<br>L.9-10.6<br>L.11-12.3<br>L.11-12.6 | SL.8.1<br>RI.9-10.3<br>RI.9-10.6<br>RI.11-12.3<br>RI.11-12.6<br>L.9-10.3<br>L.9-10.6<br>L.11-12.3<br>L.11-12.6 | SL.8.1<br>RI.9-10.3<br>RI.9-10.6<br>RI.11-12.3<br>RI.11-12.6<br>L.9-10.3<br>L.9-10.6<br>L.11-12.3<br>L.11-12.6 | SL.8.1<br>RI.9-10.3<br>RI.9-10.6<br>RI.11-12.3<br>RI.11-12.6<br>L.9-10.3<br>L.9-10.6<br>L.11-12.3<br>L.11-12.6 | SL.8.1<br>RI.9-10.3<br>RI.9-10.6<br>RI.11-12.3<br>RI.11-12.6<br>L.9-10.3<br>L.9-10.6<br>L.11-12.3<br>L.11-12.6 |
| **CSTA Computer Science** | 2-AP-13<br>2-AP-16<br>2-AP-17<br>2-AP-19 | 2-AP-11<br>2-AP-13<br>2-AP-16<br>2-AP-17 | 2-AP-11<br>2-AP-13<br>2-AP-17<br>3A-AP-17 | 2-AP-11<br>2-AP-13<br>2-AP-17<br>3A-AP-17 | 2-AP-11<br>2-AP-13<br>2-AP-17 | 2-AP-11<br>2-AP-13<br>2-AP-16<br>2-AP-17 | 2-AP-11<br>2-AP-13<br>2-AP-16<br>2-AP-17 | 2-AP-11<br>2-AP-12<br>2-AP-13<br>2-AP-15 |

| Standards | 3A-AP-17<br>3B-AP-11 | 3A-AP-17<br>3A-AP-21<br>3B-AP-11 | 3A-AP-21<br>3B-AP-11 | 3B-AP-11<br>3A-IC-26 | 3A-AP-17<br>3B-AP-11 | 3A-AP-17<br>3B-AP-11<br>3B-AP-12 | 3A-AP-17<br>3B-AP-11<br>3B-AP-12 | 2-AP-17<br>3A-AP-17<br>3B-AP-11<br>3B-AP-12 |
|---|---|---|---|---|---|---|---|---|
| ISTE Standards | 1.c, 1.d, 4.d, 5.c, 5.d | 1.c, 1.d, 4.d, 5.c, 5.d, 6.b | 1.c, 1.d, 4.d, 5.c, 5.d | 1.c, 1.d, 4.d, 5.c, 5.d | 1.c, 1.d, 4.d, 5.c, 5.d | 1.c, 1.d, 4.d, 5.c, 5.d, 6.b | 1.c, 1.d, 4.d, 5.c, 5.d, 6.b | 1.c, 1.d, 4.d, 5.c, 5.d |
| UK National Curriculum | Coming Soon | Coming Soon | Coming Soon | Coming Soon | Coming Soon | Coming Soon | Coming Soon | Coming Soon |
| Sample Application of Skills | Fix syntax errors based on feedback from error messages. | Initialize a variable | Identify the difference between the four data types | Simulate real word mathematical equations with Python | Predict the results of comparisons using math | Change the color of the Turtle object | Apply branching concepts to real world problems | Control a program's flow by using break and continue statements |

# Python 201

## Scope and Sequence

### Grades 8+

Each lesson takes about 45-60 minutes to complete.

Python 201 is a course for students in grade 8 or above who are already familiar with the basics of programming. As a complement to Python 101, which approaches Python as a tool to build games and solve visual puzzles, Python 201 takes a more traditional computer science approach. The stories, games, puzzles, and projects engage students in developing computational thinking skills in Python, as listed below from the CSTA Level 3 Computer Science standards. The Common Core State Standards for Mathematics and English Language Arts that students develop are also listed here.

| | Lesson 9 - Strings | Lesson 10 - Lists | Lesson 11 - For Loops | Lesson 12 - Functions | Lesson 13 - Dictionaries | Lesson 14 - Classes and Objects | Lesson 15 - Recursion | |
|---|---|---|---|---|---|---|---|---|
| **Key Skills and Concepts** | • Apply specific functions to strings to change the strings in certain ways<br>• Access specific parts of strings using indexes | • Create lists in Python<br>• Use indexes to find particular elements or to create new lists<br>• Add, remove, and count values in a list | • Use "for" loops to iterate through lists and manipulate each element<br>• Iterate over a range of numbers<br>• Use nested loops for drawing shapes and images | • Define and call functions<br>• Identify when to use different parameters<br>• Use global variables to store values that are changed in functions | • Create and alter values stored in dictionaries<br>• Identify differences between a key and a value in a key-value pair<br>• Iterate through dictionaries | • Define classes with functions and variables<br>• Use classes to create objects<br>• Change the values in attributes of objects | • Define recursion<br>• Write out what the base case and recursive case are for different mathematical problems | |
| **CCSS-Math Standards** | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3<br>MP.1<br>MP.2<br>MP.4 | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3<br>MP.1<br>MP.2<br>MP.4 | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3<br>MP.1<br>MP.2<br>MP.4 | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3<br>MP.1<br>MP.2<br>MP.4 | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3<br>MP.1<br>MP.2<br>MP.4 | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3<br>MP.1<br>MP.2<br>MP.4 | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3<br>MP.1<br>MP.2<br>MP.4 | |
| **CCSS-ELA Standards** | SL.8.1<br>RI.9-10.3<br>RI.9-10.6<br>RI.11-12.3<br>RI.11-12.6<br>L.9-10.3<br>L.9-10.6<br>L.11-12.3<br>L.11-12.6 | SL.8.1<br>RI.9-10.3<br>RI.9-10.6<br>RI.11-12.3<br>RI.11-12.6<br>L.9-10.3<br>L.9-10.6<br>L.11-12.3<br>L.11-12.6 | SL.8.1<br>RI.9-10.3<br>RI.9-10.6<br>RI.11-12.3<br>RI.11-12.6<br>L.9-10.3<br>L.9-10.6<br>L.11-12.3<br>L.11-12.6 | SL.8.1<br>RI.9-10.3<br>RI.9-10.6<br>RI.11-12.3<br>RI.11-12.6<br>L.9-10.3<br>L.9-10.6<br>L.11-12.3<br>L.11-12.6 | SL.8.1<br>RI.9-10.3<br>RI.9-10.6<br>RI.11-12.3<br>RI.11-12.6<br>L.9-10.3<br>L.9-10.6<br>L.11-12.3<br>L.11-12.6 | SL.8.1<br>RI.9-10.3<br>RI.9-10.6<br>RI.11-12.3<br>RI.11-12.6<br>L.9-10.3<br>L.9-10.6<br>L.11-12.3<br>L.11-12.6 | SL.8.1<br>RI.9-10.3<br>RI.9-10.6<br>RI.11-12.3<br>RI.11-12.6<br>L.9-10.3<br>L.9-10.6<br>L.11-12.3<br>L.11-12.6 | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **CSTA Computer Science Standards** | 2-AP-11<br>2-AP-12<br>2-AP-13<br>2-AP-15<br>2-AP-17<br>3A-AP-17<br>3B-AP-11<br>3B-AP-12 | 2-AP-11<br>2-AP-12<br>2-AP-13<br>2-AP-15<br>2-AP-17<br>3A-AP-17<br>3B-AP-11<br>3B-AP-12 | 2-AP-11<br>2-AP-12<br>2-AP-13<br>2-AP-15<br>2-AP-17<br>3A-AP-14<br>3A-AP-17<br>3B-AP-11<br>3B-AP-12 | 2-AP-11<br>2-AP-12<br>2-AP-13<br>2-AP-14<br>2-AP-15<br>2-AP-17<br>3A-AP-17<br>3B-AP-11<br>3B-AP-12 | 2-AP-11<br>2-AP-12<br>2-AP-13<br>2-AP-14<br>2-AP-15<br>2-AP-17<br>3A-AP-14<br>3A-AP-17<br>3B-AP-11<br>3B-AP-12 | 2-AP-11<br>2-AP-12<br>2-AP-13<br>2-AP-14<br>2-AP-15<br>2-AP-17<br>3A-AP-17<br>3B-AP-11<br>3B-AP-12 | 2-AP-11<br>2-AP-12<br>2-AP-13<br>2-AP-14<br>2-AP-15<br>2-AP-16<br>2-AP-17<br>3A-AP-17<br>3B-AP-11<br>3B-AP-12 | |
| **ISTE Standards** | 1.c, 1.d, 4.d, 5.c, 5.d | 1.c, 1.d, 4.d, 5.c, 5.d | 1.c, 1.d, 4.d, 5.c, 5.d | 1.c, 1.d, 4.d, 5.c, 5.d | 1.c, 1.d, 4.d, 5.c, 5.d | 1.c, 1.d, 4.d, 5.c, 5.d | 1.c, 1.d, 4.d, 5.c, 5.d | |
| **UK National Curriculum** | Coming Soon | Coming Soon | Coming Soon | Coming Soon | Coming Soon | Coming Soon | Coming Soon | |
| **Sample Application of Skills** | Find and replace a specific substring | Sort lists alphabetically | Create programs to count calendar days depending on the year | Create multiple functions in one program | Add and remove elements from dictionaries | Create classes that take multiple parameters | Write programs that solve problems recursively, without loops | |

# Programming 400

**Scope and Sequence**

# Grades 9+

Each lesson takes about 45-60 minutes to complete.

Introduce your high school class to coding. In this course, students get started with Python, then move on to JavaScript and HTML/CSS. Students learn on their own as they progress through interactive tutorials and coding puzzles, following along to build their own projects.

| | Lesson 1 - Intro to Python | Lesson 2 - Spiral Shapes | Lesson 3 - Intro to JavaScript | Lesson 4 - Emoji Maker | Lesson 5 - Photo Album | |
|---|---|---|---|---|---|---|
| **Key Skills and Concepts** | • Python syntax<br>• for loops<br>• Comments | • Turtle graphics<br>• Functions | • JavaScript syntax<br>• for loops | • Functions<br>• p5.JS | • HTML elements<br>• CSS styles<br>• Web pages<br>• Images | |
| **CSTA Computer Science Standards** | 3A-AP-17<br>3A-AP-21<br>3A-AP-22<br>3B-AP-10<br>3B-AP-11 | 3A-AP-17<br>3A-AP-21<br>3A-AP-22<br>3B-AP-10<br>3B-AP-11<br>3B-AP-15<br>3B-AP-22 | 3A-AP-17<br>3A-AP-21<br>3A-AP-22<br>3B-AP-10<br>3B-AP-11 | 3A-AP-17<br>3A-AP-21<br>3A-AP-22<br>3B-AP-10<br>3B-AP-11<br>3B-AP-15<br>3B-AP-22 | 3A-AP-17<br>3A-AP-21<br>3A-AP-22<br>3B-AP-10<br>3B-AP-11<br>3B-AP-15<br>3B-AP-22 | |
| **ISTE Standards** | 1.1.c<br>1.1.d<br>1.4.d<br>1.5.c | 1.1.c<br>1.1.d<br>1.4.d<br>1.5.c<br>1.6.b | 1.1.c<br>1.1.d<br>1.4.d<br>1.5.c | 1.1.c<br>1.1.d<br>1.4.d<br>1.5.c<br>1.6.b | 1.1.c<br>1.1.d<br>1.4.d<br>1.5.c<br>1.6.b | |
| **UK National Curriculum** | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | |
| **Sample Application of Skills** | Use Python commands to move a character and solve coding puzzles. | Use Python and turtle graphics to create spiral shapes. | Use JavaScript commands to and for loops to complete coding puzzles. | Create a project that uses simple shapes to draw an emoji. | Use HTML and CSS to create a web page that showcases pictures. | |

# Java 101
## Scope and Sequence

**Grades 9+**

Each unit includes a suggested pacing guide.

Students will explore programming by problem-solving with Java, a language that's widely used in introductory college computer science courses and the high-tech industry. This AP-aligned preparatory course is for all academically prepared high-school readers, with hands-on exercises exploring design strategies and methodologies, data structures, and the ethical and social implications of computing.

| | Lesson 1.1 - Images | Lesson 1.2 - Introduction | Lesson 1.3 - Visualizing Algorithms | Lesson 1.4 - Printing to the Console | Lesson 1.5 - Strings and Input | 1.6 - Algorithms Review |
|---|---|---|---|---|---|---|
| **Key Skills and Concepts** | • Technology's role in the world | • Java syntax<br>• Algorithms | • Java syntax<br>• Algorithms | • Java syntax<br>• Algorithms<br>• Print commands | • Java syntax<br>• Algorithms<br>• Strings<br>• Input<br>• Variables | • Java syntax<br>• Algorithms<br>• Commands |
| **CSTA Computer Science Standards** | 3B-IC-27 | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11 | 3B-AP-10<br>3B-AP-11<br>3A-AP-23 | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11 | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11 | 3A-IC-26 |
| **ISTE Standards** | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c |
| **UK National Curriculum** | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* |
| **Sample Application of Skills** | Learn about Java and Computer Science. | Solve 3 coding puzzles. | Visualize a program using a flowchart. | Display text in the console. | Create input prompts, save the inputs to a variable, and print the variable's value. | Reinforce what was learned so far about algorithms. |

# Java 101
**Grades 9+**

## Scope and Sequence
Each unit includes a suggested pacing guide.

Students will explore programming by problem-solving with Java, a language that's widely used in introductory college computer science courses and the high-tech industry. This AP-aligned preparatory course is for all academically prepared high-school readers, with hands-on exercises exploring design strategies and methodologies, data structures, and the ethical and social implications of computing.

| | Lesson 1.7 - Algorithms Quiz | Lesson 2.1 - Abstraction | Lesson 2.2 - More on Methods | Lesson 2.3 - Composition | Lesson 2.4 - ASCII Art Methods | 2.5 - Methods Review |
|---|---|---|---|---|---|---|
| **Key Skills and Concepts** | • Algorithms<br>• Commands | • Abstraction<br>• Methods | • Methods<br>• Algorithms<br>• Camel case | • Methods<br>• Composition<br>• Debugging | • Refinement<br>• Algorithms | • Refactoring<br>• Abstraction<br>• Methods<br>• Composition |
| **CSTA Computer Science Standards** | 3A-IC-26 | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11<br>3A-CS-01 | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11 | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11 | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11<br>3B-AP-14<br>3B-AP-22 | 3A-IC-26 |
| **ISTE Standards** | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c |
| **UK National Curriculum** | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* |
| **Sample Application of Skills** | Take a quiz on Unit 1 concepts. | Create your own methods. | Solve four coding puzzles by creating your own method. | Improve a program by simplifying the code. | Create a text image. | Review Unit 2 concepts. |

# Java 101

## Scope and Sequence

**Grades 9+**

Each unit includes a suggested pacing guide.

Students will explore programming by problem-solving with Java, a language that's widely used in introductory college computer science courses and the high-tech industry. This AP-aligned preparatory course is for all academically prepared high-school readers, with hands-on exercises exploring design strategies and methodologies, data structures, and the ethical and social implications of computing.

| | Lesson 2.6 - Methods Quiz | Lesson 3.1 - Encoding Information | Lesson 3.2 - Binary Numbers | Lesson 3.3 - Variables | Lesson 3.4 - How to Use Variables | Lesson 3.5 - Types |
|---|---|---|---|---|---|---|
| **Key Skills and Concepts** | • Refactoring<br>• Abstraction<br>• Methods<br>• Composition | • Data encoding | • Binary number system<br>• Bit and bit sequences | • Java syntax<br>• Variables<br>• Operators | • Assigning variables<br>• Variable types | • Assigning variables<br>• Variable types |
| **CSTA Computer Science Standards** | 3A-CS-01<br>3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11 | 3A-IC-26<br>3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11 | 3A-IC-26 | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11 | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11 | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11<br>3A-AP-21 |
| **ISTE Standards** | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c |
| **UK National Curriculum** | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* |
| **Sample Application of Skills** | Take a quiz on Unit 2 concepts. | View an example of encoded data in your daily lives. | Convert from binary to decimal and vice versa. | Assign your own variables and declare the values as integers or strings. | Assign and reassign variables. | Create and declare the values of int, double, float, char, and boolean types. |

# Java 101

## Grades 9+

### Scope and Sequence

Each unit includes a suggested pacing guide.

Students will explore programming by problem-solving with Java, a language that's widely used in introductory college computer science courses and the high-tech industry. This AP-aligned preparatory course is for all academically prepared high-school readers, with hands-on exercises exploring design strategies and methodologies, data structures, and the ethical and social implications of computing.

| | Lesson 3.6 - Variables Review | Lesson 3.7 - Variables Quiz | Lesson 4.1 - What's an Expression? | Lesson 4.2 - Variable Initialization in Java | Lesson 4.3 - Working with Numbers | Lesson 4.4 - Expressions, Operators, and Statements Review |
|---|---|---|---|---|---|---|
| **Key Skills and Concepts** | • Encoding<br>• Binary numbers<br>• Variables<br>• Operators<br>• Literals<br>• Naming conventions | • Encoding<br>• Binary numbers<br>• Variables<br>• Operators<br>• Literals<br>• Naming conventions | • Expressions<br>• Variable types<br>• Java syntax | • Variable initialization<br>• Operators | • Variable types<br>• Floating numbers, integers, and doubles | • Expressions<br>• Operators<br>• Java Syntax<br>• Assignment operators |
| **CSTA Computer Science Standards** | 3A-IC-26 | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11 | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11 | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11 | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11 | 3B-AP-23 |
| **ISTE Standards** | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c |
| **UK National Curriculum** | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* |
| **Sample Application of Skills** | Review Unit 3 concepts. | Take a quiz on Unit 3 concepts. | Create your own expression. | Practice using operators. | Experiment with type limits and wrapping values. | Review Unit 4 concepts. |

# Java 101

**Grades 9+**

## Scope and Sequence

Each unit includes a suggested pacing guide.

Students will explore programming by problem-solving with Java, a language that's widely used in introductory college computer science courses and the high-tech industry. This AP-aligned preparatory course is for all academically prepared high-school readers, with hands-on exercises exploring design strategies and methodologies, data structures, and the ethical and social implications of computing.

| | Lesson 4.5 - Expressions, Operators, and Statements Quiz | Lesson 5.1 - Variables and Memory | Lesson 5.2 - Properties and Behaviors | Lesson 5.3 - Classes | Lesson 5.4 - Instance Variables and Methods | Lesson 5.5 - Constructors |
|---|---|---|---|---|---|---|
| **Key Skills and Concepts** | • Expressions<br>• Operators<br>• Java Syntax<br>• Assignment operators | • Variables<br>• Value types<br>• Reference types<br>• Recursion | • Properties/attributes<br>• Behaviors<br>• Objects<br>• Object-oriented programming | • Custom types in Java<br>• Classes | • Classes<br>• Strings<br>• Instance variables<br>• Instance methods | • Constructors in Java |
| **CSTA Computer Science Standards** | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11 | 3A-AP-22<br>3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11<br>3B-AP-13<br>3B-AP-15 | 3A-IC-26 | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11 | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11<br>3B-AP-21 | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11 |
| **ISTE Standards** | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c |
| **UK National Curriculum** | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* |
| **Sample Application of Skills** | Take a quiz on Unit 4 concepts. | Trace and execute recursive methods. | Explore the fundamental ideas behind objects and object-oriented programming. | Practice making custom types (classes) in Java. | Practice assigning values to the properties of the instances. | Explore constructors and discuss why they're useful. |

# Java 101

## Grades 9+

### Scope and Sequence

Students will explore programming by problem-solving with Java, a language that's widely used in introductory college computer science courses and the high-tech industry. This AP-aligned preparatory course is for all academically prepared high-school readers, with hands-on exercises exploring design strategies and methodologies, data structures, and the ethical and social implications of computing.

| | Lesson 5.6 - Drawing Shapes | Lesson 5.7 - The Math Class | Lesson 5.8 - Review | Lesson 5.9 - Quiz | Lesson 6.1 - Introduction to Strings |
|---|---|---|---|---|---|
| **Key Skills and Concepts** | • Methods<br>• Java syntax | • Math class<br>• random()<br>• abs()<br>• pow() | • Classes<br>• Constructors<br>• Primitive types | • Classes<br>• Constructors<br>• Primitive types | • Strings<br>• Char values<br>• Concatenation |
| **CSTA Computer Science Standards** | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11 | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11 | 3A-IC-26 | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11 | 3A-AP-21<br>3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11<br>3B-AP-12 |
| **ISTE Standards** | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c |
| **UK National Curriculum** | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* |
| **Sample Application of Skills** | Use methods to create shapes on the canvas. | Explore the Java Math class. | Review Unit 5 concepts. | Take a quiz on Unit 5 concepts. | Create programs that use strings. |

# Java 101

**Grades 9+**

## Scope and Sequence

Each unit includes a suggested pacing guide.

Students will explore programming by problem-solving with Java, a language that's widely used in introductory college computer science courses and the high-tech industry. This AP-aligned preparatory course is for all academically prepared high-school readers, with hands-on exercises exploring design strategies and methodologies, data structures, and the ethical and social implications of computing.

| | Lesson 6.2 - String Methods | Lesson 6.3 - More String Methods | Lesson 6.4 - Substrings | Lesson 6.5 - Review | Lesson 6.6 - Quiz | Lesson 7.1 - Boolean Expression Basics |
|---|---|---|---|---|---|---|
| **Key Skills and Concepts** | • The String class<br>• Methods<br>• Stack overflow | • compareTo method<br>• Strings<br>• input | • Substring method in Java<br>• Strings | • Strings<br>• Types of data<br>• Operators<br>• String methods<br>• Special characters | • Strings<br>• Types of data<br>• Operators<br>• String methods<br>• Special characters | • Logic<br>• Boolean expressions<br>• Operators |
| **CSTA Computer Science Standards** | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11<br>3B-AP-16 | 3A-AP-21<br>3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11<br>3B-AP-12 | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11 | 3A-IC-26 | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11 | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11 |
| **ISTE Standards** | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c |
| **UK National Curriculum** | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* |
| **Sample Application of Skills** | Explore different ways to call on a string. | Explore more string methods. | Explore the substring method of string and practice using it. | Review Unit 6 concepts. | Take a quiz on Unit 6 concepts. | Explore examples of boolean operators in Java. |

# Java 101

## Scope and Sequence

Grades 9+

Each unit includes a suggested pacing guide.

