

# Space Cadet

## Scope and Sequence

# Grades K-2

Each lesson takes about one class period to complete.

Space Cadet is a course for students in grade K-2 who are new to Tynker. It is available for free on iPads as part of the [Everyone Can Code program](#) from Apple. You can download a free [teacher guide eBook](#) from Apple. The stories, games, puzzles, and projects in this course engage students in developing computational thinking skills, as listed below from the CSTA Level 1 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 -<br>Crash Landed!   | Lesson 2 -<br>Dance Party  | Lesson 3 -<br>Stay the<br>Course   | Lesson 4 -<br>Walk. Jump.<br>Repeat.  | Lesson 5 -<br>Glitchy Code   | Lesson 6 -<br>Asteroids   | Lesson 7 -<br>Shifty Aliens   | Lesson 8 -<br>Blast Off!   |
|--|---|--|--|---|--|---|---|--|
| Key Skills and<br>Concepts               | <ul style="list-style-type: none"> <li>Sequence blocks of code to create algorithms</li> </ul>                                      | <ul style="list-style-type: none"> <li>Create your own sequence</li> <li>Understand the importance of order in sequencing</li> </ul> | <ul style="list-style-type: none"> <li>Construct sequences in different ways</li> <li>Understand there's more than one way to solve a problem</li> </ul> | <ul style="list-style-type: none"> <li>Code with loops</li> <li>Detect patterns in code sequences</li> <li>Identify when to use loops for efficiency</li> </ul> | <ul style="list-style-type: none"> <li>Troubleshoot and debug a program</li> </ul> | <ul style="list-style-type: none"> <li>Trigger code using different events</li> <li>Broadcast and receive messages</li> </ul> | <ul style="list-style-type: none"> <li>Run code only when a condition is true</li> <li>Construct dynamic programs that respond to changing conditions</li> </ul>                        | <ul style="list-style-type: none"> <li>Construct an algorithm to solve a problem</li> <li>Design a simple program</li> </ul> |
| CCSS-Math<br>Standards                   | 1.OA.1, 2.OA.1<br>1.OA.2, 2.OA.2<br>1.OA.3, 2.OA.3<br>1.MD.4  | 1.OA.1, 2.OA.1<br>1.OA.2, 2.OA.2<br>1.OA.3, 2.OA.3<br>1.MD.4   | 1.OA.1, 2.OA.1<br>1.OA.2, 2.OA.2<br>1.OA.3, 2.OA.3<br>1.MD.4   | 1.OA.1, 2.OA.1<br>1.OA.2, 2.OA.2<br>1.OA.3, 2.OA.3<br>1.MD.4  | MP.1<br>5.G.1<br>5.G.2<br>6.NS.6   | MP.1<br>5.G.1<br>5.G.2<br>6.NS.5<br>6.NS.6  | 1.OA.1, 2.OA.1<br>1.OA.2, 2.OA.2<br>1.OA.3, 2.OA.3<br>1.MD.4  | 1.OA.1, 2.OA.1<br>1.OA.2, 2.OA.2<br>1.OA.3, 2.OA.3<br>1.MD.4   |
| CCSS-ELA<br>Standards                    | 1.RI.3, 2.RI.3<br>1.RI.6, 2.RI.6<br>1.RI.7, 2.RI.7<br>1.RI.10, 2.RI.10<br>1.RF.1, 2.RF.1<br>1.RF.4, 2.RF.4<br>1.L.3, 2.L.3<br>2.L.6 | 1.RI.3, 2.RI.3<br>1.RI.6, 2.RI.6<br>1.RI.7, 2.RI.7<br>1.RI.10, 2.RI.10<br>1.RF.1, 2.RF.1<br>1.RF.4, 2.RF.4<br>1.L.3, 2.L.3<br>2.L.6  | 1.RI.3, 2.RI.3<br>1.RI.6, 2.RI.6<br>1.RI.7, 2.RI.7<br>1.RI.10, 2.RI.10<br>1.RF.1, 2.RF.1<br>1.RF.4, 2.RF.4<br>1.L.3, 2.L.3<br>2.L.6                      | 1.RI.3, 2.RI.3<br>1.RI.6, 2.RI.6<br>1.RI.10<br>1.RF.1, 2.RF.1<br>1.RF.4, 2.RF.4<br>1.L.3, 2.L.3<br>2.L.6  | 5.RF.4<br>6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7                                      | 3.RI.3, 4.RI.3<br>3.RI.5<br>3.RI.7, 4.RI.7<br>3.RF.3, 4.RF.3<br>3.RF.4, 4.RF.4<br>3.L.4, 4.L.4                                | 3.W.3, 4.W.3<br>3.W.4, 4.W.4<br>3.W.6, 4.W.6<br>3.L.1, 4.L.1, 3.L.2, 4.L.2<br>3.L.3, 4.L.3, 3.L.4, 4.L.4<br>3.RI.3, 4.RI.3, 3.RI.5<br>3.RI.7, 4.RI.7, 3.RF.3,<br>4.RF.3, 3.RF.4, 4.RF.4 | 1.RI.3, 2.RI.3<br>1.RI.6, 2.RI.6<br>1.RI.10<br>1.RF.1, 2.RF.1<br>1.RF.4, 2.RF.4<br>1.L.3, 2.L.3<br>2.L.6                     |
| CSTA<br>Computer<br>Science<br>Standards | 1B-A-5-4<br>1B-A-3-7  | 1B-A-5-4<br>1B-A-3-7   | 1B-A-5-4<br>1B-A-3-7   | 1B-A-5-4<br>1B-A-3-7  | 1B-A-5-4<br>1B-A-3-7<br>1B-A-6-8   | 1B-A-5-4<br>1B-A-3-7  | 1B-A-5-4<br>1B-A-3-7  | 1B-A-5-4<br>1B-A-3-7   |
| UK National<br>Curriculum                | Keystages 1 & 2<br>Computing**  | Keystages 1 & 2<br>Computing*  | Keystages 1 & 2<br>Computing*  | Keystages 1 & 2<br>Computing*   | Keystages 1 & 2<br>Computing*  | Keystages 1 & 2<br>Computing*   | Keystages 1 & 2<br>Computing*   | Keystages 1 & 2<br>Computing*  |
| Sample<br>Application of<br>Skills       | Use sequencing to solve puzzles.  | Use sequencing to solve puzzles.   | Use custom and unique sequencing to solve puzzles.   | Demonstrate understanding of sequencing and loops.  | Fix the broken code to solve the puzzles.  | Create interactive games using broadcasting and receiving messages between actors.  | Use conditional statements and logic to solve puzzles.  | Use conditional statements and loops to solve puzzles  |