Students will explore programming by problem-solving with Java, a language that's widely used in introductory college computer science courses and the high-tech industry. This AP-aligned preparatory course is for all academically prepared high-school readers, with hands-on exercises exploring design strategies and methodologies, data structures, and the ethical and social implications of computing.

| | Lesson 7.2 - Logical Operators in Java | Lesson 7.3 - Using Relational and Logical Operators Together | Lesson 7.4 - Conditional Statements | Lesson 7.5 - Else and Else-If Statements | Lesson 7.6 - Conditionals Review | Lesson 7.7 - Conditionals Quiz |
|---|---|---|---|---|---|---|
| **Key Skills and Concepts** | • Branching<br>• Logical operators<br>• Boolean expressions | • Branching<br>• Short-circuiting | • Conditionals<br>• Sequential control flow | • Conditional statements | • Boolean operators<br>• Conditionals<br>• Logical operators | • Boolean operators<br>• Conditionals<br>• Logical operators |
| **CSTA Computer Science Standards** | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11<br>3B-AP-22 | 3B-AP-14 | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11 | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11 | 3A-IC-26 | 3A-AP-16<br>3B-AP-10 |
| **ISTE Standards** | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c |
| **UK National Curriculum** | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* |
| **Sample Application of Skills** | Evaluate boolean expressions. | Practice building complex boolean expressions. | Explore conditional logic. | Explore "else" and "if-else" constructs in Java. | Review Unit 7 concepts. | Take a quiz on Unit 7 concepts. |

# Java 101

## Grades 9+

### Scope and Sequence

Each unit includes a suggested pacing guide.

Students will explore programming by problem-solving with Java, a language that's widely used in introductory college computer science courses and the high-tech industry. This AP-aligned preparatory course is for all academically prepared high-school readers, with hands-on exercises exploring design strategies and methodologies, data structures, and the ethical and social implications of computing.

| | Lesson 8.1 - The While Loop | Lesson 8.2 - Take Control of Loops | Lesson 8.3 - Infinite Loops | Lesson 8.4 - Draw With Loops | Lesson 8.5 - Randomness | Lesson 8.6 - While Loop Puzzles |
|---|---|---|---|---|---|---|
| **Key Skills and Concepts** | • While loops<br>• Java syntax<br>• Iteration | • While loops<br>• Iteration<br>• Return statement<br>• Code tracing | • Infinite loop<br>• Input controlled loops<br>• Sentinel value | • While loops<br>• Algorithmic art | • Math.random()<br>• Randomness | • While loops<br>• Conditional statements |
| **CSTA Computer Science Standards** | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11 | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11 | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11 | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11 | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11 | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11 |
| **ISTE Standards** | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c |
| **UK National Curriculum** | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* |
| **Sample Application of Skills** | Solve coding puzzles that use while loops and methods. | Use break statements to halt loops. | Lean about common errors in programming with loops. | Use while loops to crate algorithmic art. | Create a computerized coin flipper. | Solve coding puzzles that use while loops and conditional statements. |

# Java 101

## Scope and Sequence

### Grades 9+

Each unit includes a suggested pacing guide.

Students will explore programming by problem-solving with Java, a language that's widely used in introductory college computer science courses and the high-tech industry. This AP-aligned preparatory course is for all academically prepared high-school readers, with hands-on exercises exploring design strategies and methodologies, data structures, and the ethical and social implications of computing.

| | Lesson 8.7 - While Loops Review | Lesson 8.8 - While Loops Quiz | Lesson 9.1 - For Loop Basics | Lesson 9.2 - Nested Loops | Lesson 9.3 - Accumulating, Filtering, Mapping | Lesson 9.4 - Loop Challenges |
|---|---|---|---|---|---|---|
| **Key Skills and Concepts** | • Conditional statements<br>• While loops<br>• Random numbers | • Conditional statements<br>• While loops<br>• Random numbers | • For loops<br>• While loops<br>• iteration | • Nested loops<br>• For loops<br>• While loops | • Accumulate, filter, and map data<br>• Software patterns | • Loops<br>• Graphing |
| **CSTA Computer Science Standards** | 3A-IC-26 | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11 | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11<br>3B-AP-22 | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11 | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11 | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11 |
| **ISTE Standards** | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c |
| **UK National Curriculum** | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* |
| **Sample Application of Skills** | Review Unit 8 concepts. | Take a quiz on Unit 8 concepts. | Solve multiple coding puzzles using loops. | Solve coding puzzles using nested loops, for loops, and while loops. | Read about accumulating, filtering, and mapping. | Create images on a graph using loops. |

# Java 101

## Scope and Sequence

**Grades 9+**

Each unit includes a suggested pacing guide.

Students will explore programming by problem-solving with Java, a language that's widely used in introductory college computer science courses and the high-tech industry. This AP-aligned preparatory course is for all academically prepared high-school readers, with hands-on exercises exploring design strategies and methodologies, data structures, and the ethical and social implications of computing.

| | Lesson 9.5 - Common Pitfalls | Lesson 9.6 - Review | Lesson 9.7 - Quiz | Lesson 10.1 - Lab: Algorithms | Lesson 11.1 - Constructors | Lesson 11.2 - this |
|---|---|---|---|---|---|---|
| **Key Skills and Concepts** | • Common loop errors<br>• For loops<br>• While loops<br>• Mapping | • Loops<br>• Accumulating<br>• Filtering<br>• Mapping<br>• Common loop errors<br>• Nesting loops | • Loops<br>• Accumulating<br>• Filtering<br>• Mapping<br>• Common loop errors<br>• Nesting loops | • Debugging<br>• Java syntax<br>• Algorithms<br>• Variables<br>• Strings | • Constructors | • Java keyword 'this'<br>• Behavior |
| **CSTA Computer Science Standards** | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11 | 3A-IC-26 | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11 | 3A-AP-16<br>3A-AP-17<br>3A-AP-21<br>3A-AP-22<br>3B-AP-10<br>3B-AP-11<br>3B-AP-14<br>3B-AP-22 | 3A-IC-26<br>3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11 | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11<br>3B-AP-13 |
| **ISTE Standards** | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c |
| **UK National Curriculum** | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* |
| **Sample Application of Skills** | Fix identified errors and create working "for" and "while" loops. | Review Unit 9 concepts. | Take a quiz on Unit 9 concepts. | Complete challenge questions with algorithms. | Apply coding concepts to make a constructor. | Practice working with a custom Java type (class) called Radio. |

# Java 101

**Grades 9+**

## Scope and Sequence

Each unit includes a suggested pacing guide.

Students will explore programming by problem-solving with Java, a language that's widely used in introductory college computer science courses and the high-tech industry. This AP-aligned preparatory course is for all academically prepared high-school readers, with hands-on exercises exploring design strategies and methodologies, data structures, and the ethical and social implications of computing.

| | Lesson 11.3 - Intangible Objects | Lesson 11.4 - Adding Functionality | Lesson 11.5 - toString | Lesson 11.6 - Separation of Concerns | Lesson 11.7 - Setters and Getters | Lesson 11.8 - Writing Setters and Getters |
|---|---|---|---|---|---|---|
| **Key Skills and Concepts** | • Copy constructor | • Class behaviors | • Instance variables<br>• Instance methods<br>• Strings | • Separation of concerns<br>• Debugging<br>• Variables | • Setters<br>• Getters | • Logic<br>• Setters<br>• Getters |
| **CSTA Computer Science Standards** | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11<br>3B-AP-14 | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11<br>3B-AP-14 | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11<br>3B-AP-12<br>3B-AP-14 | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11<br>3B-AP-14 | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11 | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11 |
| **ISTE Standards** | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c |
| **UK National Curriculum** | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* |
| **Sample Application of Skills** | Practice using Java's rules for custom constructors. | Practice adding more functionality to classes. | Practice assigning values to the properties of the instances. | Practice making variables private. | Practice using getters and setters. | Create a project that uses getters and setters. |

# Java 101

**Grades 9+**

## Scope and Sequence

Each unit includes a suggested pacing guide.

Students will explore programming by problem-solving with Java, a language that's widely used in introductory college computer science courses and the high-tech industry. This AP-aligned preparatory course is for all academically prepared high-school readers, with hands-on exercises exploring design strategies and methodologies, data structures, and the ethical and social implications of computing.

| | Lesson 11.9 - National Parks | Lesson 11.10 - Impact of Computing | Lesson 11.11 - Review | Lesson 11.12 - Quiz | Lesson 12.1 - The Case for Arrays | Lesson 12.2 - Creating Arrays |
|---|---|---|---|---|---|---|
| **Key Skills and Concepts** | • toString()<br>• Strings<br>• Methods | • Open source<br>• Open access<br>• Plagiarism<br>• System reliability<br>• Intellectual property | • State<br>• Behavior<br>• Constructors<br>• 'this'<br>• toString() | • State<br>• Behavior<br>• Constructors<br>• 'this'<br>• toString() | • Strings<br>• Arrays | • Arrays<br>• Data types<br>• For-loops |
| **CSTA Computer Science Standards** | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11 | 3A-IC-24<br>3A-IC-30<br>3B-IC-25<br>3A-NI-07<br>3B-NI-03 | 3A-IC-26 | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11 | 3A-IC-26<br>3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11 | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11<br>3B-AP-22 |
| **ISTE Standards** | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c |
| **UK National Curriculum** | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* |
| **Sample Application of Skills** | Practice creating classes and implementing the toString() method in Java. | Identify ethical and social implications of computing systems. | Review Unit 11 concepts. | Take a quiz on Unit 11 concepts. | Use arrays with integers or Strings in code. | Create arrays using different data types. |

# Java 101

## Scope and Sequence

### Grades 9+

Each unit includes a suggested pacing guide.

Students will explore programming by problem-solving with Java, a language that's widely used in introductory college computer science courses and the high-tech industry. This AP-aligned preparatory course is for all academically prepared high-school readers, with hands-on exercises exploring design strategies and methodologies, data structures, and the ethical and social implications of computing.

| | Lesson 12.3 - For Loops and Arrays | Lesson 12.4 - Processing Arrays | Lesson 12.5 - Taking Flight | Lesson 12.6 - Arrays Review | Lesson 12.7 - Arrays Quiz | Lesson 13.1 - Arrays and Objects |
|---|---|---|---|---|---|---|
| **Key Skills and Concepts** | • Iterate<br>• .length<br>• Arrays<br>• Loops | • Arrays<br>• Instance variables<br>• Instance methods | • Arrays<br>• Java syntax<br>• Methods | • Arrays<br>• Creating arrays<br>• Assessing arrays<br>• Iterating through arrays<br>• Prescribing arrays | • Arrays<br>• Creating arrays<br>• Assessing arrays<br>• Iterating through arrays<br>• Prescribing arrays | • Arrays<br>• Methods |
| **CSTA Computer Science Standards** | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11<br>3B-AP-22 | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11 | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11 | 3A-IC-26<br>3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11 | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11 | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11 |
| **ISTE Standards** | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c |
| **UK National Curriculum** | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* |
| **Sample Application of Skills** | Iterate through arrays using for-loops. | Determine the smallest value within an array. | Process an array to determine the cheapest price. | Review Unit 12 concepts. | Take a quiz on Unit 12 concepts. | Pass arrays to methods. |

# Java 101
## Scope and Sequence

**Grades 9+**

Each unit includes a suggested pacing guide.

Students will explore programming by problem-solving with Java, a language that's widely used in introductory college computer science courses and the high-tech industry. This AP-aligned preparatory course is for all academically prepared high-school readers, with hands-on exercises exploring design strategies and methodologies, data structures, and the ethical and social implications of computing.

| | Lesson 13.2 - Comma Separated Values | Lesson 13.3 - Arrays and Objects 2 | Lesson 13.4 - ToDo List | Lesson 13.5 - Dream Journal | Lesson 13.6 - Global Temperatures | Lesson 13.7 - Graphing Temperatures |
|---|---|---|---|---|---|---|
| **Key Skills and Concepts** | • Parsing<br>• Comma-separated data | • Methods<br>• Arrays<br>• Precondition<br>• Postcondition | • Classes<br>• Java syntax<br>• Arrays | • Classes<br>• Variables<br>• Booleans | • Data<br>• Methods | • Graphing data<br>• Java methods |
| **CSTA Computer Science Standards** | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11 | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11 | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11 | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11 | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11 | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11<br>3A-DA-10<br>3A-DA-11<br>3A-DA-12<br>3B-DA-05<br>3B-DA-06<br>3B-DA-07 |
| **ISTE Standards** | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c |
| **UK National Curriculum** | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* |
| **Sample Application of Skills** | Begin creating a program that analyzes the top YouTube videos of the day. | Create and call methods. | Create a To-Do application. | Create a dream journal application. | Create methods to determine high and low values. | Use real world NASA temperature data and Java methods to tell a story. |

# Java 101

## Scope and Sequence

## Grades 9+

Each unit includes a suggested pacing guide.

Students will explore programming by problem-solving with Java, a language that's widely used in introductory college computer science courses and the high-tech industry. This AP-aligned preparatory course is for all academically prepared high-school readers, with hands-on exercises exploring design strategies and methodologies, data structures, and the ethical and social implications of computing.

| | Lesson 13.8 - Regional Temperatures | Lesson 13.9 - ForEach Loop | Lesson 13.10 - Arrays and Objects Review | Lesson 13.11 - Arrays and Objects Quiz | Lesson 14.1 - Lab: Array Challenges | Lesson 15.1 - Introduction to Inheritance |
|---|---|---|---|---|---|---|
| **Key Skills and Concepts** | • Graphing data<br>• Java methods | • For-each loop<br>• Iteration<br>• Arrays | • Arrays<br>• Creating arrays<br>• Assessing array elements | • Arrays<br>• Creating arrays<br>• Assessing array elements | • Arrays<br>• Methods<br>• Conditionals<br>• Variables | • Inheritance<br>• Subclass<br>• Superclass |
| **CSTA Computer Science Standards** | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11<br>3A-DA-10<br>3A-DA-11<br>3A-DA-12, 3B-DA-05, 3B-DA-06, 3B-DA-07 | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11 | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11 | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11 | 3A-AP-16<br>3A-AP-17<br>3A-AP-21<br>3A-AP-22<br>3B-AP-10<br>3B-AP-11<br>3B-AP-14<br>3B-AP-22 | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11 |
| **ISTE Standards** | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c | 1.1.c, 1.1.d, 1.4.d, 1.5.c |
| **UK National Curriculum** | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* |
| **Sample Application of Skills** | Create a new analysis on three data sets using previously created methods. | Write a method that uses a for-each loop. | Review Unit 13 concepts. | Take a quiz on Unit 13 concepts. | Complete coding challenge questions with arrays. | Explore ways of organizing parent-child relationships (inheritance) programmatically. |

# Java 101

## Scope and Sequence

**Grades 9+**

Each unit includes a suggested pacing guide.

Students will explore programming by problem-solving with Java, a language that's widely used in introductory college computer science courses and the high-tech industry. This AP-aligned preparatory course is for all academically prepared high-school readers, with hands-on exercises exploring design strategies and methodologies, data structures, and the ethical and social implications of computing.

| | Lesson 15.2 - Extends | Lesson 15.3 - Equals | Lesson 15.4 - Super | Lesson 15.5 - Polymorphism | Lesson 15.6 - Inheritance Review | Lesson 15.7 - Inheritance Quiz |
|---|---|---|---|---|---|---|
| **Key Skills and Concepts** | • 'extends' keyword | • Equals method<br>• Equality operator<br>• Unique objects | • Strings<br>• 'super' keyword | • Polymorphism<br>• User interface | • Inheritance<br>• Output<br>• Strings<br>• Classes | • Inheritance<br>• Output<br>• Strings<br>• Classes |
| **CSTA Computer Science Standards** | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11<br>3B-AP-14 | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11 | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11<br>3B-AP-14<br>3B-AP-22 | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11<br>3B-AP-14 | 3A-IC-26 | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11<br>3B-AP-14 |
| **ISTE Standards** | 1.1.c, 1.1.d, 1.4.d, 1.5.c, 1.5.d | 1.1.c, 1.1.d, 1.4.d, 1.5.c, 1.5.d | 1.1.c, 1.1.d, 1.4.d, 1.5.c, 1.5.d | 1.1.c, 1.1.d, 1.4.d, 1.5.c, 1.5.d | 1.1.c, 1.1.d, 1.4.d, 1.5.c, 1.5.d | 1.1.c, 1.1.d, 1.4.d, 1.5.c, 1.5.d |
| **UK National Curriculum** | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* |
| **Sample Application of Skills** | Continue to explore inheritance in Java. | Explore different ways of comparing objects. | Practice overriding toString. | Build a Java application with user interface elements. | Review Unit 15 concepts. | Take a quiz on Unit 15 concepts. |

# Java 101

## Scope and Sequence

**Grades 9+**

Each unit includes a suggested pacing guide.

Students will explore programming by problem-solving with Java, a language that's widely used in introductory college computer science courses and the high-tech industry. This AP-aligned preparatory course is for all academically prepared high-school readers, with hands-on exercises exploring design strategies and methodologies, data structures, and the ethical and social implications of computing.

| | Lesson 16.1 - Capstone Project, Day One | Lesson 16.2 - Day Two: Find Bugs | Lesson 16.3 - Day Three: Minimum Viable Product | Lesson 16.4 - Day Four: Polish, Test, and Document | | |
|---|---|---|---|---|---|---|
| **Key Skills and Concepts** | • Java syntax<br>• Debugging | • Functions<br>• Variables<br>• Java syntax<br>• Debugging | • Algorithms<br>• Comments<br>• Debugging | • Algorithms<br>• Comments<br>• Debugging | | |
| **CSTA Computer Science Standards** | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11 | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11<br>3B-AP-14<br>3B-AP-15<br>3B-AP-16, 3B-AP-21 | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11, 3B-AP-14,<br>3B-AP-15, 3B-AP-16,<br>3B-AP-21, 3B-AP-22,<br>3B-AP-23 | 3A-AP-16<br>3A-AP-17<br>3B-AP-10<br>3B-AP-11<br>3B-AP-14, 3B-AP-15<br>3B-AP-16, 3B-AP-21<br>3B-AP-22 , 3B-AP-23 | | |
| **ISTE Standards** | 1.1.c, 1.1.d, 1.4.d, 1.5.c, 1.5.d | 1.1.c, 1.1.d, 1.4.d, 1.5.c, 1.5.d | 1.1.c, 1.1.d, 1.4.d, 1.5.c, 1.5.d | 1.1.c, 1.1.d, 1.4.d, 1.5.c, 1.5.d | | |
| **UK National Curriculum** | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | | |
| **Sample Application of Skills** | Decide what kind of program you'd like to write. | Design a program of your choosing. | Design a program then comment out the code. | Describe a program's functionality. | | |

# Data Science 1
## Scope and Sequence

# Grades 9+
Each unit includes a suggested pacing guide.

In this introductory course, students will learn the foundations of data analysis using Python. Additionally, they'll explore science, sports, politics, climate change, and much more while learning to interrogate a data set, just like a data pro, and make their own conclusions. The coding exercises, projects, and quizzes in this course engage students in developing computational thinking skills, as listed below from the CSTA Computer Science standards and UK Computing standards.

| | Lesson 1.1 - Welcome | Lesson 1.2 - What is Data Analysis? | Lesson 1.3 - Python Refresher | Lesson 1.4 - Start your Analysis | Lesson 1.5 - Back to Python | Lesson 1.6 - Review | Lesson 1.7 - Quiz | Lesson 1.8 - Lab: Exploring Real Data |
|---|---|---|---|---|---|---|---|---|
| **Key Skills and Concepts** | • Recognize how data analysis is a diverse field | • Analyze given data | • Review strings, Python syntax, lists, operators, functions, comments, commands, and loops. | • Identify different types of data such as numerical, categorical, discrete, continuous. | • Use pandas. | • Review statistics, critical thinking, bias, inspecting data. | • Demonstrate an understanding of data analysis and data collection. | • Create DataFrames using Python. |
| **CSTA Computer Science Standards** | 3A-IC-24 | 3A-DA-10 3B-DA-05 3A-IC-24 | 3A-AP-13 3A-AP-17 3B-AP-10 3B-AP-11 | 3A-IC-24 | 3A-DA-10 3B-DA-05 3A-AP-13 3B-AP-10 3B-AP-11 | 3A-AP-13 3B-AP-10 3B-AP-11 | 3A-IC-24 3A-AP-17 | 3A-AP-13 3B-AP-10 3B-AP-11 |
| **ISTE Standards** | 1.c, 1.d | 1.c, 1.d 4.d, 5.c | 1.c, 1.d 4.d, 5.c | 1.c, 1.d 3.b, 4.b | 1.c, 1.d 3.b, 5.c, 5.d | 1.c, 1.d 4.d, 5.c, 5.d | 1.c, 1.d 3.b | 1.c, 1.d, 4.d 5.c 5.d |
| **UK National Curriculum** | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* |
| **Sample Application of Skills** | Read an introduction about data and the course. | Answer open-ended questions about provided data. | Practice using Python. | Examine different ways to analyze data. | Practice calculating the mean, median, mode, and range of datasets. | Complete a review activity in Tynker. | Take a Unit 1 quiz. | Practice working with DataFrames and complete a lab. |

*See individual lesson guides for details on UK Computer standards

# Data Science 1
## Scope and Sequence

**Grades 9+**

Each unit includes a suggested pacing guide.

In this introductory course, students will learn the foundations of data analysis using Python. Additionally, they'll explore science, sports, politics, climate change, and much more while learning to interrogate a data set, just like a data pro, and make their own conclusions. The coding exercises, projects, and quizzes in this course engage students in developing computational thinking skills, as listed below from the CSTA Computer Science standards and UK Computing standards.

| | Lesson 2.1 - Bar Graphs and Histogram | Lesson 2.2 - Line Graphs | Lesson 2.3 - Scatter Plots | Lesson 2.4 - Customizing Plots | Lesson 2.5 - Review | Lesson 2.6 - Quiz | Lesson 2.7 - Lab |
|---|---|---|---|---|---|---|---|
| **Key Skills and Concepts** | • Explore bar graphs, histograms, and binning data. | • Explore numerical data vs numerical data within line graphs. | • Generate a linear regression and plot trend lines in Python. | • Use functions and parameters. | • Review graphs and plots. | • Demonstrate an understanding of graphs, trend lines, and plots customizations. | • Use a CSV editor. |
| **CSTA Computer Science Standards** | 3A-DA-10 3A-DA-11 3A-DA-12 3B-DA-05 | 3A-DA-10 3A-DA-11 3A-DA-12 3B-DA-05 | 3A-DA-10 3A-DA-11 3A-DA-12 3B-DA-05 | 3A-DA-10 3A-DA-11 3A-DA-12 3B-DA-05 | 3A-DA-10 3A-DA-11 3A-DA-12 3B-DA-05 | 3A-DA-10 3A-DA-11 3A-DA-12 3B-DA-05 | 3A-DA-10 3A-DA-11 3A-DA-12 3B-DA-05 3A-AP-13 3B-AP-10 |
| **ISTE Standards** | 1.c, 1.d, 3.b, 5.c, 5.d | 1.c, 1.d, 3.b, 5.c, 5.d | 1.c, 1.d, 3.b, 5.c, 5.d | 1.c, 1.d, 3.b, 5.c, 5.d | 1.c, 1.d, 3.b, 5.c, 5.d | 1.c, 1.d, 3.b, 5.c, 5.d | 1.c, 1.d, 3.b, 5.c, 5.d |
| **UK National Curriculum** | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* |
| **Sample Application of Skills** | Create histograms and bar graphs in code. | Analyze a dataset about pea plants. | Determine if there is a relationship between data. | Change colors and text styles of plots. | Complete a review activity in Tynker. | Take a Unit 2 quiz. | Create 3 graphs in code based on data. |

*See individual lesson guides for details on UK Computer standards

# Data Science 1

**Grades 9+**

## Scope and Sequence

Each unit includes a suggested pacing guide.

In this introductory course, students will learn the foundations of data analysis using Python. Additionally, they'll explore science, sports, politics, climate change, and much more while learning to interrogate a data set, just like a data pro, and make their own conclusions. The coding exercises, projects, and quizzes in this course engage students in developing computational thinking skills, as listed below from the CSTA Computer Science standards and UK Computing standards.

| | Lesson 3.1 - Creating DataFrames | Lesson 3.2 - Creating Random Data | Lesson 3.3 - Manipulating Data Sets | Lesson 3.4 - Variation of a Normal Curve | Lesson 3.5 - Review | Lesson 3.6 - Quiz | Lesson 3.7 - Lab: Olympic Medals |
|---|---|---|---|---|---|---|---|
| **Key Skills and Concepts** | • Reinforce DataFrames. | • Define and explore random data. | • Use code to manipulate data. | • Review normal distribution curves. | • Review creating and manipulating DataFrames <br> • Review random data | • Demonstrate an understanding of Unit 3 concepts. | • Analyze a provided dataset. |
| **CSTA Computer Science Standards** | 3A-DA-10 3B-DA-05 <br> 3A-AP-13 <br> 3B-AP-10 <br> 3B-AP-11 | 3A-DA-10 <br> 3B-DA-05 <br> 3A-AP-13 <br> 3B-AP-10 <br> 3B-AP-11 | 3A-DA-10 <br> 3B-DA-05 <br> 3A-AP-13 <br> 3B-AP-10 <br> 3B-AP-11 | 3A-DA-10 <br> 3B-DA-05 3A-AP-13 <br> 3B-AP-10 3B-AP-11 | 3B-AP-10 <br> 3B-AP-11 | 3A-IC-24 <br> 3A-AP-17 | 3A-AP-13 3B-AP-10 <br> 3B-AP-11 |
| **ISTE Standards** | 1.c, 1.d, <br> 3.b, 5.c, 5.d | 1.c, 1.d, <br> 3.b, 5.c, 5.d | 1.c, 1.d, <br> 3.b, 5.c, 5.d | 1.c, 1.d, <br> 3.b, 5.c, 5.d | 1.c, 1.d, <br> 4.d, 5.c, 5.d | 1.c, 1.d, <br> 3.b | 1.c, 1.d, <br> 3.b, 5.c, 5.d |
| **UK National Curriculum** | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* |
| **Sample Application of Skills** | Recognize different ways to create a DataFrame. | Combine random data into a DataFrame. | Create a DIY project. | Practice creating graphs using Python. | Complete a review activity in Tynker. | Take a Unit 3 quiz. | Create 3 graphs using Python. |

*See individual lesson guides for details on UK Computer standards

# Data Science 1

## Grades 9+

## Scope and Sequence

Each unit includes a suggested pacing guide.

In this introductory course, students will learn the foundations of data analysis using Python. Additionally, they'll explore science, sports, politics, climate change, and much more while learning to interrogate a data set, just like a data pro, and make their own conclusions. The coding exercises, projects, and quizzes in this course engage students in developing computational thinking skills, as listed below from the CSTA Computer Science standards and UK Computing standards.

| | Lesson 4.1 - Box Plot | Lesson 4.2 - Area Plots | Lesson 4.3 - Lab: Sea Level Anomaly | Lesson 4.4 - Creating Figures | Lesson 4.5 - Customizing Figures | Lesson 4.6 - Complex Bar Graph |
|---|---|---|---|---|---|---|
| **Key Skills and Concepts** | • Calculate box plot values by hand | • Use the fill between function to create area plots. | • Create an area plot. | • Create complex figures using mock graphs and random numbers. | • Create an advanced figure. | • Compare and contrast grouped and stacked bar graphs. |
| **CSTA Computer Science Standards** | 3A-DA-10 3A-DA-11 3A-DA-12 3B-DA-05 | 3A-DA-10 3A-DA-11 3A-DA-12 3B-DA-05 3B-AP-10 | 3A-DA-10 3A-DA-11 3A-DA-12 3B-DA-05 | 3A-DA-10 3A-DA-11 3A-DA-12 3B-DA-05 | 3A-DA-10 3A-DA-11 3A-DA-12 3B-DA-05 | 3A-DA-10 3A-DA-11 3A-DA-12 3B-DA-05 |
| **ISTE Standards** | 1.c, 1.d, 3.b, 5.c, 5.d | 1.c, 1.d, 3.b, 5.c, 5.d | 1.c, 1.d, 3.b, 5.c, 5.d | 1.c, 1.d, 3.b, 5.c, 5.d | 1.c, 1.d, 3.b, 5.c, 5.d | 1.c, 1.d, 3.b, 5.c, 5.d |
| **UK National Curriculum** | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* |
| **Sample Application of Skills** | Analyze a dataset and create box plots in code. | Create area plots in code. | Use real data from NASA to plot sea levels. | Identify parts of a figure. | Add a DataFrame based panda graph into a figure. | Create a grouped bar graph. |

*See individual lesson guides for details on UK Computer standards

# Data Science 1

## Scope and Sequence

**Grades 9+**

Each unit includes a suggested pacing guide.

In this introductory course, students will learn the foundations of data analysis using Python. Additionally, they'll explore science, sports, politics, climate change, and much more while learning to interrogate a data set, just like a data pro, and make their own conclusions. The coding exercises, projects, and quizzes in this course engage students in developing computational thinking skills, as listed below from the CSTA Computer Science standards and UK Computing standards.

|  | Lesson 4.7 - Pie Charts | Lesson 4.8 - Review | Lesson 4.9 - Quiz |  |  |  |
|---|---|---|---|---|---|---|
| **Key Skills and Concepts** | • Compare and contrast pie charts and donut graphs. | • Review different types of graphs and figure customizations. | • Demonstrate an understanding of Unit 4 concepts. |  |  |  |
| **CSTA Computer Science Standards** | 3A-DA-10<br>3A-DA-11<br>3A-DA-12<br>3B-DA-05 | 3A-DA-10<br>3A-DA-11<br>3A-DA-12<br>3B-DA-05 | 3A-DA-10<br>3A-DA-12<br>3B-DA-05 |  |  |  |
| **ISTE Standards** | 1.c, 1.d,<br>3.b, 5.c, 5.d | 1.c, 1.d,<br>3.b, 5.c, 5.d | 1.c, 1.d,<br>3.b, 5.c, 5.d |  |  |  |
| **UK National Curriculum** | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* |  |  |  |
| **Sample Application of Skills** | Create a pie chart and a donut graph. | Complete a review activity in Tynker. | Take a Unit 4 quiz. |  |  |  |

*See individual lesson guides for details on UK Computer standards

# Data Science 1

## Scope and Sequence

**Grades 9+**

Each unit includes a suggested pacing guide.

In this introductory course, students will learn the foundations of data analysis using Python. Additionally, they'll explore science, sports, politics, climate change, and much more while learning to interrogate a data set, just like a data pro, and make their own conclusions. The coding exercises, projects, and quizzes in this course engage students in developing computational thinking skills, as listed below from the CSTA Computer Science standards and UK Computing standards.

| | Lesson 5.1 - Day 1: Intro to Final Lab | Lesson 5.2 - Day 2: Load Data, Initial Charts | Lesson 5.3 - Charts Complete | Lesson 5.4 - Polish, Customize, Document | Lesson 5.5 - Final Report | | |
|---|---|---|---|---|---|---|---|
| **Key Skills and Concepts** | • Write a research proposal. | • Use code to edit the file. | • Use the text code editor to experiment with datasets. | • Add final touches to charts and graphs. | • Finalize charts. | | |
| **CSTA Computer Science Standards** | 3A-AP-13 3B-AP-10 3B-AP-11 | 3A-AP-13 3B-AP-10 3B-AP-11 | 3A-AP-13 3B-AP-10 3B-AP-11 | 3A-AP-13 3B-AP-10 3B-AP-11 | 3A-AP-13 3B-AP-10 3B-AP-11 | | |
| **ISTE Standards** | 1.c, 1.d, 4.d, 5.c, 5.d | 1.c, 1.d, 4.d, 5.c, 5.d | 1.c, 1.d, 4.d, 5.c, 5.d | 1.c, 1.d, 4.d, 5.c, 5.d | 1.c, 1.d, 4.d, 5.c, 5.d | | |
| **UK National Curriculum** | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | | |
| **Sample Application of Skills** | Choose a research prompt of your choice. | Practice loading and printing data. | Create at least 4 different charts or graphs. | Practice adding comments to code. | Finish the Final Lab project. | | |

*See individual lesson guides for details on UK Computer standards

# Intro to Programming and Art

## Scope and Sequence

**Grades 9+**

Each unit includes a suggested pacing guide.

Students are introduced to art, design, and animation principles using Processing, a programming language designed for the visual arts community. Additionally, students will explore the works of famous masters such as Monet, Raphael, Picasso, and Seurat---then recreate digital art in the same style using JavaScript and p5.js.The coding exercises, hand-on Labs, and quizzes in this course engage students in developing computational thinking skills, as listed below from the CSTA Computer Science standards and UK Computing standards.

|  | Lesson 1.1 - Welcome | Lesson 1.2 - Introduction to JavaScript | Lesson 1.3 - Colors | Lesson 1.4 - Lab: Piet Mondrian | Lesson 1.5 - Ellipses and Circles | Lesson 1.6 - More Basic Shapes |
|---|---|---|---|---|---|---|
| **Key Skills and Concepts** | • Read an introduction to p5.js. | • Explore syntax, functions, and comments. | • Use basic functions to create a canvas and draw shapes. | • Use the p5.js coordinate system. | • Draw an ellipse and circle on the screen. | • Read about radians, arcs, angles, and constants. |
| **CSTA Computer Science Standards** | 3A-IC-24 | 3A-AP-13 3A-AP-17 3A-AP-18 3B-AP-10 3B-AP-11 | 3A-AP-13 3A-AP-17 3A-AP-18 3B-AP-10 3B-AP-11 | 3A-AP-13 3A-AP-17 3A-AP-18 3B-AP-10 3B-AP-11 | 3A-AP-13 3A-AP-17 3A-AP-18 3B-AP-10 3B-AP-11 | 3A-AP-13 3A-AP-17 3A-AP-18 3B-AP-10 3B-AP-11 |
| **ISTE Standards** | 1.c | 1.c 1.d 4.d 5.c 5.d | 1.c 1.d 4.d 5.c 5.d | 1.c 1.d 4.d 5.c 5.d | 1.c 1.d 4.d 5.c 5.d | 1.c 1.d 4.d 5.c 5.d |
| **UK National Curriculum** | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* |
| **Sample Application of Skills** | Explore what lies ahead in the course. | Identify coding errors. | Explore what a pixel is and locate one on a p5 canvas. | Practice using key functions to draw shapes. | Read about command order and why it matters. | Use the line, triangle, quad, and arc commands. |

*See individual lesson guides for details on UK Computer standards

# Intro to Programming and Art

## Scope and Sequence

**Grades 9+**

Each unit includes a suggested pacing guide.

Students are introduced to art, design, and animation principles using Processing, a programming language designed for the visual arts community. Additionally, students will explore the works of famous masters such as Monet, Raphael, Picasso, and Seurat---then recreate digital art in the same style using JavaScript and p5.js.The coding exercises, hand-on Labs, and quizzes in this course engage students in developing computational thinking skills, as listed below from the CSTA Computer Science standards and UK Computing standards.

| | Lesson 1.7 - Lab: Emojis | Lesson 1.8 - Review | Lesson 1.9 - Quiz | | | |
|---|---|---|---|---|---|---|
| **Key Skills and Concepts** | • Practice using the p5 coordinate system. | • Review functions, pixels, coordinate system, and colors. | • Demonstrate an understanding of RGB, canvas coordinates, functions, and RGB values. | | | |
| **CSTA Computer Science Standards** | 3A-AP-13<br>3A-AP-17<br>3A-AP-18<br>3B-AP-10<br>3B-AP-11 | 3A-AP-13<br>3B-AP-11 | 3A-AP-13<br>3A-AP-17<br>3A-AP-18<br>3B-AP-10<br>3B-AP-11 | | | |
| **ISTE Standards** | 1.c<br>1.d<br>4.d<br>5.c<br>5.d | 1.c<br>1.d<br>4.d<br>5.c<br>5.d | 1.c<br>1.d<br>4.d<br>5.c<br>5.d | | | |
| **UK National Curriculum** | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | | | |
| **Sample Application of Skills** | Create 4 emojis of your own invention. | Complete a Unit 1 review activity in Tynker. | Complete a Unit 1 quiz. | | | |

*See individual lesson guides for details on UK Computer standards

# Intro to Programming and Art                    Grades 9+

## Scope and Sequence

Students are introduced to art, design, and animation principles using Processing, a programming language designed for the visual arts community. Additionally, students will explore the works of famous masters such as Monet, Raphael, Picasso, and Seurat---then recreate digital art in the same style using JavaScript and p5.js.The coding exercises, hand-on Labs, and quizzes in this course engage students in developing computational thinking skills, as listed below from the CSTA Computer Science standards and UK Computing standards.

|  | Lesson 2.1 - Color Schemes | Lesson 2.2 - Irregular Shapes | Lesson 2.3 - Lab: Create a Logo | Lesson 2.4 - User-Defined Functions | Lesson 2.5 - Lab: Silhouette | Lesson 2.6 - Transparency |
|---|---|---|---|---|---|---|
| **Key Skills and Concepts** | • Explore hues, shades, and brightness. | • Apply knowledge of coordinates to draw shapes. | • Take a problem (image) and break it down into smaller parts. | • Create functions. | • Practice selecting and using a color scheme. | • Use opacity to create depth of field. |
| **CSTA Computer Science Standards** | 3A-IC-24 | 3A-AP-13<br>3A-AP-17<br>3A-AP-18<br>3B-AP-10<br>3B-AP-11 | 3A-AP-13<br>3A-AP-17<br>3A-AP-18<br>3B-AP-10<br>3B-AP-11 | 3A-AP-13<br>3A-AP-17<br>3A-AP-18<br>3B-AP-10<br>3B-AP-11 | 3A-AP-13<br>3A-AP-17<br>3A-AP-18<br>3B-AP-10<br>3B-AP-11 | 3A-AP-13<br>3A-AP-17<br>3A-AP-18<br>3B-AP-10<br>3B-AP-11 |
| **ISTE Standards** | 1.c | 1.c<br>1.d<br>4.d<br>5.c<br>5.d | 1.c<br>1.d<br>4.d<br>5.c<br>5.d | 1.c<br>1.d<br>4.d<br>5.c<br>5.d | 1.c<br>1.d<br>4.d<br>5.c<br>5.d | 1.c<br>1.d<br>4.d<br>5.c<br>5.d |
| **UK National Curriculum** | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* |
| **Sample Application of Skills** | Discuss the design of logos. | Draw irregular shapes on the canvas. | Create a logo of your own design. | Write user-defined functions. | Use code to create a silhouette. | Use opacity as a shading tool. |

*See individual lesson guides for details on UK Computer standards

# Intro to Programming and Art
## Scope and Sequence

### Grades 9+

Each unit includes a suggested pacing guide.

Students are introduced to art, design, and animation principles using Processing, a programming language designed for the visual arts community. Additionally, students will explore the works of famous masters such as Monet, Raphael, Picasso, and Seurat---then recreate digital art in the same style using JavaScript and p5.js.The coding exercises, hand-on Labs, and quizzes in this course engage students in developing computational thinking skills, as listed below from the CSTA Computer Science standards and UK Computing standards.

| | Lesson 2.7 - Lab: Origami Art | Lesson 2.8 - Randomness | Lesson 2.9 - Lab: Cubism | Lesson 2.10 - Review | Lesson 2.11 - Quiz | |
|---|---|---|---|---|---|---|
| **Key Skills and Concepts** | • Practice selecting and using a color scheme. | • Use the random() function. | • Practice selecting and using a color scheme. | • Review color combinations, HSB color space, transparcety, opacity, intervals, and functions. | • Demonstrate an understanding of functions, HSB color space, saturation and brightness, and drawing shapes. | |
| **CSTA Computer Science Standards** | 3A-AP-13 3A-AP-17 3A-AP-18 3B-AP-10 3B-AP-11 | 3A-AP-13 3A-AP-17 3A-AP-18 3B-AP-10 3B-AP-11 | 3A-AP-13 3A-AP-17 3A-AP-18 3B-AP-10 3B-AP-11 | 3A-AP-13 3B-AP-11 | 3A-AP-13 3A-AP-17 3A-AP-18 3B-AP-10 | |
| **ISTE Standards** | 1.c 1.d 4.d 5.c 5.d | 1.c 1.d 4.d 5.c 5.d | 1.c 1.d 4.d 5.c 5.d | 1.c 1.d 4.d 5.c 5.d | 1.c 1.d 4.d 5.c 5.d | |
| **UK National Curriculum** | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | |
| **Sample Application of Skills** | Create art that mimics the style of origami. | Apply the use of randomness to fill shapes. | Create Cubism art. | Complete a Unit 2 review activity in Tynker. | Complete a Unit 2 quiz. | |

*See individual lesson guides for details on UK Computer standards

# Intro to Programming and Art

## Scope and Sequence

## Grades 9+

Each unit includes a suggested pacing guide.

Students are introduced to art, design, and animation principles using Processing, a programming language designed for the visual arts community. Additionally, students will explore the works of famous masters such as Monet, Raphael, Picasso, and Seurat---then recreate digital art in the same style using JavaScript and p5.js.The coding exercises, hand-on Labs, and quizzes in this course engage students in developing computational thinking skills, as listed below from the CSTA Computer Science standards and UK Computing standards.

|  | Lesson 3.1 - Variables | Lesson 3.2 - Logic and Decisions | Lesson 3.3 - Loops | Lesson 3.4 - Lab: Refactoring | Lesson 3.5 - Gradients | Lesson 3.6 - Lab: Create a Cityscape |
|---|---|---|---|---|---|---|
| Key Skills and Concepts | • Declare and use variables. | • Use conditionals. | • Use "for" loops to reduce lines of code. | • Practice using loops and variables. | • Explore different examples of gradients. | • Practice using gradients. |
| CSTA Computer Science Standards | 3A-AP-13 3A-AP-17 3A-AP-18 3B-AP-10 3B-AP-11 | 3A-AP-13 3A-AP-17 3A-AP-18 3B-AP-10 3B-AP-11 | 3A-AP-13 3A-AP-17 3A-AP-18 3B-AP-10 3B-AP-11 | 3A-AP-13 3A-AP-17 3A-AP-18 3B-AP-10 3B-AP-11 | 3A-AP-13 3A-AP-17 3A-AP-18 3B-AP-10 3B-AP-11 | 3A-AP-13 3A-AP-17 3A-AP-18 3B-AP-10 3B-AP-11 |
| ISTE Standards | 1.c 1.d 4.d 5.c 5.d | 1.c 1.d 4.d 5.c 5.d | 1.c 1.d 4.d 5.c 5.d | 1.c 1.d 4.d 5.c 5.d | 1.c 1.d 4.d 5.c 5.d | 1.c 1.d 4.d 5.c 5.d |
| UK National Curriculum | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* |
| Sample Application of Skills | Identify common bugs. | Detect whether a condition is true and only run code in certain cases. | Identify patterns in yoru code. | Create a program that uses refactoring. | Use code to create gradients. | Use code to draw a cityscape scene. |

*See individual lesson guides for details on UK Computer standards

# Intro to Programming and Art

## Scope and Sequence

**Grades 9+**

Each unit includes a suggested pacing guide.

Students are introduced to art, design, and animation principles using Processing, a programming language designed for the visual arts community. Additionally, students will explore the works of famous masters such as Monet, Raphael, Picasso, and Seurat---then recreate digital art in the same style using JavaScript and p5.js.The coding exercises, hand-on Labs, and quizzes in this course engage students in developing computational thinking skills, as listed below from the CSTA Computer Science standards and UK Computing standards.

| | Lesson 3.7 - Shading: Creating Depth | Lesson 3.8 - Lab: Sol LeWitt | Lesson 3.9 - Review | Lesson 3.10 - Quiz | | |
|---|---|---|---|---|---|---|
| **Key Skills and Concepts** | • Create shapes that illustrate light sources, highlights, and depth. | • Use variables and functions to create irregular shapes. | • Review variables, scope, conditional statements, variables, loops, gradients, and highlights. | • Demonstrate an understanding of Unit 3 concepts. | | |
| **CSTA Computer Science Standards** | 3A-AP-13 3A-AP-17 3A-AP-18 3B-AP-10 3B-AP-11 | 3A-AP-13 3A-AP-17 3A-AP-18 3B-AP-10 3B-AP-11 | 3A-AP-13 3B-AP-11 | 3A-AP-13 3A-AP-17 3A-AP-18 3B-AP-10 | | |
| **ISTE Standards** | 1.c 1.d 4.d 5.c 5.d | 1.c 1.d 4.d 5.c 5.d | 1.c 1.d 4.d 5.c 5.d | 1.c 1.d 4.d 5.c 5.d | | |
| **UK National Curriculum** | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | | |
| **Sample Application of Skills** | Solve coding puzzles. | Use code to create Sol LeWitt-style art. | Complete a Unit 3 review activity in Tynker. | Complete a Unit 3 quiz. | | |

*See individual lesson guides for details on UK Computer standards

# Intro to Programming and Art

## Scope and Sequence

**Grades 9+**

Each unit includes a suggested pacing guide.

Students are introduced to art, design, and animation principles using Processing, a programming language designed for the visual arts community. Additionally, students will explore the works of famous masters such as Monet, Raphael, Picasso, and Seurat---then recreate digital art in the same style using JavaScript and p5.js.The coding exercises, hand-on Labs, and quizzes in this course engage students in developing computational thinking skills, as listed below from the CSTA Computer Science standards and UK Computing standards.

| | Lesson 4.1 - Advanced Logic | Lesson 4.2 - Pixel Art Challenge | Lesson 4.3 - Draw | Lesson 4.4 - Implicit Shading | Lesson 4.5 - Lab: Impressionism | 4.6 - Transformations |
|---|---|---|---|---|---|---|
| **Key Skills and Concepts** | • Use math operations. | • Use logical operators and conditional statements to colorize a grid. | • Use the draw function. | • Explore implicit functions. | • Use implicit shading to create an art effect. | • Explore coordinate transformations. |
| **CSTA Computer Science Standards** | 3A-AP-13 3A-AP-17 3A-AP-18 3B-AP-10 3B-AP-11 | 3A-AP-13 3A-AP-17 3A-AP-18 3B-AP-10 3B-AP-11 | 3A-AP-13 3A-AP-17 3A-AP-18 3B-AP-10 3B-AP-11 | 3A-AP-13 3A-AP-17 3A-AP-18 3B-AP-10 3B-AP-11 | 3A-AP-13 3A-AP-17 3A-AP-18 3B-AP-10 3B-AP-11 | 3A-AP-13 3A-AP-17 3A-AP-18 3B-AP-10 3B-AP-11 |
| **ISTE Standards** | 1.c, 1.d 4.d, 5.c 5.d | 1.c, 1.d 4.d, 5.c 5.d | 1.c, 1.d 4.d, 5.c 5.d | 1.c, 1.d 4.d, 5.c 5.d | 1.c, 1.d 4.d, 5.c 5.d | 1.c, 1.d 4.d, 5.c 5.d |
| **UK National Curriculum** | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* |
| **Sample Application of Skills** | Create programs that use logical operators. | Create patterns using conditional statements and loops. | Create interactive art work. | Use implicit functions to shade shapes and regions using code. | Create an artwork of your own imagination. | Create animations and moving pieces of art. |

*See individual lesson guides for details on UK Computer standards

## Scope and Sequence

Each unit includes a suggested pacing guide.

Students are introduced to art, design, and animation principles using Processing, a programming language designed for the visual arts community. Additionally, students will explore the works of famous masters such as Monet, Raphael, Picasso, and Seurat---then recreate digital art in the same style using JavaScript and p5.js.The coding exercises, hand-on Labs, and quizzes in this course engage students in developing computational thinking skills, as listed below from the CSTA Computer Science standards and UK Computing standards.

| | Lesson 4.7 - Lab: Perspective | Lesson 4.8 - Review | Lesson 4.9 - Quiz | | | |
|---|---|---|---|---|---|---|
| **Key Skills and Concepts** | • Use the coordinate transformation commands. | • Review the modulo operator, logical operators, De Morgan's Law, rows/columns, and equations. | • Demonstrate an understanding of the modulo operator, logical operators, De Morgan's Law, rows/columns, and equations. | | | |
| **CSTA Computer Science Standards** | 3A-AP-13 3A-AP-17 3A-AP-18 3B-AP-10 3B-AP-11 3B-AP-12 | 3A-AP-13 3A-AP-17 3A-AP-18 3B-AP-10 3B-AP-11 | 3A-AP-13 3A-AP-17 3A-AP-18 3B-AP-10 3B-AP-11 | | | |
| **ISTE Standards** | 1.c, 1.d 4.d,  5.c 5.d | 1.c, 1.d 4.d,  5.c 5.d | 1.c, 1.d 4.d,  5.c 5.d | | | |
| **UK National Curriculum** | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | | | |
| **Sample Application of Skills** | Use code to mimic a vanishing point artistic perspective. | Complete a Unit 4 review activity in Tynker. | Complete a Unit 4 quiz. | | | |

*See individual lesson guides for details on UK Computer standards

## Intro to Programming and Art

**Scope and Sequence**

**Grades 9+**

Each unit includes a suggested pacing guide.

Students are introduced to art, design, and animation principles using Processing, a programming language designed for the visual arts community. Additionally, students will explore the works of famous masters such as Monet, Raphael, Picasso, and Seurat---then recreate digital art in the same style using JavaScript and p5.js.The coding exercises, hand-on Labs, and quizzes in this course engage students in developing computational thinking skills, as listed below from the CSTA Computer Science standards and UK Computing standards.

| | Lesson 5.1 - Images | Lesson 5.2 - Lab: Pointillism | Lesson 5.3 - Arrays | Lesson 5.4 - Pixel Data | Lesson 5.5 - Lab: Andy Warhol | 5.6 - Steganography |
|---|---|---|---|---|---|---|
| **Key Skills and Concepts** | • Use RGB to tint digital photographs with code. | • Determine which file formats are accepted in p5.js. | • Explore array data type. | • Change pixel data. | • Use masks to recolor and change images. | • Practice iterating through pixel data. |
| **CSTA Computer Science Standards** | 3A-AP-13<br>3A-AP-17<br>3A-AP-18<br>3B-AP-10<br>3B-AP-11<br>3B-AP-17 | 3A-AP-13<br>3A-AP-17<br>3A-AP-18<br>3B-AP-10<br>3B-AP-11<br>3B-AP-17 | 3A-AP-13<br>3A-AP-17<br>3A-AP-18<br>3B-AP-10<br>3B-AP-11<br>3B-AP-17 | 3A-AP-13<br>3A-AP-17<br>3A-AP-18<br>3B-AP-10<br>3B-AP-11<br>3B-AP-17 | 3A-AP-13<br>3A-AP-17<br>3A-AP-18<br>3B-AP-10<br>3B-AP-11 | 3A-AP-13<br>3A-AP-17<br>3A-AP-18<br>3B-AP-10<br>3B-AP-11<br>3B-AP-21 |
| **ISTE Standards** | 1.c, 1.d<br>4.d, 5.c<br>5.d | 1.c, 1.d<br>4.d, 5.c<br>5.d | 1.c, 1.d<br>4.d, 5.c<br>5.d | 1.c, 1.d<br>4.d, 5.c<br>5.d | 1.c, 1.d<br>4.d, 5.c<br>5.d | 1.c, 1.d<br>4.d, 5.c<br>5.d |
| **UK National Curriculum** | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | |
| **Sample Application of Skills** | Create an image variable then draw it on the canvas. | Create an image with a pointillist effect. | Practice using advanced array manipulation techniques. | Create animations by changing pixel data. | Create an Andy Warhol inspired picture. | Decode a secret message hiding in an image. |

*See individual lesson guides for details on UK Computer standards

# Intro to Programming and Art

## Scope and Sequence

Students are introduced to art, design, and animation principles using Processing, a programming language designed for the visual arts community. Additionally, students will explore the works of famous masters such as Monet, Raphael, Picasso, and Seurat---then recreate digital art in the same style using JavaScript and p5.js.The coding exercises, hand-on Labs, and quizzes in this course engage students in developing computational thinking skills, as listed below from the CSTA Computer Science standards and UK Computing standards.

|  | Lesson 5.7 - Pixel Art Effects | Lesson 5.8 - Lab: Halftones | Lesson 5.9 - Review | Lesson 5.10 - Quiz |  |  |
|---|---|---|---|---|---|---|
| Key Skills and Concepts | • Compare methods of pixelated art to new methods. | • Use image processing. | • Review working with images, pixels/pixel art, steganography, and functions. | • Demonstrate an understanding of working with images, pixels/pixel art, steganography, and functions. |  |  |
| CSTA Computer Science Standards | 3A-AP-13<br>3A-AP-17<br>3A-AP-18<br>3B-AP-10<br>3B-AP-11<br>3B-AP-12 | 3A-AP-13<br>3A-AP-17<br>3A-AP-18<br>3B-AP-10<br>3B-AP-11<br>3B-AP-22 | 3A-AP-13<br>3A-AP-17<br>3A-AP-18<br>3B-AP-10<br>3B-AP-11 | 3A-AP-13<br>3A-AP-17<br>3A-AP-18<br>3B-AP-10<br>3B-AP-11 |  |  |
| ISTE Standards | 1.c, 1.d<br>4.d, 5.c<br>5.d | 1.c, 1.d<br>4.d, 5.c<br>5.d | 1.c, 1.d<br>4.d, 5.c<br>5.d | 1.c, 1.d<br>4.d, 5.c<br>5.d |  |  |
| UK National Curriculum | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* |  |  |
| Sample Application of Skills | Resize images and add pixelation. | Create a halfoning photo filter. | Complete a Unit 5 review activity in Tynker. | Complete a Unit 5 quiz. |  |  |

*See individual lesson guides for details on UK Computer standards

# Intro to Programming and Art

## Scope and Sequence

### Grades 9+

Each unit includes a suggested pacing guide.

Students are introduced to art, design, and animation principles using Processing, a programming language designed for the visual arts community. Additionally, students will explore the works of famous masters such as Monet, Raphael, Picasso, and Seurat---then recreate digital art in the same style using JavaScript and p5.js.The coding exercises, hand-on Labs, and quizzes in this course engage students in developing computational thinking skills, as listed below from the CSTA Computer Science standards and UK Computing standards.

| | Lesson 6.1 - Mouse Interaction | Lesson 6.2 - Keyboard Interaction | Lesson 6.3 - Objects | Lesson 6.4 - Lab: Picture Slideshow | Lesson 6.5 - Lab: Art Gallery | Lesson 6.6 - Review | Lesson 6.7 - Quiz |
|---|---|---|---|---|---|---|---|
| **Key Skills and Concepts** | • Explore mouse events in p5.js. | • Explore human-computer interaction | • Explore the object data type and learn how to make custom objects. | • Explore object-oriented programming. | • Convert p5.js projects into objects. | • Review mouse/keyboard interaction, creating objects, and creating classes. | • Take a quiz on mouse/keyboard interaction, creating objects, and creating classes. |
| **CSTA Computer Science Standards** | 3A-AP-13 3A-AP-17 3B-AP-10 3B-AP-11 3B-AP-17 3B-AP-18 | 3A-AP-13 3A-AP-17 3B-AP-10 3B-AP-11 3B-AP-17 3B-AP-18 | 3A-AP-13 3A-AP-17 3B-AP-10 3B-AP-11 3B-AP-17 3B-AP-18 | 3A-AP-13 3A-AP-17 3B-AP-10 3B-AP-11 3B-AP-17 3B-AP-18 | 3A-AP-13 3A-AP-17 3B-AP-10 3B-AP-11 3B-AP-17 3B-AP-18 | 3A-AP-13 3A-AP-17 3B-AP-10 3B-AP-11 3B-AP-17 3B-AP-18 | 3A-AP-13 3A-AP-17 3B-AP-10 3B-AP-11 3B-AP-17 3B-AP-18 |
| **ISTE Standards** | 1.c, 1.d 4.d, 5.c 5.d | 1.c, 1.d 4.d, 5.c 5.d | 1.c, 1.d 4.d, 5.c 5.d | 1.c, 1.d 4.d, 5.c 5.d | 1.c, 1.d 4.d, 5.c 5.d | 1.c, 1.d 4.d, 5.c 5.d | 1.c, 1.d 4.d, 5.c 5.d |
| **UK National Curriculum** | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* |
| **Sample Application of Skills** | Create a program that uses mouse events. | Add functions linked to specific keys being pressed. | Create a project using a custom object. | Create a slideshow project displaying art. | Create an art gallery project. | Complete a review activity in Tynker. | Complete a Unit 6 quiz. |

*See individual lesson guides for details on UK Computer standards

# Intro to Programming and Art

## Scope and Sequence

**Grades 9+**

Each unit includes a suggested pacing guide.

Students are introduced to art, design, and animation principles using Processing, a programming language designed for the visual arts community. Additionally, students will explore the works of famous masters such as Monet, Raphael, Picasso, and Seurat---then recreate digital art in the same style using JavaScript and p5.js.The coding exercises, hand-on Labs, and quizzes in this course engage students in developing computational thinking skills, as listed below from the CSTA Computer Science standards and UK Computing standards.

| | Lesson 7.1 - Intro to the Final Project | Lesson 7.2 - Breaking Bricks | Lesson 7.3 - Your Final Project | | | |
|---|---|---|---|---|---|---|
| **Key Skills and Concepts** | • Explore the technology industry and job titles in that industry. | • Explore a demonstrated project. | • Read about previous successful student-driven projects. | | | |
| **CSTA Computer Science Standards** | 3A-AP-17<br>3B-AP-10<br>3B-AP-17 | 3A-AP-13<br>3A-AP-17<br>3A-AP-18<br>3A-AP-19<br>3B-AP-10<br>3B-AP-11<br>3B-AP-17<br>3B-AP-22<br>3B-AP-23 | 3A-AP-13<br>3A-AP-17<br>3A-AP-18<br>3A-AP-19<br>3B-AP-10<br>3B-AP-11<br>3B-AP-17<br>3B-AP-22<br>3B-AP-23 | | | |
| **ISTE Standards** | 1.c, 1.d<br>4.d, 5.c<br>5.d | 1.c, 1.d<br>4.d, 5.c<br>5.d | 1.c, 1.d<br>4.d, 5.c<br>5.d | | | |
| **UK National Curriculum** | Key stage 3&4* | Key stage 3&4* | Key stage 3&4* | | | |
| **Sample Application of Skills** | Begin planning for a final group project. | Build a Brick Breaking game using code. | Write proposal documents. | | | |

*See individual lesson guides for details on UK Computer standards

# AP Computer Science Principles  Grades 9+

## Scope and Sequence
<span style="float:right">Each unit includes a suggested pacing guide.</span>

Tynker is recognized by the College Board as an endorsed provider of curriculum and professional development for AP® Computer Science Principles (AP CSP). The AP CSP course is a year-long high school curriculum that's designed to introduce students to the central ideas of computer science and prepare them for the AP CS Principles Exam. This course includes 12 units, 110 lessons. Here is the AP CSP College Board document: https://apcentral.collegeboard.org/pdf/ap-computer-science-principles-course-and-exam-description.pdf.

| | Lesson 1.1 - Welcome to AP CS Principles | Lesson 1.2 - Computing Innovations | Lesson 1.3 - Python Basics | Lesson 1.4 - What is Data? | Lesson 1.5 - Algorithms | Lesson 1.6 - Python Loops | Lesson 1.7 - Loops Part 2 | Lesson 1.8 - Computing Systems and Networks |
|---|---|---|---|---|---|---|---|---|
| **Big Ideas** | ● Impact of Computing **(IOC)** | ● Creative Development **(CRD)** | ● Algorithm and Programming **(AAP)** | ● Algorithm and Programming **(AAP)** ● Creative Development **(CRD)** | ● Algorithm and Programming **(AAP)** ● Creative Development **(CRD)** | ● Algorithm and Programming **(AAP)** | ● Algorithm and Programming **(AAP)** | ● Computing Systems and Networks **(CSN)** |
| **Computational Thinking Practices** | ● Computing Innovations **(5)** | ● Computing Innovations **(5)** | ● Algorithms and Program Development **(2)** | ● Abstraction in Program Development **(3)** | ● Algorithms and Program Development **(2)** ● Abstraction in Program Development **(3)** | ● Algorithms and Program Development **(2)** ● Code Analysis **(4)** | ● Algorithms and Program Development **(2)** ● Code Analysis **(4)** | ● Computing Innovations **(5)** |
| **Learning Objective** | IOC-1.A | CRD-2.A | AAP-2.B | AAP-1.A CRD-2.C CRD-2.D | AAP-2.A AAP-2.G AAP-2.J CRD-2.D | AAP-2.B AAP-2.C AAP-2.K | AAP-2.B AAP-2.C AAP-2.K | CSN-1.A |
| **Skills** | 5.C | 5.C | 2.B | 3.A | 2.A, 3.A | 2.B 4.B | 2.B 4.B | 5.A |
| **CSTA Computer Science Standards** | 3B-IC-25 | | | | | 3A-AP-21 3B-AP-10 3B-AP-11 | 3A-AP-21 3B-AP-10 3B-AP-11 | |
| **Sample Activity** | Explore computing innovation and how computing has changed our lives. | Define the term "Computing Innovation" and give several examples. | Solve coding puzzles using Python. | Explore how the term "data" applies to programming code and computing innovations. | Investigate the definition of "algorithm" and write an algorithm for an activity. | Implement iteration by using "for" loops. | Use nested "for" loops to reduce lines of code. | Make a graphical representation of computing networks. |

# AP Computer Science Principles    Grades 9+

## Scope and Sequence

Tynker is recognized by the College Board as an endorsed provider of curriculum and professional development for AP® Computer Science Principles (AP CSP). The AP CSP course is a year-long high school curriculum that's designed to introduce students to the central ideas of computer science and prepare them for the AP CS Principles Exam. This course includes 12 units, 110 lessons. Here is the AP CSP College Board document: https://apcentral.collegeboard.org/pdf/ap-computer-science-principles-course-and-exam-description.pdf.

| | Lesson 1.9 - Impacts on Society | Lesson 1.10 - Creative Development | Lesson 1.11 - Review | Lesson 1.12 - Quiz | Lesson 2.1 - Conditional Algorithms Unplugged | Lesson 2.2 - Conditional Logic Puzzles | Lesson 2.3 - Conditional Loops | Lesson 2.4 - Variables in Python |
|---|---|---|---|---|---|---|---|---|
| **Big Ideas** | ● Impact of Computing **(IOC)** | ● Creative Development **(CRD)** | | | ● Algorithm and Programming **(AAP)** | ● Algorithm and Programming **(AAP)** | ● Algorithm and Programming **(AAP)** | ● Algorithm and Programming **(AAP)** |
| **Computational Thinking Practices** | ● Computing Innovations **(5)** | ● Computational Solution Design **(1)** | | | ● Algorithms and Program Development **(2)** ● Code Analysis **(4)** | ● Algorithms and Program Development **(2)** ● Code Analysis **(4)** | ● Algorithms and Program Development **(2)** ● Code Analysis **(4)** | ● Code Implementation **(3)** ● Code Analysis **(4)** |
| **Learning Objectives** | IOC-1.A IOC-1.B | CRD-1.A CRD-1.B CRD-1.C | | | AAP-2.G AAP-2.H | AAP-2.H AAP-2.F AAP-2.E | AAP-2.K | AAP-1.A AAP-1.B |
| **Skills** | 5.C | 1.C | | | 2.A 2.B, 4.B | 2.B 4.B | 2.B 4.B | 3.A 4.B |
| **CSTA Computer Science Standards** | 3B-IC-25 | 3A-AP-21 3A-AP-22 3B-AP-10 3B-AP-11 3B-AP-21 3B-AP-23 | | | 3A-IC-26 | 3A-AP-21 3B-AP-10 3B-AP-11 | 3A-AP-21 3B-AP-10 3B-AP-11 | 3A-AP-21 3B-AP-10 3B-AP-11 3B-AP-12 |
| **Sample Activity** | Give examples of beneficial and harmful effects of recent computer innovations. | Work with a partner to create a program. | Discuss concepts and skills taught in Unit 1. | Complete a Unit 1 quiz. | Write a logical statement. | Use logical operators to check multiple conditions at the same time. | Use" while" loops to continue executing code while a condition is true. | Change the value of a variable and declare multiple variables. |

# AP Computer Science Principles

**Scope and Sequence**

**Grades 9+**

Each unit includes a suggested pacing guide.

Tynker is recognized by the College Board as an endorsed provider of curriculum and professional development for AP® Computer Science Principles (AP CSP). The AP CSP course is a year-long high school curriculum that's designed to introduce students to the central ideas of computer science and prepare them for the AP CS Principles Exam. This course includes 12 units, 110 lessons. Here is the AP CSP College Board document: https://apcentral.collegeboard.org/pdf/ap-computer-science-principles-course-and-exam-description.pdf.

| | Lesson 2.5 - User Input | Lesson 2.6 - Create Tasks: Trivia Game | Lesson 2.7 - Random Integers in Python | Lesson 2.8 - Rock Paper Scissors: Pair Programming | Lesson 2.9 - Conditionals with the Robot Language | Lesson 2.10 - Number Guessing Game | Lesson 2.11 - Create Task: Math Facts Game | Lesson 2.12 - Review |
|---|---|---|---|---|---|---|---|---|
| **Big Ideas** | ● Algorithm and Programming **(AAP)** | ● Creative Development **(CRD)** | ● Algorithm and Programming **(AAP)** | ● Creative Development **(CRD)** | ● Algorithm and Programming **(AAP)** | ● Creative Development **(CRD)** ● Algorithm and Programming **(AAP)** | ● Creative Development **(CRD)** | |
| **Computational Thinking Practices** | ● Code Implementation **(3)** ● Algorithms and Program Development **(2)** | | ● Algorithms and Program Development **(2)** ● Code Analysis **(4)** | ● Code Analysis **(4)** | ● Computational Solution Design **(1)** ● Algorithms and Program Development **(2)** | ● Algorithms and Program Development **(2)** ● Code Analysis **(4)** | ● Code Analysis **(4)** | |
| **Learning Objectives** | AAP-1.C AAP-2.E | CRD-2.E | AAP-3.E | CRD-2.J | AAP-2.A AAP-2.G AAP-2.J AAP-2.L | CRD-2.J AAP-2.K | CRD-2.J | |
| **Skills** | 2.B 3.A | | 2.B 4.B | 4.C | 1.D 2.A | 2.B 4.C | 4.C | |
| **CSTA Computer Science Standards** | 3A-AP-21 3B-AP-10 3B-AP-11 3B-AP-12 | 3A-AP-21 3B-AP-10 3B-AP-11 3B-AP-12 | 3A-AP-21 3B-AP-10 3B-AP-11 3B-AP-12 | 3A-AP-21 3A-AP-23 3B-AP-09 3B-AP-10 3B-AP-11 3B-AP-12 | 3A-AP-21 3B-AP-10 3B-AP-11 | 3A-AP-21 3A-AP-23 3B-AP-09 3B-AP-10 3B-AP-11 3B-AP-12 | 3A-AP-21 3A-AP-23 3B-AP-09 3B-AP-10 3B-AP-11 3B-AP-12 3B-AP-22 | |
| **Sample Activity** | Write programs that accept user input and process it using rational operators. | Implement a trivia game that counts the number of correct answers. | Generate random integers in Python. | Write a program that plays Rock Paper Scissors against the computer. | Follow a conditional algorithm using the AP CSP Robot Language. | Practice using an and operator in a Boolean expression that controls a while loop. | Practice using continue and break inside a while loop. | Discuss concepts and skills taught in Unit 2. |

# AP Computer Science Principles

## Scope and Sequence

**Grades 9+**

Each unit includes a suggested pacing guide.

Tynker is recognized by the College Board as an endorsed provider of curriculum and professional development for AP® Computer Science Principles (AP CSP). The AP CSP course is a year-long high school curriculum that's designed to introduce students to the central ideas of computer science and prepare them for the AP CS Principles Exam. This course includes 12 units, 110 lessons. Here is the AP CSP College Board document: https://apcentral.collegeboard.org/pdf/ap-computer-science-principles-course-and-exam-description.pdf.

| | Lesson 2.13 - Quiz | Lesson 3.1 - What is Abstraction | Lesson 3.2 - Variables and Expressions | Lesson 3.3 - Lists | Lesson 3.4 - The Turtle | Lesson 3.5 - User Interaction | Lesson 3.6 - Writing Functions with Parameters | Lesson 3.7 - Review |
|---|---|---|---|---|---|---|---|---|
| **Big Ideas** | | ● Data **(DAT)** | ● Algorithm and Programming **(AAP)** | ● Algorithm and Programming **(AAP)** | ● Algorithm and Programming **(AAP)** ● Data **(DAT)** | ● Creative Development **(CRD)** | ● Algorithm and Programming **(AAP)** | |
| **Computational Thinking Practices** | | ● Algorithms and Program Development **(2)** ● Code Implementation **(3)** | ● Algorithms and Program Development **(2)** ● Code Implementation **(3)** ● Code Analysis **(4)** | ● Code Implementation **(3)** | | ● Code Implementation **(3)** | ● Code Implementation **(3)** | |
| **Learning Objectives** | | DAT-1.A DAT-1.C | AAP-1.A AAP-1.B AAP-2.B | AAP-1.C AAP-1.D | AAP-3.D AAP-2.F AAP-3.E DAT-1.A | CRD-2.C | AAP-3.A AAP-3.B | |
| **Skills** | | 2.B 3.C | 3.A, 4.B 2.B | 3.A 3.B 3.C | 2.B 3.C | 3.A | 3.B 3.C | |
| **CSTA Computer Science Standards** | | 3A-CS-01 3A-DA-09 | 3A-AP-21 3B-AP-10 3B-AP-11 3B-AP-12 | 3A-AP-17 3A-AP-21 3B-AP-10 3B-AP-11 3B-AP-12 | 3A-AP-21 3B-AP-10 3B-AP-11 3B-AP-14 3B-AP-16 3A-DA-09 | 3A-AP-21 3B-AP-10 3B-AP-11 3B-AP-12 | 3A-AP-21 3B-AP-10 3B-AP-11 | |
| **Sample Activity** | Complete a Unit 2 quiz. | Define abstraction and give several examples. | Write programs that use Python's float type. | Explain how to create a list. | Create custom shapes using the turtle object. | Create a program that reacts to keyboard input. | Use Turtle commands such as speed, color, width, and shape. | Discuss concepts and skills taught in Unit 3. |

# AP Computer Science Principles

## Grades 9+

### Scope and Sequence

Each unit includes a suggested pacing guide.

Tynker is recognized by the College Board as an endorsed provider of curriculum and professional development for AP® Computer Science Principles (AP CSP). The AP CSP course is a year-long high school curriculum that's designed to introduce students to the central ideas of computer science and prepare them for the AP CS Principles Exam. This course includes 12 units, 110 lessons. Here is the AP CSP College Board document: https://apcentral.collegeboard.org/pdf/ap-computer-science-principles-course-and-exam-description.pdf.

| | Lesson 3.8 - Quiz | Lesson 3.9 - End of Quarter Create Task | Lesson 4.1 - Computing Networks | Lesson 4.2 - Dictionaries | Lesson 4.3 - The Internet | Lesson 4.4 - Packet Routing Simulation Program | Lesson 4.5 - Parallel Computing | Lesson 4.6 - Distributed Computing |
|---|---|---|---|---|---|---|---|---|
| **Big Ideas** | | ● Creative Development **(CRD)** | ● Computing Systems and Networks **(CSN)** | ● Data **(DAT)** | ● Computing Systems and Networks **(CSN)** | ● Creative Development **(CRD)** | ● Computing Systems and Networks **(CSN)** | ● Computing Systems and Networks **(CSN)** |
| **Computational Thinking Practices** | | ● Computational Solution Design **(1)** | ● Computing Innovations **(5)** | ● Algorithms and Program Development **(2)** | ● Computational Solution Design **(1)** <br> ● Computing Innovations **(5)** | ● Code Analysis **(4)** | ● Computational Solution Design **(1)** | ● Computational Solution Design **(1)** |
| **Learning Objectives** | | CRD-2.E <br> CRD-2.F <br> CRD-2.G | CSN-1.A | DAT-2.D | CSN-1.B <br> CSN-1.C <br> CSN-1.D <br> CSN-1.E | CRD-2.I | CSN-2.A <br> CSN-2.B | CSN-2.A <br> CSN-2.B |
| **Skills** | | 1.B <br> 4.A | 5.A | 2.B | 1.D <br> 5.A | 4.C | 1.D | 1.D |
| **CSTA Computer Science Standards** | | 3A-AP-21 <br> 3A-AP-23 <br> 3B-AP-10 <br> 3B-AP-11 <br> 3B-AP-12 <br> 3B-AP-14 <br> 3B-AP-22 | 3A-AP-21 <br> 3B-AP-10 <br> 3B-AP-11 | 3A-AP-21 <br> 3B-AP-10 <br> 3B-AP-11 <br> 3B-AP-12 | 3B-IC-25 | 3A-AP-21 <br> 3B-AP-10 <br> 3B-AP-11 <br> 3B-AP-16 | 3A-AP-21 <br> 3B-AP-10 <br> 3B-AP-11 <br> 3B-AP-16 | 3A-IC-24 |
| **Sample Activity** | Complete a Unit 3 quiz. | Design a program of your own choosing. | Use the terms path, routing, and bandwidth as they relate to networks. | Implement a letter frequency program with a dictionary. | Act out an Internet simulation of dynamic packet routing. | Create a Python program that simulates packet routing. | Practice an unplugged parallel computing algorithm. | Research a current example of distributed computing. |

# AP Computer Science Principles

## Scope and Sequence

**Grades 9+**

Each unit includes a suggested pacing guide.

Tynker is recognized by the College Board as an endorsed provider of curriculum and professional development for AP® Computer Science Principles (AP CSP). The AP CSP course is a year-long high school curriculum that's designed to introduce students to the central ideas of computer science and prepare them for the AP CS Principles Exam. This course includes 12 units, 110 lessons. Here is the AP CSP College Board document: https://apcentral.collegeboard.org/pdf/ap-computer-science-principles-course-and-exam-description.pdf.

| | Lesson 4.7 - Introduction to Game Design | Lesson 4.8 - Practice with Update | Lesson 4.9 - Pair Programming Challenges | Lesson 4.10 - Review | Lesson 4.11 - Quiz | Lesson 5.1 - Pong | Lesson 5.2 - Snake | Lesson 5.3 - Snake Follow Up |
|---|---|---|---|---|---|---|---|---|
| **Big Ideas** | ● Algorithm and Programming **(AAP)** | ● Creative Development **(CRD)** | ● Creative Development **(CRD)** | | | ● Algorithm and Programming **(AAP)** | ● Algorithm and Programming **(AAP)** | ● Algorithm and Programming **(AAP)** |
| **Computational Thinking Practices** | ● Algorithms and Program Development **(2)** | ● Code Implementation **(3)** | ● Code Analysis **(4)** | | | | | |
| **Learning Objectives** | AAP-2.N | CRD-2.D | CRD-2.J | | | AAP-2.H AAP-2.I | AAP-2.N | AAP-3.C AAP-1.D |
| **Skills** | 2.B | 3.A | 4.C | | | | | |
| **CSTA Computer Science Standards** | 3A-AP-21 3B-AP-10 3B-AP-11 3B-AP-12 | 3A-AP-21 3B-AP-10 3B-AP-11 | 3A-AP-21 3B-AP-10 3B-AP-11 3A-AP-22 | | | 3A-AP-21 3B-AP-10 3B-AP-11 | 3A-AP-21 3B-AP-10 3B-AP-11 | 3A-AP-21 3B-AP-10 3B-AP-11 |
| **Sample Activity** | Demonstrate advanced list manipulations. | Use constraints to make a program easier to understand. | Practice pair programming. | Review Unit 4 concepts. | Complete a Unit 1 quiz. | Create a replica of the classic arcade game Pong. | Create state variables that track the state of the game. | Create a complex algorithm that contains parameters. |

# AP Computer Science Principles

## Grades 9+

## Scope and Sequence

Each unit includes a suggested pacing guide.

Tynker is recognized by the College Board as an endorsed provider of curriculum and professional development for AP® Computer Science Principles (AP CSP). The AP CSP course is a year-long high school curriculum that's designed to introduce students to the central ideas of computer science and prepare them for the AP CS Principles Exam. This course includes 12 units, 110 lessons. Here is the AP CSP College Board document: https://apcentral.collegeboard.org/pdf/ap-computer-science-principles-course-and-exam-description.pdf.

| | Lesson 5.4 - Connect 4 | Lesson 5.5 - Connect 4 Follow-Up | Lesson 5.6 - Tetris | Lesson 5.7 - Tetris Follow Up | Lesson 5.8 - Frogga | Lesson 5.9 - Pair Programming Challenges | Lesson 5.10 - Review | Lesson 5.11 - Quiz |
|---|---|---|---|---|---|---|---|---|
| **Big Ideas** | ● Algorithm and Programming **(AAP)** | ● Algorithm and Programming **(AAP)** | ● Algorithm and Programming **(AAP)** | ● Algorithm and Programming **(AAP)** | ● Algorithm and Programming **(AAP)** | ● Creative Development **(CRD)** | | |
| **Computational Thinking Practices** | | | | | | ● Code Analysis **(4)** | | |
| **Learning Objectives** | AAP-2.M | AAP-3.C | AAP-2.M | AAP-3.C | AAP-2.M | CRD-2.J | | |
| **Skills** | | | | | | 4.C | | |
| **CSTA Computer Science Standards** | 3A-AP-21 3B-AP-10 3B-AP-11 | 3A-AP-13 3A-AP-21 3B-AP-10 3B-AP-11 | 3A-AP-21 3B-AP-10 3B-AP-11 | 3A-AP-21 3B-AP-10 3B-AP-11 | 3A-AP-21 3B-AP-10 3B-AP-11 | 3A-AP-21 3A-AP-22 3B-AP-10 3B-AP-11 | | |
| **Sample Activity** | Use nested "for" loops to create a grid. | Understand how rows and columns are represented by using a list of lists. | Apply coding concepts to store information about the shapes in a tuple. | Make a scoring sprite. | Create a Frogger game. | Solve challenges that review previous concepts. | Complete a review assignment in Tynker. | Complete a Unit 5 quiz. |

# AP Computer Science Principles

## Grades 9+

## Scope and Sequence

Each unit includes a suggested pacing guide.

Tynker is recognized by the College Board as an endorsed provider of curriculum and professional development for AP® Computer Science Principles (AP CSP). The AP CSP course is a year-long high school curriculum that's designed to introduce students to the central ideas of computer science and prepare them for the AP CS Principles Exam. This course includes 12 units, 110 lessons. Here is the AP CSP College Board document: https://apcentral.collegeboard.org/pdf/ap-computer-science-principles-course-and-exam-description.pdf.

| | Lesson 6.1 - Introduction | Lesson 6.2 - Starting and Testing | Lesson 6.3 - Data Abstraction | Lesson 6.4 - Choosing a Function |
|---|---|---|---|---|
| **Big Ideas** | ● Creative Development **(CRD)**<br>● Algorithm and Programming **(AAP)** | ● Creative Development **(CRD)**<br>● Algorithm and Programming **(AAP)** | ● Creative Development **(CRD)**<br>● Algorithm and Programming **(AAP)** | ● Creative Development **(CRD)**<br>● Algorithm and Programming **(AAP)** |
| **Computational Thinking Practices** | | | | |
| **Learning Objectives** | CRD-2.B<br>CRD-2.E<br>CRD-2.F<br>CRD-2.G<br>CRD-2.H<br>AAP-1.D<br>AAP-2.B<br>AAP-2.H<br>AAP-2.K<br>AAP-2.M<br>AAP-3.A<br>AAP-3.B<br>AAP-3.C | CRD-2.B<br>CRD-2.E<br>CRD-2.F<br>CRD-2.G<br>CRD-2.H<br>AAP-1.D<br>AAP-2.B<br>AAP-2.H<br>AAP-2.K<br>AAP-2.M<br>AAP-3.A<br>AAP-3.B<br>AAP-3.C | CRD-2.B<br>CRD-2.E<br>CRD-2.F<br>CRD-2.G<br>CRD-2.H<br>AAP-1.D<br>AAP-2.B<br>AAP-2.H<br>AAP-2.K<br>AAP-2.M<br>AAP-3.A<br>AAP-3.B<br>AAP-3.C | CRD-2.B<br>CRD-2.E<br>CRD-2.F<br>CRD-2.G<br>CRD-2.H<br>AAP-1.D<br>AAP-2.B<br>AAP-2.H<br>AAP-2.K<br>AAP-2.M<br>AAP-3.A<br>AAP-3.B<br>AAP-3.C |
| **Skills** | | | | |
| **CSTA Computer Science Standards** | 3A-AP-22 | 3A-AP-21<br>3B-AP-10<br>3B-AP-11 | 3A-AP-21<br>3B-AP-10<br>3B-AP-11 | 3A-AP-21<br>3B-AP-10<br>3B-AP-11 |
| **Sample Activity** | Create an elevator speech explaining your program. | Add comments to your code. | Describe the Data Abstractions in your program. | Describe a function in your program that has parameters. |

# AP Computer Science Principles

## Grades 9+

## Scope and Sequence

Each unit includes a suggested pacing guide.

Tynker is recognized by the College Board as an endorsed provider of curriculum and professional development for AP® Computer Science Principles (AP CSP). The AP CSP course is a year-long high school curriculum that's designed to introduce students to the central ideas of computer science and prepare them for the AP CS Principles Exam. This course includes 12 units, 110 lessons. Here is the AP CSP College Board document: https://apcentral.collegeboard.org/pdf/ap-computer-science-principles-course-and-exam-description.pdf.

| | Lesson 6.5 - Explaining How a Function Works | Lesson 6.6 - Completing Your Program | Lesson 6.7 - Written Responses | Lesson 6.8 - Reflection |
|---|---|---|---|---|
| Big Ideas | ● Creative Development **(CRD)**<br>● Algorithm and Programming **(AAP)** | ● Creative Development **(CRD)**<br>● Algorithm and Programming **(AAP)** | ● Creative Development **(CRD)**<br>● Algorithm and Programming **(AAP)** | ● Creative Development **(CRD)**<br>● Algorithm and Programming **(AAP)** |
| Computational Thinking Practices | 3A-AP-21, 3B-AP-10, 3B-AP-11 | 3A-AP-21, 3B-AP-10, 3B-AP-11 | 3B-AP-23 | |
| Learning Objectives | CRD-2.B<br>CRD-2.E<br>CRD-2.F<br>CRD-2.G<br>CRD-2.H<br>AAP-1.D<br>AAP-2.B<br>AAP-2.H<br>AAP-2.K<br>AAP-2.M<br>AAP-3.A<br>AAP-3.B<br>AAP-3.C | CRD-2.B<br>CRD-2.E<br>CRD-2.F<br>CRD-2.G<br>CRD-2.H<br>AAP-1.D<br>AAP-2.B<br>AAP-2.H<br>AAP-2.K<br>AAP-2.M<br>AAP-3.A<br>AAP-3.B<br>AAP-3.C | CRD-2.B<br>CRD-2.E<br>CRD-2.F<br>CRD-2.G<br>CRD-2.H<br>AAP-1.D<br>AAP-2.B<br>AAP-2.H<br>AAP-2.K<br>AAP-2.M<br>AAP-3.A<br>AAP-3.B<br>AAP-3.C | CRD-2.B<br>CRD-2.E<br>CRD-2.F<br>CRD-2.G<br>CRD-2.H<br>AAP-1.D<br>AAP-2.B<br>AAP-2.H<br>AAP-2.K<br>AAP-2.M<br>AAP-3.A<br>AAP-3.B<br>AAP-3.C |
| Skills | | | | |
| CSTA Computer Science Standards | | | | |
| Sample Activity | Describe how your function works. | Document assistance from others in your code. | Demonstrate a working program to the instructor. | Complete a reflection on the Semester Performance Task. |

# AP Computer Science Principles                           Grades 9+
## Scope and Sequence                       Each unit includes a suggested pacing guide.

Tynker is recognized by the College Board as an endorsed provider of curriculum and professional development for AP® Computer Science Principles (AP CSP). The AP CSP course is a year-long high school curriculum that's designed to introduce students to the central ideas of computer science and prepare them for the AP CS Principles Exam. This course includes 12 units, 110 lessons. Here is the AP CSP College Board document: https://apcentral.collegeboard.org/pdf/ap-computer-science-principles-course-and-exam-description.pdf.

| | Lesson 7.1 - Operations on Lists | Lesson 7.2 - Functions that Return a List | Lesson 7.3 - CSV Files | Lesson 7.4 - Making Plots | Lesson 7.5 - Thinking about Data Sets | Lesson 7.6 - Binary Search | Lesson 7.7 - Pair Programming Challenges | Lesson 7.8 - Review |
|---|---|---|---|---|---|---|---|---|
| Big Ideas | ● Algorithm and Programming (AAP) | ● Algorithm and Programming (AAP) | ● Data (DAT) | ● Data (DAT) | ● Data (DAT) | ● Algorithm and Programming (AAP) | ● Creative Development (CRD) | |
| Computational Thinking Practices | | | | | | | ● Code Analysis (4) | |
| Learning Objectives | AAP-2.O | AAP-2.O | DAT-2.D | DAT-2.E | DAT-2.A | AAP-2.P | CRD-2.J | |
| Skills | | | | | | | 4.C | |
| CSTA Computer Science Standards | 3A-AP-21 3B-AP-10 3A-AP-14 3B-AP-11 3B-AP-12 | 3A-AP-21 3B-AP-10 3B-AP-11 3B-AP-12 | 3A-DA-11 3A-DA-12 3A-AP-21 3B-AP-10 3B-AP-11 3B-AP-15 | 3A-DA-11 3A-AP-21 3B-AP-10 3B-AP-11 3B-AP-15 3B-AP-16 | 3A-DA-10 | 3A-AP-21 3B-AP-10 3B-AP-09 3B-AP-11 | 3A-AP-21 3B-AP-10 3B-AP-11 3B-AP-18 3B-AP-20 3B-AP-21 3B-AP-23 | |
| Sample Activity | Write functions that act on a single list. | Write functions that return a list. | Write a Python program that uses the csv module to open and read a CSV file into a list of lists. | Write a Python program that uses the pygal module to make plots. | Answer questions about what can and cannot be discovered from data sets. | Write a program that plays a number guessing game. | Practice pair programming. | Create a review assignment in Tynker. |

# AP Computer Science Principles

## Scope and Sequence

**Grades 9+**

Each unit includes a suggested pacing guide.

Tynker is recognized by the College Board as an endorsed provider of curriculum and professional development for AP® Computer Science Principles (AP CSP). The AP CSP course is a year-long high school curriculum that's designed to introduce students to the central ideas of computer science and prepare them for the AP CS Principles Exam. This course includes 12 units, 110 lessons. Here is the AP CSP College Board document: https://apcentral.collegeboard.org/pdf/ap-computer-science-principles-course-and-exam-description.pdf.

|  | Lesson 7.9 - Quiz | Lesson 8.1 - What is Artificial Intelligence? | Lesson 8.2 - Game of Stones | Lesson 8.3 - Rock Paper Scissors with Lists | Lesson 8.4 - Rock Paper Scissors with Prediction | Lesson 8.5 - Beneficial and Harmful Effects | Lesson 8.6 - Pair Programming Challenge: Final Jeopardy | Lesson 8.7 - Sentiment Analysis |
|---|---|---|---|---|---|---|---|---|
| **Big Ideas** |  | • Impact of Computing **(IOC)** | • Algorithm and Programming **(AAP)** | • Algorithm and Programming **(AAP)** | • Algorithm and Programming **(AAP)** | • Algorithm and Programming **(AAP)** | • Algorithm and Programming **(AAP)** | • Creative Development **(CRD)** <br> • Algorithm and Programming **(AAP)** <br> • Impact of Computing **(IOC)** |
| **Computational Thinking Practices** |  |  |  |  |  |  |  |  |
| **Learning Objectives** |  | IOC-1.A | AAP-2.C | AAP-1.D <br> AAP-2.K | AAP-1.D | IOC-1.B <br> IOC-1.D | AAP-2.L <br> AAP-2.I | CRD-2.C <br> AAP-3.D <br> IOC-1.E |
| **Skills** |  |  |  |  |  |  |  |  |
| **CSTA Computer Science Standards** |  | 3B-AP-08 | 3A-AP-21 <br> 3B-AP-10 <br> 3B-AP-11 <br> 3B-AP-12 | 3A-AP-21 <br> 3B-AP-09 <br> 3B-AP-10 <br> 3B-AP-11 <br> 3B-AP-12 | 3A-AP-21 <br> 3B-AP-09 <br> 3B-AP-10 <br> 3B-AP-11 <br> 3B-AP-12 | 3B-AP-08 | 3A-AP-21 <br> 3B-AP-09 <br> 3B-AP-10 <br> 3B-AP-11 <br> 3B-AP-12 <br> 3B-AP-18 <br> 3B-AP-20 <br> 3B-AP-21 | 3A-AP-21 <br> 3B-AP-10 <br> 3B-AP-11 <br> 3B-AP-16 |
| **Sample Activity** | Complete a Unit 7 quiz. | Provide examples of Artificial Intelligence. | Write a program that plays a game with a winning strategy. | Write a program that uses lists in a meaningful way. | Write different algorithms that model artificial intelligence. | Explore the beneficial and harmful effects of artificial intelligence. | Write a program that uses a complicated algorithm to make a Final Jeopardy wager. | Write a program that uses a dictionary to store information. |

# AP Computer Science Principles

**Grades 9+**

## Scope and Sequence

Each unit includes a suggested pacing guide.

Tynker is recognized by the College Board as an endorsed provider of curriculum and professional development for AP® Computer Science Principles (AP CSP). The AP CSP course is a year-long high school curriculum that's designed to introduce students to the central ideas of computer science and prepare them for the AP CS Principles Exam. This course includes 12 units, 110 lessons. Here is the AP CSP College Board document: https://apcentral.collegeboard.org/pdf/ap-computer-science-principles-course-and-exam-description.pdf.

| | Lesson 8.8 - Sentiment Analysis Part 2 | Lesson 8.9 - Simulations | Lesson 8.10 - Simulations Part 2 | Lesson 8.11 - Review | Lesson 8.12 - Quiz | Lesson 9.1 - Starting with the Directions | Lesson 9.2 - Understanding the Scoring Guide | Lesson 9.3 - Create Task Week 1 |
|---|---|---|---|---|---|---|---|---|
| **Big Ideas** | • Data **(DAT)** | • Algorithm and Programming **(AAP)** | • Data **(DAT)** | | | | | |
| **Computational Thinking Practices** | | | | | | | | |
| **Learning Objectives** | DAT-2.C DAT-2.D DAT-2.E | AAP-3.F | DAT-2.E | | | | | |
| **Skills** | | | | | | | | |
| **CSTA Computer Science Standards** | 3A-AP-21 3B-AP-10 3B-AP-11 | 3A-AP-21 3B-AP-10 3B-AP-11 3B-AP-12 3B-AP-16 | 3A-AP-21 3B-AP-10 3B-AP-11 3B-AP-12 3B-AP-16 | | | | | 3A-AP-13 3A-AP-15 3A-AP-16 3A-AP-21 3B-AP-10 3B-AP-11 3B-AP-19 |
| **Sample Activity** | Write a program that uses artificial intelligence concepts to make a prediction. | Explain how creating a computer simulation has benefits over real-world experiments. | Write a program that uses turtle graphics to make a histogram from dictionary information. | Complete review exercises in Tynker. | Complete a Unit 8 quiz. | Independently write a program for submission to AP Digital Portfolio. | Review Scoring Guidelines for the Create Task. | Independently write a program for submission to AP Digital Portfolio. |

# AP Computer Science Principles                                    Grades 9+
## Scope and Sequence                    Each unit includes a suggested pacing guide.

Tynker is recognized by the College Board as an endorsed provider of curriculum and professional development for AP® Computer Science Principles (AP CSP). The AP CSP course is a year-long high school curriculum that's designed to introduce students to the central ideas of computer science and prepare them for the AP CS Principles Exam. This course includes 12 units, 110 lessons. Here is the AP CSP College Board document: https://apcentral.collegeboard.org/pdf/ap-computer-science-principles-course-and-exam-description.pdf.

|  | Lesson 9.4 - Create Task Week 2 | Lesson 9.5 - Create Task Week 3 | Lesson 10.1 - Encryption | Lesson 10.2 - Substitution Cipher | Lesson 10.3 - Reasonable Time | Lesson 10.4 - The Knapsack Problem | Lesson 10.5 - Physical and Theoretical Limits | Lesson 10.6 - Compression |
|---|---|---|---|---|---|---|---|---|
| **Big Ideas** |  |  | • Algorithm and Programming **(AAP)** | • Algorithm and Programming **(AAP)** | • Algorithm and Programming **(AAP)** | • Algorithm and Programming **(AAP)** | • Algorithm and Programming **(AAP)** | • Data **(DAT)** |
| **Computational Thinking Practices** |  |  |  |  |  |  |  |  |
| **Learning Objectives** |  |  | AAP-2.D AAP-2.M | AAP-2.D AAP-3.D | AAP-4.A | AAP-4.A | AAP-4.B | DAT-1.D |
| **Skills** |  |  |  |  |  |  |  |  |
| **CSTA Computer Science Standards** | 3A-AP-13 3A-AP-16 3A-AP-21 3B-AP-10 3B-AP-11 3B-AP-19 | 3A-AP-13 3A-AP-16 3A-AP-21 3B-AP-10 3B-AP-11 3B-AP-19 | 3A-AP-21 3B-AP-10 3B-AP-11 | 3A-AP-21 3B-AP-10 3B-AP-11 3B-AP-12 | 3A-AP-21 3B-AP-10 3B-AP-11 3B-AP-12 | 3A-AP-21 3B-AP-10 3B-AP-11 3B-AP-12 | 3A-AP-21 3B-AP-10 3B-AP-11 3B-AP-12 3A-IC-26 | 3A-AP-21 3B-AP-10 3B-AP-11 |
| **Sample Activity** | Independently write a program for submission to AP Digital Portfolio. | Independently write a program for submission to AP Digital Portfolio. | Write a program that encrypts and decrypts messages made using the Caesar cipher. | Write a program that uses string concatenation to build a word, letter by letter. | Explain the concept of reasonable time, and provide an example of an algorithm that does not run in reasonable time. | Write a program that uses tuples to store information. | Explain theoretical limitations of computers. | Write a program that implements lossy/lossless compression. |

# AP Computer Science Principles

## Grades 9+

## Scope and Sequence

Each unit includes a suggested pacing guide.

Tynker is recognized by the College Board as an endorsed provider of curriculum and professional development for AP® Computer Science Principles (AP CSP). The AP CSP course is a year-long high school curriculum that's designed to introduce students to the central ideas of computer science and prepare them for the AP CS Principles Exam. This course includes 12 units, 110 lessons. Here is the AP CSP College Board document: https://apcentral.collegeboard.org/pdf/ap-computer-science-principles-course-and-exam-description.pdf.

| | Lesson 10.7 - Computing Concerns | Lesson 10.8 - Review | Lesson 10.9 - Quiz | Lesson 11.1 - Understanding the Exam Format | Lesson 11.2 - Taking a Practice Exam | Lesson 11.3 - Reviewing Your Answers | Lesson 11.4 - The Day Before the Exam | Lesson 12.1 - Introduction to Recursion |
|---|---|---|---|---|---|---|---|---|
| Big Ideas | • Impact of Computing **(IOC)** | | | | | | | • Algorithm and Programming **(AAP)** |
| Computational Thinking Practices | | | | | | | | |
| Learning Objectives | IOC-1.C<br>IOC-1.D<br>IOC-1.F | | | | | | | AAP-3.A |
| Skills | | | | | | | | |
| CSTA Computer Science Standards | 3A-IC-24<br>3A-IC-25<br>3A-IC-30 | | | | | | | 3A-AP-21<br>3B-AP-10<br>3B-AP-11<br>3B-AP-14 |
| Sample Activity | Explain the idea behind public key encryption. | Complete a review assignment in Tynker. | Complete a Unit 10 quiz. | Read about the AP's pseudocode styles and its equivalent in Python. | Independently take an official AP Practice Example, provided by the College Board. | Self-grade their completed Practice Exam and review incorrect answers. | Revisit previous sections of the course. | Write a Python program that uses recursion to find the nth Fibonacci number. |

# AP Computer Science Principles

**Grades 9+**

## Scope and Sequence

Each unit includes a suggested pacing guide.

Tynker is recognized by the College Board as an endorsed provider of curriculum and professional development for AP® Computer Science Principles (AP CSP). The AP CSP course is a year-long high school curriculum that's designed to introduce students to the central ideas of computer science and prepare them for the AP CS Principles Exam. This course includes 12 units, 110 lessons. Here is the AP CSP College Board document: https://apcentral.collegeboard.org/pdf/ap-computer-science-principles-course-and-exam-description.pdf.

| | Lesson 12.2 - The Flood Fill Algorithm | Lesson 12.3 - Fractals | Lesson 12.4 - Fractal Trees | Lesson 12.5 - Review | Lesson 12.6 - Quiz | Lesson 12.7 - Optional Explore Task | | |
|---|---|---|---|---|---|---|---|---|
| **Big Ideas** | • Algorithm and Programming **(AAP)** | • Algorithm and Programming **(AAP)** | • Algorithm and Programming **(AAP)** | | | • Impact of Computing **(IOC)** <br>• Creative Development **(CRD)** | | |
| **Computational Thinking Practices** | | | | | | | | |
| **Learning Objectives** | AAP-3.A | AAP-3.A | AAP-3.A | | | CRD-2.A <br> IOC-1.A <br> IOC-2.A | | |
| **Skills** | | | | | | | | |
| **CSTA Computer Science Standards** | 3A-AP-21 <br> 3B-AP-10 <br> 3B-AP-11 <br> 3B-AP-13 <br> 3B-AP-14 <br> 3B-AP-17 | 3A-AP-21 <br> 3B-AP-10 <br> 3B-AP-11 <br> 3B-AP-17 | 3A-AP-21 <br> 3B-AP-10 <br> 3B-AP-11 <br> 3B-AP-17 | | | 3A-AP-13 <br> 3A-AP-16 <br> 3A-AP-20 <br> 3A-IC-24 <br> 3A-IC-28 <br> 3A-IC-30 | | |
| **Sample Activity** | Write a Python program that uses a recursive Flood Fill algorithm to fill an image. | Write a Python program that uses a recursive algorithm to make a fractal snowflake. | Write a Python program that uses a recursive algorithm to make a fractal tree. | Complete review exercises inTynker, then discuss results. | Complete a Unit 12 quiz. | Perform open-ended research. | | |

# AP Computer Science A

## Grades 9+

## Scope and Sequence

Each unit includes a suggested pacing guide.

Tynker is recognized by the College Board as an endorsed provider of curriculum and professional development for AP® Computer Science A (AP CSA). The AP CSA course is a year-long high school curriculum that's designed to introduce students to the central ideas of computer science and prepare them for the AP CS A Exam. This course includes 10 units, 123 lessons. Here's a link to the AP CSA document: https://apcentral.collegeboard.org/pdf/ap-computer-science-a-course-and-exam-description.pdf?course=ap-computer-science-a.

| | Lesson 1.1 - Warm Up! | Lesson 1.2 - Introduction | Lesson 1.3 - Visualizing Algorithms | Lesson 1.4 - Printing to the Console | Lesson 1.5 - Strings and Input | Lesson 1.6 - Abstraction | Lesson 1.7 - More on Methods | Lesson 1.8 - Composition |
|---|---|---|---|---|---|---|---|---|
| **Big Ideas** | | | | • Modularity **(MOD)** | • Variables **(VAR)** | • Modularity **(MOD)** | | • Variables **(VAR)** |
| **Computational Thinking Practices** | • Program Design and Algorithm Development **(1)** | • Program Design and Algorithm Development **(1)** | • Program Design and Algorithm Development **(1)** <br> • Code Testing **(4)** | | | | • Program Design and Algorithm Development **(1)** <br> • Code Testing **(4)** | • Program Design and Algorithm Development **(1)** <br> • Code Testing **(4)** |
| **Learning Objectives** | | | | MOD-1.A <br> MOD-1.A.1 <br> MOD-1.A.2 | VAR-1.A <br> VAR-1.A.1 | MOD-1.E.2 | | VAR-1.G.3 |
| **Skills** | 1.A | 1.A <br> 1.B | 1.A <br> 1.B <br> 4.B | | | | 1.B <br> 4.B | 1.B <br> 4.B |
| **CSTA Computer Science Standards** | | 3A-IC-26 <br> 3B-AP-10 <br> 3B-AP-11 | 3A-IC-26 <br> 3B-AP-10 <br> 3B-AP-11 | 3A-AP-21 <br> 3B-AP-10 <br> 3B-AP-11 | 3A-AP-21 <br> 3B-AP-10 <br> 3B-AP-11 | 3A-CS-01 <br> 3B-AP-10 <br> 3B-AP-11 | 3B-AP-10 <br> 3B-AP-11 | 3B-AP-10 <br> 3B-AP-11 |
| **Sample Activity** | Identify Java as a programming language. | Define what an algorithm is. | Visualize an algorithm using a flowchart. | Create shapes using print commands. | Concatenate strings. | Identify how to create a method. | Solve coding puzzles by creating your own method. | Compare composition within coding to other art forms. |

# AP Computer Science A

## Scope and Sequence

**Grades 9+**

*Each unit includes a suggested pacing guide.*

Tynker is recognized by the College Board as an endorsed provider of curriculum and professional development for AP® Computer Science A (AP CSA). The AP CSA course is a year-long high school curriculum that's designed to introduce students to the central ideas of computer science and prepare them for the AP CS A Exam. This course includes 10 units, 123 lessons. Here's a link to the AP CSA document: https://apcentral.collegeboard.org/pdf/ap-computer-science-a-course-and-exam-description.pdf?course=ap-computer-science-a.

| | Lesson 1.9 - ASCII Art Methods | Lesson 1.10 - Algorithms and Methods Review | Lesson 1.11 - Algorithms and Methods Quiz | Lesson 1.12 - Encoding Information | Lesson 1.13 - Binary Numbers | Lesson 1.14 - Variables | Lesson 1.15 - How to Use Variables | Lesson 1.16 - Types |
|---|---|---|---|---|---|---|---|---|
| **Big Ideas** | • Modularity **(MOD)**<br>• Variables **(VAR)** | | | | | • Control **(CON)** | • Variables **(VAR)** | • Variables **(VAR)** |
| **Computational Thinking Practices** | • Program Design and Algorithm Development **(1)**<br>• Code Testing **(4)** | | | • Program Design and Algorithm Development **(1)**<br>• Code Testing **(4)** | • Logic Code **(2)** | | | |
| **Learning Objectives** | MOD-1.E.2<br>VAR-1.G.3 | VAR-1.G.3 | | | | CON-1.B.1<br>CON-1.B.2<br>CON-1.B.3 | VAR-1.C.4 | VAR-1.B.2<br>VAR-1.B.3<br>VAR-1.C<br>VAR-1.C.1<br>VAR-1.C.2<br>VAR-1.C.3 |
| **Skills** | 1.B<br>4.B | | | 1.B<br>4.B | 2.B | | | |
| **CSTA Computer Science Standards** | 3A-AP-21<br>3A-AP-13<br>3B-AP-10<br>3B-AP-11 | 3B-AP-10<br>3B-AP-11 | | 3B-AP-10<br>3B-AP-11 | 3B-AP-10<br>3B-AP-11 | 3B-AP-10<br>3B-AP-11 | 3A-AP-21<br>3B-AP-10<br>3B-AP-11 | 3A-AP-21<br>3B-AP-10<br>3B-AP-11 |
| **Sample Activity** | Write methods that print out text to form basic shapes. | Reinforce concepts such as methods, abstraction, and composition. | Demonstrate your understanding of refactoring, abstraction, methods, and composition. | Explain what encoding is. | Convert from binary to decimal and vice versa. | Assign variables and declare the values as integers or strings. | Assign and reassign variables. | Identify the different types of values in Java. |

# AP Computer Science A

## Scope and Sequence

**Grades 9+**

Each unit includes a suggested pacing guide.

Tynker is recognized by the College Board as an endorsed provider of curriculum and professional development for AP® Computer Science A (AP CSA). The AP CSA course is a year-long high school curriculum that's designed to introduce students to the central ideas of computer science and prepare them for the AP CS A Exam. This course includes 10 units, 123 lessons. Here's a link to the AP CSA document: https://apcentral.collegeboard.org/pdf/ap-computer-science-a-course-and-exam-description.pdf?course=ap-computer-science-a.

| | Lesson 1.17 - What's an Expression | Lesson 1.18 - Variable Initialization in Java | Lesson 1.19 - Working with Numbers | Lesson 1.20 - Information Representation and Expressions Review | Lesson 1.21 - Information Representation and Expressions Quiz | | | |
|---|---|---|---|---|---|---|---|---|
| **Big Ideas** | • Variables **(VAR)**<br>• Control **(CON)** | • Control **(CON)** | • Control **(CON)** | • Control **(CON)** | • Variables **(VAR)** | | | |
| **Computational Thinking Practices** | | | | | | | | |
| **Learning Objectives** | CON-1.A<br>CON-1.A.2<br>VAR-1.C<br>CON-1.A.6<br>CON-1.B<br>CON-1.B.1<br>CON-1.B.2<br>CON-1.B.3<br>CON-1.C | CON-1.A.2<br>CON-1.A.3<br>CON-1.A.4<br>CON-1.A.5<br>CON-1.A.7<br>CON-1.B.5 | CON-1.A.8 | CON-1.A.6<br>CON-1.A.8<br>CON-1.B.4<br>CON-1.C | VAR-1.B | | | |
| **Skills** | | | | | | | | |
| **CSTA Computer Science Standards** | 3A-AP-21<br>3B-AP-10<br>3B-AP-11 | 3B-AP-10<br>3B-AP-11 | 3B-AP-10<br>3B-AP-11 | | | | | |
| **Sample Activity** | Explain the difference between syntax and semantic errors. | Practice using operators with extra focus on the modulus operator. | Demonstrate how to truncate values. | Complete review exercises inTynker, then discuss results. | Complete a Unit 1 quiz. | | | |

# AP Computer Science A

## Grades 9+

### Scope and Sequence

Each unit includes a suggested pacing guide.

Tynker is recognized by the College Board as an endorsed provider of curriculum and professional development for AP® Computer Science A (AP CSA). The AP CSA course is a year-long high school curriculum that's designed to introduce students to the central ideas of computer science and prepare them for the AP CS A Exam. This course includes 10 units, 123 lessons. Here's a link to the AP CSA document:
https://apcentral.collegeboard.org/pdf/ap-computer-science-a-course-and-exam-description.pdf?course=ap-computer-science-a.

| | Lesson 2.1 - Variables and Memory | Lesson 2.2 - Properties and Behaviors | Lesson 2.3 - Classes | Lesson 2.4 - Instance Variables and Methods | Lesson 2.5 - Constructors | Lesson 2.6 - Drawing Shapes | Lesson 2.7 - Introduction to Strings | Lesson 2.8 - String Methods |
|---|---|---|---|---|---|---|---|---|
| **Big Ideas** | • Modularity **(MOD)** | • Modularity **(MOD)** | • Variables **(VAR)**<br>• Modularity **(MOD)** | • Modularity **(MOD)** | • Modularity **(MOD)** | • Modularity **(MOD)** | • Variables **(VAR)** | • Variables **(VAR)** |
| **Computational Thinking Practices** | | | | | | | | |
| **Learning Objectives** | MOD-2.F.5<br>MOD-2.F.6 | MOD-2.B.1<br>MOD-1.E.1 | VAR-1<br>MOD-2.B.1<br>MOD-1.E.1 | MOD-2.B.1<br>MOD-1.E.1 | MOD-2.B.1<br>MOD-1.E.1 | MOD-2.B.1<br>MOD-1.D.1<br>MOD-1.D.3 | VAR-1.E.1<br>VAR-1.E.3<br>VAR-1.E.4 | VAR-1.E.10<br>VAR-1.E.12 |
| **Skills** | | | | | | | | |
| **CSTA Computer Science Standards** | 3A-AP-21<br>3B-AP-10<br>3B-AP-11<br>3B-AP-21 | 3A-AP-13<br>3A-AP-23<br>3B-AP-10<br>3B-AP-11<br>3B-AP-14 | 3A-AP-17<br>3A-AP-21<br>3B-AP-10<br>3B-AP-11<br>3B-AP-21 | 3A-AP-21<br>3B-AP-10<br>3B-AP-11<br>3B-AP-14<br>3B-AP-21 | 3A-AP-21<br>3B-AP-10<br>3B-AP-11<br>3B-AP-14<br>3B-AP-21 | 3A-AP-21<br>3B-AP-10<br>3B-AP-11<br>3B-AP-14<br>3B-AP-21 | 3A-AP-21<br>3B-AP-10<br>3B-AP-11 | 3A-AP-21<br>3B-AP-10<br>3B-AP-11<br>3B-AP-16 |
| **Sample Activity** | Trace and execute recursive methods. | Explore the fundamental ideas behind objects and object-oriented programming. | Practice making custom types (classes) in Java. | Create custom types, classes, and instances. | Create a simple class that represents a bicycle. | Use methods to create shapes on the canvas. | Create programs that use strings. | View examples of various String methods. |

# AP Computer Science A

## Grades 9+

### Scope and Sequence

Each unit includes a suggested pacing guide.

Tynker is recognized by the College Board as an endorsed provider of curriculum and professional development for AP® Computer Science A (AP CSA). The AP CSA course is a year-long high school curriculum that's designed to introduce students to the central ideas of computer science and prepare them for the AP CS A Exam. This course includes 10 units, 123 lessons. Here's a link to the AP CSA document: https://apcentral.collegeboard.org/pdf/ap-computer-science-a-course-and-exam-description.pdf?course=ap-computer-science-a.

| | Lesson 2.9 - More String Methods | Lesson 2.10 - Substrings | Lesson 2.11 - The Math Class | Lesson 2.12 - Review | Lesson 2.13 - Quiz | | | |
|---|---|---|---|---|---|---|---|---|
| Big Ideas | • Variables **(VAR)** | • Variables **(VAR)** | • Modularity **(MOD)** <br> • Control **(CON)** | • Variables **(VAR)** | • Variables **(VAR)** <br> • Modularity **(MOD)** | | | |
| Computational Thinking Practices | | | | | | | | |
| Learning Objectives | VAR-1.E.10 <br> VAR-1.E.12 | VAR-1.E.10 <br> VAR-1.E.12 | MOD-1.H.1 <br> CON-1.D <br> CON-1.D.1 <br> CON-1.D.2 <br> CON-1.D.3 | VAR-1.E.1 <br> VAR-1.E.3 <br> VAR-1.E.4 | VAR-1.E.1 <br> VAR-1.E.3 <br> VAR-1.E.4 <br> MOD-2.B.1 <br> MOD-1.E.1 | | | |
| Skills | | | | | | | | |
| CSTA Computer Science Standards | 3A-AP-21 <br> 3B-AP-10 <br> 3B-AP-11 | 3A-AP-21 <br> 3B-AP-10 <br> 3B-AP-11 | 3A-AP-21 <br> 3B-AP-10 <br> 3B-AP-11 | 3A-AP-13 <br> 3A-AP-23 <br> 3B-AP-10 <br> 3B-AP-11 <br> 3B-AP-14 | 3A-AP-21 <br> 3B-AP-10 <br> 3B-AP-11 | | | |
| Sample Activity | Explore more String methods, such as "compareTo" and "equal". | Explore the substring method of String and practice using it. | Explore the Java Math class. | Complete review exercises in Tynker, then discuss results. | Complete a Unit 2 quiz. | | | |

# AP Computer Science A

## Scope and Sequence

**Grades 9+**

Each unit includes a suggested pacing guide.

Tynker is recognized by the College Board as an endorsed provider of curriculum and professional development for AP® Computer Science A (AP CSA). The AP CSA course is a year-long high school curriculum that's designed to introduce students to the central ideas of computer science and prepare them for the AP CS A Exam. This course includes 10 units, 123 lessons. Here's a link to the AP CSA document:
https://apcentral.collegeboard.org/pdf/ap-computer-science-a-course-and-exam-description.pdf?course=ap-computer-science-a.

| | Lesson 3.1 - Simple Boolean Expressions | Lesson 3.2 - Logical Operators in Java | Lesson 3.3 - Using Relational and Logical Operators Together | Lesson 3.4 - Conditional Statements | Lesson 3.5 - Else and Else-If Statements | Lesson 3.6 - Conditionals Review | Lesson 3.7 - Conditionals Quiz | |
|---|---|---|---|---|---|---|---|---|
| **Big Ideas** | • Control **(CON)** | • Control **(CON)** | • Control **(CON)** | • Control **(CON)** | • Control **(CON)** | • Control **(CON)** | • Control **(CON)** | |
| **Computational Thinking Practices** | | | | | | | | |
| **Learning Objectives** | CON-1<br>CON-1.E<br>CON-1.E.2<br>CON-1.E.3<br>CON-1.F.2 | CON-1.A.6<br>CON-1.B.2 | CON-1.B.4 | CON-2.C<br>CON-2.C.2<br>CON-2.C.4 | CON-2.A.1<br>CON-2.A.2<br>CON-2.A.5<br>CON-2.C.2<br>CON-2.C.4<br>CON-1.G<br>CON-1.G.1<br>CON-1.G.3<br>CON-1.H<br>CON-1.H.1<br>CON-1.H.2<br>CON-1.H.3<br>CON-1.H.4 | CON-1.A.6<br>CON-1.B.2<br>CON-1<br>CON-1.E<br>CON-1.E.2<br>CON-1.E.3<br>CON-1.F.2 | CON-1.A.6<br>CON-1.B.2<br>CON-1<br>CON-1.E<br>CON-1.E.2<br>CON-1.E.3<br>CON-1.F.2<br>CON-2.C.2<br>CON-2.C.4 | |
| **Skills** | | | | | | | | |
| **CSTA Computer Science Standards** | 3A-AP-21<br>3B-AP-10<br>3B-AP-11 | | 3A-AP-17 | 3A-AP-21<br>3B-AP-10<br>3B-AP-11 | 3A-AP-21<br>3B-AP-10<br>3B-AP-11<br>3B-AP-21 | 3A-AP-21<br>3B-AP-10<br>3B-AP-11 | 3A-AP-21<br>3B-AP-10<br>3B-AP-11 | |
| **Sample Activity** | Explore examples of boolean operators in Java. | Use logical expressions as a tool for branching in a programming language. | Build complex boolean expressions in Java from relational and boolean operators. | Read about the "else" clause in an "if-else" construct. | Explore "else" and "if-else" constructs in Java. | Complete review exercises in Tynker, then discuss results. | Complete a Unit 3 quiz. | |

# AP Computer Science A
## Scope and Sequence

**Grades 9+**

Each unit includes a suggested pacing guide.

Tynker is recognized by the College Board as an endorsed provider of curriculum and professional development for AP® Computer Science A (AP CSA). The AP CSA course is a year-long high school curriculum that's designed to introduce students to the central ideas of computer science and prepare them for the AP CS A Exam. This course includes 10 units, 123 lessons. Here's a link to the AP CSA document: https://apcentral.collegeboard.org/pdf/ap-computer-science-a-course-and-exam-description.pdf?course=ap-computer-science-a.

| | Lesson 4.1 - The While Loop | Lesson 4.2 - Take Control of Loops | Lesson 4.3 - Infinite Loops | Lesson 4.4 - Draw with Loops | Lesson 4.5 - Randomness | Lesson 4.6 - While Loop Puzzles | Lesson 4.7 - While Loops Review | Lesson 4.8 - While Loops Quiz |
|---|---|---|---|---|---|---|---|---|
| **Big Ideas** | • Control **(CON)** | • Control **(CON)** | • Control **(CON)** | • Control **(CON)** | • Control **(CON)** | • Control **(CON)** | • Control **(CON)** | • Control **(CON)** |
| **Computational Thinking Practices** | | | | | | | | |
| **Learning Objectives** | CON-2.C.1 CON-2.C.2 | CON-2.D.2 | CON-2.C.1 CON-2.C.2 CON-2.C.3 | CON-2 CON-2.C.1 CON-2.C.2 CON-2.C.3 CON-2.C.4 | CON-1.D | CON-2.G | CON-2.C.1 CON-2.C.2 | CON-2.C |
| **Skills** | | | | | | | | |
| **CSTA Computer Science Standards** | 3A-AP-21 3B-AP-10 3B-AP-11 3B-AP-21 | 3A-AP-21 3B-AP-10 3B-AP-11 3B-AP-21 | 3A-AP-21 3B-AP-10 3B-AP-11 3B-AP-21 | 3A-AP-17 3A-AP-21 3B-AP-10 3B-AP-11 3B-AP-21 | 3A-AP-17 3A-AP-21 3B-AP-10 3B-AP-11 3B-AP-21 | 3A-AP-21 3B-AP-10 3B-AP-11 3B-AP-21 | 3A-AP-21 3B-AP-10 3B-AP-11 3B-AP-21 | 3A-AP-21 3B-AP-10 3B-AP-11 3B-AP-21 |
| **Sample Activity** | Solve coding puzzles using "while" loops and methods. | Implement different styles of counters. | Read about "input controlled loops" and "sentinel values." | Design a custom open-ended art project using code. | Create a computerized coin flipper. | Use "while" loops and conditional statements to solve coding puzzles. | Reinforce while loops, conditional statements, and random numbers. | Take a quiz to demonstrate your understanding of while loops, conditional statements, and random numbers. |

Tynker is recognized by the College Board as an endorsed provider of curriculum and professional development for AP® Computer Science A (AP CSA). The AP CSA course is a year-long high school curriculum that's designed to introduce students to the central ideas of computer science and prepare them for the AP CS A Exam. This course includes 10 units, 123 lessons. Here's a link to the AP CSA document: https://apcentral.collegeboard.org/pdf/ap-computer-science-a-course-and-exam-description.pdf?course=ap-computer-science-a.

| | Lesson 4.9 - For Loop Basic | Lesson 4.10 - Nested Loops | Lesson 4.11- Accumulating, Filtering, Mapping | Lesson 4.12 - Loop Challenges | Lesson 4.13 - Common Pitfalls | Lesson 4.14 - Review | Lesson 4.15 - Quiz | Lesson 4.16 - Lab: Consumer Review |
|---|---|---|---|---|---|---|---|---|
| **Big Ideas** | • Control **(CON)** | • Control **(CON)** | • Control **(CON)** | • Control **(CON)** | • Control **(CON)** | • Control **(CON)** | • Control **(CON)** | • Variables **(VAR)**<br>• Modularity **(MOD)**<br>• Control **(CON)**<br>• Impact of Computing **(IOC)** |
| **Computational Thinking Practices** | | | | | | | | |
| **Learning Objectives** | CON-2.E.1<br>CON-2.E.2<br>CON-2.E.3<br>CON-2.E.4<br>CON-2.H.1 | CON-2.E.1<br>CON-2.E.2<br>CON-2.E.3<br>CON-2.E.4<br>CON-2.G.1<br>CON-2.G.2 | CON-2.E.1<br>CON-2.E.2<br>CON-2.E.3<br>CON-2.E.4 | CON-2.E.1<br>CON-2.E.2<br>CON-2.E.3<br>CON-2.E.4<br>CON-2.G.1<br>CON-2.G.2 | CON-2.E.1<br>CON-2.E.2<br>CON-2.E.3<br>CON-2.E.4<br>CON-2.E.5 | CON-2.E.1<br>CON-2.E.2<br>CON-2.E.3<br>CON-2.E.4 | CON-2.G.1 | VAR-1.A<br>MOD-1.H<br>MOD-1.G<br>VAR-1.E<br>CON-2.A<br>CON-1.F<br>CON-2.C<br>CON-2.F<br>CON-2.G<br>IOC-1.A |
| **Skills** | | | | | | | | |
| **CSTA Computer Science Standards** | 3A-AP-21<br>3B-AP-10<br>3B-AP-11 | 3A-AP-21<br>3B-AP-10<br>3B-AP-11 | 3A-AP-21<br>3B-AP-10<br>3B-AP-11<br>3B-AP-12<br>3B-AP-21 | 3A-AP-21<br>3B-AP-10<br>3B-AP-11<br>3B-AP-21 | 3A-AP-21<br>3B-AP-10<br>3B-AP-11<br>3B-AP-21 | 3A-AP-13<br>3B-AP-10<br>3B-AP-11 | 3A-AP-21<br>3B-AP-10<br>3B-AP-11<br>3B-AP-21 | 3A-AP-21<br>3A-AP-23<br>3B-AP-10<br>3B-AP-11<br>3B-AP-14<br>3B-AP-21 |
| **Sample Activity** | Filter values and manipulate Strings. | Use loops to create images on a graph. | Filter values and manipulate Strings. | Use loops to create images on a graph. | Fix identified errors and create working "for" and "while" loops. | Complete review exercises in Tynker, then discuss results. | Take a Unit 4 quiz. | Create an open-ended project. |

# AP Computer Science A

**Scope and Sequence**

## Grades 9+

Each unit includes a suggested pacing guide.

Tynker is recognized by the College Board as an endorsed provider of curriculum and professional development for AP® Computer Science A (AP CSA). The AP CSA course is a year-long high school curriculum that's designed to introduce students to the central ideas of computer science and prepare them for the AP CS A Exam. This course includes 10 units, 123 lessons. Here's a link to the AP CSA document: https://apcentral.collegeboard.org/pdf/ap-computer-science-a-course-and-exam-description.pdf?course=ap-computer-science-a.

| | Lesson 5.1 - Constructors | Lesson 5.2 - 'this' | Lesson 5.3 - Intangible Objects | Lesson 5.4 - Adding Functionality | Lesson 5.5 - toString | Lesson 5.6 - Separation of Concerns | Lesson 5.7 - Setters and Getters | Lesson 5.8 - Writing Setters and Getters |
|---|---|---|---|---|---|---|---|---|
| **Big Ideas** | • Modularity **(MOD)** | • Variables **(VAR)**<br>• Modularity **(MOD)** | • Modularity **(MOD)** | • Modularity **(MOD)** | • Modularity **(MOD)** | • Modularity **(MOD)** | • Modularity **(MOD)** | • Modularity **(MOD)** |
| **Computational Thinking Practices** | | | | | | | | |
| **Learning Objectives** | MOD-2.B.1<br>MOD-1.E.1 | MOD-2.B<br>MOD-2.B.1<br>VAR-1.G.4<br>VAR-1.H<br>VAR-1.H.1 | MOD-2.B.5<br>MOD-2.C<br>MOD-2.C.1<br>MOD-2.C.2<br>MOD-2.C.3<br>MOD-2.C.4<br>MOD-2.C.5 | MOD-2.D.6<br>MOD-2.E | MOD-2.D.7<br>MOD-2.D.6 | MOD-3.A<br>MOD-3.A.2<br>MOD-3.A.3 | MOD-3.A.4<br>MOD-2.D.1 | MOD-2.E |
| **Skills** | | | | | | | | |
| **CSTA Computer Science Standards** | 3A-AP-21<br>3B-AP-10<br>3B-AP-11<br>3B-AP-21 | 3A-AP-21 3B-AP-10<br>3B-AP-11 3B-AP-21 | 3A-AP-21<br>3B-AP-10<br>3B-AP-11 | 3A-AP-21<br>3B-AP-10<br>3B-AP-11<br>3B-AP-21 | 3A-AP-21<br>3B-AP-10<br>3B-AP-11<br>3B-AP-21<br>3B-AP-22 | 3A-AP-21<br>3B-AP-10<br>3B-AP-11<br>3B-AP-21 | 3A-AP-21<br>3B-AP-10<br>3B-AP-11<br>3B-AP-21 | 3A-AP-21<br>3B-AP-10<br>3B-AP-11<br>3B-AP-21 |
| **Sample Activity** | Apply coding concepts to make a constructor. | Explore the keyword 'this' in a Java class. | Explore objects that are written to represent abstract (non-concrete) ideas. | Practice adding more functionality to classes. | Explore the ideas of instance variables (the properties of a class) and instance methods (the behaviors) of a class. | Practice making variables private. | Explore getters and setters methods. | Create a project that uses getters and setters. |

# AP Computer Science A

## Scope and Sequence

**Grades 9+**

Each unit includes a suggested pacing guide.

Tynker is recognized by the College Board as an endorsed provider of curriculum and professional development for AP® Computer Science A (AP CSA). The AP CSA course is a year-long high school curriculum that's designed to introduce students to the central ideas of computer science and prepare them for the AP CS A Exam. This course includes 10 units, 123 lessons. Here's a link to the AP CSA document: https://apcentral.collegeboard.org/pdf/ap-computer-science-a-course-and-exam-description.pdf?course=ap-computer-science-a.

| | Lesson 5.9 - National Parks | Lesson 5.10 - Impact of Computing | Lesson 5.11 - Review | Lesson 5.12 - Quiz | | | | |
|---|---|---|---|---|---|---|---|---|
| **Big Ideas** | • Modularity **(MOD)** | • Impact of Computing **(IOC)** | • Modularity **(MOD)** | • Modularity **(MOD)** | | | | |
| **Computational Thinking Practices** | | | | | | | | |
| **Learning Objectives** | MOD-2<br>MOD-2.D.1<br>MOD-2.G.2<br>MOD-2.G.3<br>MOD-2.G.4<br>MOD-2.G.5 | IOC-1.A<br>IOC-1.A.1<br>IOC-1.A.2<br>IOC-1.A.3 | MOD-2.B.1<br>MOD-1.E.1<br>MOD-2.D.6 | MOD-2.B.1<br>MOD-1.E.1<br>MOD-2.D.6<br>MOD-2.E | | | | |
| **Skills** | | | | | | | | |
| **CSTA Computer Science Standards** | 3A-AP-21<br>3B-AP-10<br>3B-AP-11<br>3B-AP-21 | 3A-IC-24<br>3A-IC-26<br>3A-IC-27<br>3A-IC-30<br>3B-NI-03<br>3B-IC-25<br>3B-IC-27<br>3B-AP-15<br>3B-AP-18 | 3A-AP-13<br>3B-AP-10<br>3B-AP-11 | 3A-AP-13<br>3B-AP-10<br>3B-AP-11 | | | | |
| **Sample Activity** | Practice creating classes and implementing the **toString()** method in Java. | Identify ethical and social implications of computing systems. | Complete review exercises in Tynker, then discuss results. | Complete a Unit 5 quiz. | | | | |

# AP Computer Science A

## Scope and Sequence

**Grades 9+**

Each unit includes a suggested pacing guide.

Tynker is recognized by the College Board as an endorsed provider of curriculum and professional development for AP® Computer Science A (AP CSA). The AP CSA course is a year-long high school curriculum that's designed to introduce students to the central ideas of computer science and prepare them for the AP CS A Exam. This course includes 10 units, 123 lessons. Here's a link to the AP CSA document: https://apcentral.collegeboard.org/pdf/ap-computer-science-a-course-and-exam-description.pdf?course=ap-computer-science-a.

| | Lesson 6.1 The Case for Arrays | Lesson 6.2 - Creating Arrays | Lesson 6.3 - For Loops and Arrays | Lesson 6.4 - Processing Arrays | Lesson 6.5 - Taking Flight | Lesson 6.6 - Arrays Review | Lesson 6.7 - Arrays Quiz | Lesson 6.8 - Arrays and Objects |
|---|---|---|---|---|---|---|---|---|
| **Big Ideas** | • Variables **(VAR)** | • Variables **(VAR)** | • Variables **(VAR)** | • Control **(CON)** | • Control **(CON)**<br>• Variables **(VAR)** | • Variables **(VAR)** | • Variables **(VAR)** | • Variables **(VAR)** |
| **Computational Thinking Practices** | | | | | | | | |
| **Learning Objectives** | VAR-2.A<br>VAR-2.A.1<br>VAR-2.A.6<br>VAR-2.A.7 | VAR-2.A.3<br>VAR-2.A.4 | VAR-2.B.3<br>VAR-2.B.1<br>VAR-2.B.2 | CON-2.I.1 | CON-2.I.1<br>VAR-2.B.2<br>VAR-2.B<br>CON-2.I | VAR-2.A<br>VAR-2.A.1 | VAR-2.A<br>VAR-2.B.2 | VAR-2.A.3<br>VAR-2.A.6 |
| **Skills** | | | | | | | | |
| **CSTA Computer Science Standards** | 3A-AP-21<br>3B-AP-10<br>3B-AP-11<br>3B-AP-12<br>3B-AP-13 | 3A-AP-21 3B-AP-10<br>3B-AP-11 3B-AP-21<br>3B-AP-22 | 3A-AP-21<br>3B-AP-10<br>3B-AP-11<br>3B-AP-21<br>3B-AP-22 | 3A-AP-21<br>3B-AP-10<br>3B-AP-11 | 3A-AP-21<br>3B-AP-10<br>3B-AP-11<br>3B-AP-22 | 3B-AP-13 | 3A-AP-13<br>3B-AP-10<br>3B-AP-11<br>3B-AP-14 | 3A-AP-21<br>3B-AP-10<br>3B-AP-11 |
| **Sample Activity** | Use arrays with integers or Strings in code.. | Add a "for-loop" with an array. | Determine the length of arrays. | Generate a random card from a standard deck of cards. | Process an array to determine the cheapest price. | Reinforce what you learned in Unit 6 about arrays. | Take a quiz to demonstrate your understanding of arrays. | Practice creating arrays. |

# AP Computer Science A

## Scope and Sequence

**Grades 9+**

Each unit includes a suggested pacing guide.

Tynker is recognized by the College Board as an endorsed provider of curriculum and professional development for AP® Computer Science A (AP CSA). The AP CSA course is a year-long high school curriculum that's designed to introduce students to the central ideas of computer science and prepare them for the AP CS A Exam. This course includes 10 units, 123 lessons. Here's a link to the AP CSA document: https://apcentral.collegeboard.org/pdf/ap-computer-science-a-course-and-exam-description.pdf?course=ap-computer-science-a.

| | Lesson 6.9 - Comma Separated Values | Lesson 6.10 - Arrays and Objects 2 | Lesson 6.11 - ToDo List | Lesson 6.12 - Dream Journal | Lesson 6.13 - Arrays and Objects Review | Lesson 6.14 - Arrays and Objects Quiz | Lesson 6.15 - Global Temperatures | Lesson 6.16 - Graphing Temperatures |
|---|---|---|---|---|---|---|---|---|
| **Big Ideas** | | | • Variables **(VAR)** | • Variables **(VAR)** | | | | • Control **(CON)** |
| **Computational Thinking Practices** | • Program Design and Algorithm Development **(1)**<br>• Code Testing **(4)** | • Program Design and Algorithm Development **(1)**<br>• Code Testing **(4)** | • Program Design and Algorithm Development **(1)**<br>• Code Testing **(4)** | • Program Design and Algorithm Development **(1)**<br>• Code Testing **(4)** | | | • Program Design and Algorithm Development **(1)** | |
| **Learning Objectives** | | | VAR-2.A | VAR-2.A<br>VAR-2.A.3 | | | | CON-2.I.1 |
| **Skills** | 1.A<br>1.B<br>4.B | 1.A<br>1.B<br>4.B | 1.A<br>1.B<br>4.B | 1.A<br>1.B<br>4.B | | | 1.A | |
| **CSTA Computer Science Standards** | 3A-AP-21<br>3B-AP-10<br>3B-AP-11<br>3B-AP-21<br>3B-AP-22 | 3A-AP-21<br>3B-AP-10<br>3B-AP-11<br>3B-AP-21 | 3A-AP-13<br>3A-AP-21<br>3B-AP-10<br>3B-AP-11<br>3B-AP-21 | 3A-AP-13<br>3A-AP-16<br>3A-AP-21<br>3A-AP-22<br>3B-AP-10<br>3B-AP-11<br>3B-AP-19<br>3B-AP-21<br>3B-AP-22<br>3B-AP-23 | | | 3A-AP-21<br>3B-AP-10<br>3B-AP-11<br>3B-AP-21<br>3B-AP-22 | 3A-AP-20<br>3A-AP-21<br>3A-DA-10<br>3A-DA-11<br>3A-DA-12<br>3B-AP-10<br>3B-AP-11<br>3B-AP-21<br>3B-DA-05<br>3B-DA-06 |
| **Sample Activity** | Create a program that analyzes the top YouTube videos of the day. | Begin creating a Java portfolio. | Create a To-Do application. | Create a Dream Journal application. | Review Unit 6 concepts. | Take a quiz on arrays and objects. | Create methods to determine high and low values. | Use NASA temperature data and Java methods to tell a story. |

# AP Computer Science A

## Scope and Sequence

**Grades 9+**

Each unit includes a suggested pacing guide.

Tynker is recognized by the College Board as an endorsed provider of curriculum and professional development for AP® Computer Science A (AP CSA). The AP CSA course is a year-long high school curriculum that's designed to introduce students to the central ideas of computer science and prepare them for the AP CS A Exam. This course includes 10 units, 123 lessons. Here's a link to the AP CSA document: https://apcentral.collegeboard.org/pdf/ap-computer-science-a-course-and-exam-description.pdf?course=ap-computer-science-a.

| | Lesson 6.17 - Regional Temperatures | Lesson 6.18 - ForEach Loops | Lesson 6.19 - Lab: Music | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Big Ideas** | • Control **(CON)** | • Variables **(VAR)** | • Variables **(VAR)** | | | | | |
| **Computational Thinking Practices** | | | • Program Design and Algorithm Development **(1)** <br> • Code Testing **(4)** | | | | | |
| **Learning Objectives** | CON-2.I | VAR-2.C <br> VAR-2.C.1 <br> VAR-2.C.2 <br> VAR-2.C.3 <br> VAR-2.C.4 | VAR-2.A <br> VAR-2.A.1 <br> VAR-2.A.3 | | | | | |
| **Skills** | | | 1.A <br> 1.B <br> 4.B | | | | | |
| **CSTA Computer Science Standards** | 3A-AP-21 <br> 3A-DA-11 <br> 3B-AP-10 <br> 3B-AP-11 <br> 3B-AP-21 <br> 3B-DA-05 <br> 3B-DA-06 | 3A-AP-21 <br> 3B-AP-10 <br> 3B-AP-11 <br> 3B-AP-21 | 3A-AP-13 <br> 3A-AP-16 <br> 3A-AP-21 <br> 3A-AP-23 <br> 3B-AP-10 <br> 3B-AP-11 <br> 3B-AP-21 <br> 3B-AP-22 | | | | | |
| **Sample Activity** | Identify differences between data sets. | Write a method that uses a "for-each" loop. | Design an open-ended project of your choosing. | | | | | |

# AP Computer Science A

**Grades 9+**

## Scope and Sequence

Each unit includes a suggested pacing guide.

Tynker is recognized by the College Board as an endorsed provider of curriculum and professional development for AP® Computer Science A (AP CSA). The AP CSA course is a year-long high school curriculum that's designed to introduce students to the central ideas of computer science and prepare them for the AP CS A Exam. This course includes 10 units, 123 lessons. Here's a link to the AP CSA document:
https://apcentral.collegeboard.org/pdf/ap-computer-science-a-course-and-exam-description.pdf?course=ap-computer-science-a.

| | Lesson 7.1 - Introduction to ArrayLists | Lesson 7.2 - Using ArrayLists | Lesson 7.3 - Traversing ArrayLists | Lesson 7.4 - ArrayLists | Lesson 7.5 - Parallel ArrayLists | Lesson 7.6 - Selection Sort | Lesson 7.7 - Insertion Sort |
|---|---|---|---|---|---|---|---|
| **Big Ideas** | • Variables **(VAR)** | • Variables **(VAR)** | • Variables **(VAR)** | • Variables **(VAR)** | • Control **(CON)** | • Control **(CON)** | • Control **(CON)** |
| **Computational Thinking Practices** | | | | | | | |
| **Learning Objectives** | VAR-2.D<br>VAR-2.D.1<br>VAR-2.D.2<br>VAR-2.D.3<br>VAR-2.D.4<br>VAR-2.D.5<br>VAR-2.D.6 | VAR-2.D<br>VAR-2.D.7 | VAR-2.D.7<br>VAR-2.E<br>VAR-2.E.1<br>VAR-2.E.4 | VAR-2.E.1<br>VAR-2.E.2<br>VAR-2.E.3<br>VAR-2.E.4 | CON-2.J<br>CON-2.J.1<br>CON-2.J.2 | CON-2.L<br>CON-2.L.1 | CON-2.L<br>CON-2.L.1 |
| **Skills** | | | | | | | |
| **CSTA Computer Science Standards** | 3A-AP-14<br>3A-AP-21<br>3B-AP-10<br>3B-AP-11<br>3B-AP-21 | 3A-AP-14<br>3A-AP-21<br>3B-AP-10<br>3B-AP-11<br>3B-AP-21 | 3A-AP-14<br>3A-AP-21<br>3B-AP-10<br>3B-AP-11<br>3B-AP-21 | 3A-AP-14<br>3A-AP-21<br>3B-AP-10<br>3B-AP-11<br>3B-AP-21 | 3A-AP-14<br>3A-AP-21<br>3B-AP-10<br>3B-AP-11<br>3B-AP-21 | 3A-AP-21<br>3B-AP-10<br>3B-AP-11<br>3B-AP-21 | 3A-AP-21<br>3B-AP-10<br>3B-AP-11<br>3B-AP-21 |
| **Sample Activity** | Explore the ArrayList data type, including the syntax to import the required Java packages. | Explore how elements are arranged in ArrayLists. | Explore loops used to traverse ArrayLists. | Practice general techniques for processing ArrayLists. | Traverse multiple ArrayLists simultaneously. | Explore famous searching and sorting algorithms. | Implement the insertion sort algorithm in Java. |

# AP Computer Science A

**Scope and Sequence**

# Grades 9+

Each unit includes a suggested pacing guide.

Tynker is recognized by the College Board as an endorsed provider of curriculum and professional development for AP® Computer Science A (AP CSA). The AP CSA course is a year-long high school curriculum that's designed to introduce students to the central ideas of computer science and prepare them for the AP CS A Exam. This course includes 10 units, 123 lessons. Here's a link to the AP CSA document:
https://apcentral.collegeboard.org/pdf/ap-computer-science-a-course-and-exam-description.pdf?course=ap-computer-science-a.

|  | Lesson 7.8 - Binary Search | Lesson 7.9 - Risks to Privacy | Lesson 7.10 - Review | Lesson 7.11 - Unit Quiz | Lesson 7.12 - Lab: Data |  |  |
|---|---|---|---|---|---|---|---|
| **Big Ideas** | • Control **(CON)** | • Control **(CON)**<br>• Impact of Computing **(IOC)** | • Control **(CON)**<br>• Variables **(VAR)**<br>• Impact of Computing **(IOC)** | • Control **(CON)** | • Modularity **(MOD)**<br>• Variables **(VAR)**<br>• Impact of Computing **(IOC)** |  |  |
| **Computational Thinking Practices** |  |  |  |  |  |  |  |
| **Learning Objectives** | CON-2.K<br>CON-2.K.1<br>CON-2.K.2<br>CON-2.L<br>CON-2.L.1 | CON-2.M<br>CON-2.M.1<br>IOC-1.B<br>IOC-1.B.1<br>IOC-1.B.2 | VAR-2.D<br>VAR-2.D.1 - VAR-2.D.7<br>VAR-2.E.1 - VAR-2.E.4<br>CON-2.J.1 - CON-2.J.2<br>CON-2.K.1 - CON-2.K.2<br>CON-2.L.1<br>CON-2.M.1<br>IOC-1.B<br>IOC-1.B.1<br>IOC-1.B.2 | CON-2.J<br>CON-2.K.1 | MOD-2.B<br>MOD-2.B.1<br>VAR-1.G.4<br>IOC-1.B<br>IOC-1.B.1<br>IOC-1.B.2 |  |  |
| **Skills** |  |  |  |  |  |  |  |
| **CSTA Computer Science Standards** | 3A-AP-21<br>3B-AP-10<br>3B-AP-11<br>3B-AP-21 | 3A-NI-07<br>3A-AP-21<br>3B-AP-10<br>3B-AP-11 | 3A-AP-13<br>3B-AP-10<br>3B-AP-11 | 3A-AP-13<br>3B-AP-10<br>3B-AP-11 | 3A-IC-27, 3A-IC-30<br>3A-AP-13, 3A-AP-21<br>3A-AP-23, 3A-DA-10<br>3A-DA-11, 3A-DA-12<br>3B-AP-10, 3B-AP-11<br>3B-AP-21, 3B-IC-25<br>3B-DA-05, 3B-DA-06 |  |  |
| **Sample Activity** | Implement a selection sort that works with integers. | Perform calculations exploring the idea of computational efficiency. | Complete a review assessment in Tynker. | Demonstrate your knowledge of Unit 7 concepts by completing a quiz. | Create 3 lab projects. |  |  |

Tynker is recognized by the College Board as an endorsed provider of curriculum and professional development for AP® Computer Science A (AP CSA). The AP CSA course is a year-long high school curriculum that's designed to introduce students to the central ideas of computer science and prepare them for the AP CS A Exam. This course includes 10 units, 123 lessons. Here's a link to the AP CSA document: https://apcentral.collegeboard.org/pdf/ap-computer-science-a-course-and-exam-description.pdf?course=ap-computer-science-a.

| | Lesson 8.1 - Arrays of Arrays | Lesson 8.2 - Random Haikus | Lesson 8.3 - Multidimensional Arrays | Lesson 8.4 - Dimensions | Lesson 8.5 - Processing in Multiple Dimensions | Lesson 8.6 - Review | Lesson 8.7 - Unit Quiz | Lesson 8.8 - Lab: The 256 Game |
|---|---|---|---|---|---|---|---|---|
| **Big Ideas** | • Modularity (**MOD**) | • Variables (**VAR**) | • Variables (**VAR**) | • Variables (**VAR**) | • Variables (**VAR**)<br>• Control (**CON**) | • Variables (**VAR**)<br>• Control (**CON**) | • Variables (**VAR**)<br>• Control (**CON**) | • Variables (**VAR**)<br>• Control (**CON**) |
| **Computational Thinking Practices** | | | | | | | | |
| **Learning Objectives** | VAR-2.F<br>VAR-2.F.1<br>VAR-2.F.2<br>VAR-2.F.3<br>VAR-2.F.4<br>VAR-2.G<br>VAR-2.G.1 | VAR-2.F<br>VAR-2.F.1<br>VAR-2.F.2<br>VAR-2.F.3 | VAR-2.F<br>VAR-2.F.1<br>VAR-2.F.2<br>VAR-2.F.3<br>VAR-2.F.4<br>VAR-2.F.5<br>VAR-2.G<br>VAR-2.G.2<br>VAR-2.G.3 | VAR-2.F.2<br>VAR-2.F.3<br>VAR-2.F.4<br>VAR-2.F.5<br>VAR-2.G.1<br>VAR-2.G.2<br>VAR-2.G.3 | VAR-2.F.2<br>VAR-2.F.3<br>VAR-2.F.4<br>VAR-2.F.5<br>VAR-2.G.1<br>VAR-2.G.2<br>VAR-2.G.3<br>CON-2.N.1<br>CON-2.N.2 | VAR-2.F.1<br>VAR-2.F.2<br>VAR-2.F.3<br>VAR-2.F.4<br>VAR-2.F.5<br>VAR-2.G.1<br>VAR-2.G.2<br>VAR-2.G.3<br>CON-2.N.1<br>CON-2.N.2 | VAR-2.F<br>VAR-2.F.4<br>CON-2.N | VAR-2.F<br>VAR-2.F.4<br>VAR-2.G<br>VAR-2.G.2<br>CON-2.N<br>CON-2.N.1 |
| **Skills** | | | | | | | | |
| **CSTA Computer Science Standards** | 3A-AP-21<br>3B-AP-10<br>3B-AP-11<br>3B-AP-21<br>3B-AP-22 | 3A-AP-21<br>3B-AP-10<br>3B-AP-11<br>3B-AP-21 | 3A-AP-21<br>3B-AP-10<br>3B-AP-11<br>3B-AP-21 | 3A-AP-21<br>3B-AP-10<br>3B-AP-11<br>3B-AP-21 | 3A-AP-21<br>3B-AP-10<br>3B-AP-11<br>3B-AP-21 | 3A-AP-13<br>3B-AP-10<br>3B-AP-11 | 3A-AP-13<br>3B-AP-10<br>3B-AP-11 | 3A-AP-21<br>3A-AP-23<br>3B-AP-10<br>3B-AP-11<br>3B-AP-21 |
| **Sample Activity** | Explore multidimension-al arrays. | Explore computational poetry and string manipulation. | Practice creating new multidimensional arrays. | Organize multidimensi-onal arrays. | Write an algorithm to find the maximum value of a multidimension-al array. | Complete a review assignment in Tynker. | Take a quiz to demonstrate your knowledge of 2D arrays. | Apply coding concepts to create a game called "256." |

# AP Computer Science A

## Scope and Sequence

**Grades 9+**

Each unit includes a suggested pacing guide.

Tynker is recognized by the College Board as an endorsed provider of curriculum and professional development for AP® Computer Science A (AP CSA). The AP CSA course is a year-long high school curriculum that's designed to introduce students to the central ideas of computer science and prepare them for the AP CS A Exam. This course includes 10 units, 123 lessons. Here's a link to the AP CSA document: https://apcentral.collegeboard.org/pdf/ap-computer-science-a-course-and-exam-description.pdf?course=ap-computer-science-a.

| | Lesson 9.1 - Introduction to Inheritance | Lesson 9.2 - Extends | Lesson 9.3 - Equals | Lesson 9.4 - Super | Lesson 9.5 - Polymorphism | Lesson 9.6 - Inheritance Review | Lesson 9.7 - Quiz |
|---|---|---|---|---|---|---|---|
| **Big Ideas** | • Modularity **(MOD)** | • Modularity **(MOD)** | • Modularity **(MOD)** | • Modularity **(MOD)** | • Modularity **(MOD)** | • Modularity **(MOD)** | • Modularity **(MOD)** |
| **Computational Thinking Practices** | | | | | | | |
| **Learning Objectives** | MOD-3.B | MOD-3.B<br>MOD-3.B.1<br>MOD-3.B.2<br>MOD-3.B.3<br>MOD-3.B.4<br>MOD-3.E.3<br>MOD-3.E.4 | MOD-3.E.3<br>MOD-3.E.4 | MOD-3<br>MOD-3.B.5<br>MOD-3.B.6<br>MOD-3.B.7<br>MOD-3.B.8<br>MOD-3.B.9<br>MOD-3.B.10<br>MOD-3.B.11<br>MOD-3.B.12<br>MOD-3.B.13<br>MOD-3.B.14<br>MOD-3.B.15 | MOD-3.C<br>MOD-3.C.1<br>MOD-3.C.2<br>MOD-3.C.3<br>MOD-3.C.4<br>MOD-3.D<br>MOD-3.D.1<br>MOD-3.D.2<br>MOD-3.D.3 | MOD-3.B.1 - MOD-3.B.15<br>MOD-3.C.1 - MOD-3.C.4<br>MOD-3.D.1 - MOD-3.D.3<br>MOD-3.E.1 - MOD-3.E.3 | MOD-3.B<br>MOD-3.C.2<br>MOD-3.E |
| **Skills** | | | | | | | |
| **CSTA Computer Science Standards** | 3A-AP-13<br>3B-AP-10<br>3B-AP-11<br>3B-AP-14 | 3A-AP-21<br>3B-AP-10<br>3B-AP-11<br>3B-AP-21 | 3A-AP-21<br>3B-AP-10<br>3B-AP-11<br>3B-AP-21 | 3A-AP-21<br>3B-AP-10<br>3B-AP-11<br>3B-AP-21 | 3A-AP-13<br>3B-AP-10<br>3B-AP-11 | 3A-AP-13<br>3B-AP-10<br>3B-AP-11 | 3A-AP-13<br>3B-AP-10<br>3B-AP-11 |
| **Sample Activity** | Apply Java's syntax to real world class hierarchies. | Represent a class hierarchy of your own using code. | Explore different ways of comparing objects. | Explore constructors and objects. | Explore polymorphism as it relates to inheritance. | Complete a review assessment in Tynker. | Demonstrate your knowledge of inheritance by completing a quiz. |

# AP Computer Science A

**Scope and Sequence**

**Grades 9+**

Each unit includes a suggested pacing guide.

Tynker is recognized by the College Board as an endorsed provider of curriculum and professional development for AP® Computer Science A (AP CSA). The AP CSA course is a year-long high school curriculum that's designed to introduce students to the central ideas of computer science and prepare them for the AP CS A Exam. This course includes 10 units, 123 lessons. Here's a link to the AP CSA document: https://apcentral.collegeboard.org/pdf/ap-computer-science-a-course-and-exam-description.pdf?course=ap-computer-science-a.

| | Lesson 10.1 - Introduction to Recursion | Lesson 10.2 - Recursive Algorithms | Lesson 10.3 - Tracing Recursive Algorithms | Lesson 10.4 - Binary Search | Lesson 10.5 - Merge Sort | Lesson 10.6 - Recursion Practice | Lesson 10.7 - Recursion Review | Lesson 10.8 - Recursion Quiz |
|---|---|---|---|---|---|---|---|---|
| **Big Ideas** | ● Control **(CON)** | ● Control **(CON)** | ● Control **(CON)** | ● Control **(CON)** | ● Control **(CON)** | ● Control **(CON)** | ● Control **(CON)** | |
| **Computational Thinking Practices** | | | | | | | | |
| **Learning Objectives** | CON-2.O<br>CON-2.O.1<br>CON-2.O.2 | CON-2.O<br>CON-2.O.3<br>CON-2.O.6<br>CON-2.P | CON-2.O.4<br>CON-2.O.5 | CON-2.P.1<br>CON-2.P.2<br>CON-2.P.3<br>CON-2.P.4 | CON-2.Q<br>CON-2.Q.1 | CON-2.O.5<br>CON-2.O.6<br>CON-2.P | CON-2.O<br>CON-2.O.1<br>CON-2.O.2<br>CON-2.O.3<br>CON-2.O.4<br>CON-2.O.5<br>CON-2.O.6<br>CON-2.P<br>CON-2.P.1<br>CON-2.P.2<br>CON-2.P.3<br>CON-2.P.4<br>CON-2.Q<br>CON-2.Q.1 | |
| **Skills** | | | | | | | | |
| **CSTA Computer Science Standards** | 3A-AP-21<br>3B-AP-10<br>3B-AP-11<br>3B-AP-21 | 3A-AP-13<br>3B-AP-10<br>3B-AP-11 | 3A-AP-13<br>3B-AP-10<br>3B-AP-11 | 3A-AP-21<br>3B-AP-10<br>3B-AP-11<br>3B-AP-21 | 3A-AP-21<br>3B-AP-10<br>3B-AP-11<br>3B-AP-21 | 3A-AP-15<br>3A-AP-21<br>3B-AP-10<br>3B-AP-11<br>3B-AP-21 | 3A-AP-13<br>3B-AP-10<br>3B-AP-11 | 3A-AP-13<br>3B-AP-10<br>3B-AP-11 |
| **Sample Activity** | Write a "countdown" recursive method with the teacher. | Watch a video showing merge sort in action. | Visually trace several recursive methods. | Implement the binary search algorithm recursively. | Explore how to implement merge sort with Java. | Practice writing recursive programs on your own. | Complete a review assignment in Tynker. | Take a quiz to demonstrate your understanding of recursion. |