# Dragon Spells

## Scope and Sequence

# Grades 3-5

Each lesson takes about one class period to complete.

Dragon Spells is a course for students in grade 3-5 who are new to programming. It is available for free on iPads as part of the [Everyone Can Code program](#) from Apple. You can download a free [teacher guide iBook](#) from Apple. The stories, games, puzzles, and projects in this course engage students in developing computational thinking skills, as listed below from the CSTA Level 1 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 -<br>Dragon Eggs   | Lesson 2 -<br>Blast Through   | Lesson 3 -<br>Deja Vu  | Lesson 4 -<br>Twisted Trees   | Lesson 5 -<br>Dragon Scrolls   | Lesson 6 -<br>Ancient Spells   | Lesson 7 -<br>Catch Me If<br>You Can  | Lesson 8 -<br>The Long Road  |
|------------------------------------|---|---|--|---|--|--|---|--|
| Key Skills and<br>Concepts         | <ul style="list-style-type: none"> <li>Build a set of commands, forming an algorithm</li> <li>Sequence commands in a specific order to solve a problem</li> </ul> | <ul style="list-style-type: none"> <li>Analyze algorithms to find errors</li> <li>Troubleshoot and debug a program</li> </ul> | <ul style="list-style-type: none"> <li>Detect patterns in code sequences</li> <li>Build more efficient algorithms using loops</li> </ul> | <ul style="list-style-type: none"> <li>Deconstruct problems into manageable parts</li> <li>Assemble parts of a program to solve a larger problem</li> <li>Recognize patterns for decomposition</li> </ul> | <ul style="list-style-type: none"> <li>Group common elements together to create efficient processes in solving problems</li> </ul> | <ul style="list-style-type: none"> <li>Create functions to shorten code</li> <li>Call a function to trigger a set of code</li> </ul> | <ul style="list-style-type: none"> <li>Run code when a condition is true</li> <li>Construct dynamic programs that can respond to changing conditions</li> </ul>                         | <ul style="list-style-type: none"> <li>Repeatedly run a series of commands based on whether certain conditions are true</li> <li>Build programs with loops inside other loops</li> </ul> |
| CCSS-Math<br>Standards             | MP.1  | MP.1<br>5.G.1<br>5.G.2<br>6.NS.6  | MP.1<br>1.OA.1, 2.OA.1<br>1.OA.2, 2.OA.2<br>1.OA.3, 2.OA.3<br>1.MD.4   | MP.1<br>1.OA.1, 2.OA.1<br>1.OA.2, 2.OA.2<br>1.OA.3, 2.OA.3<br>1.MD.4  | MP.1   | MP.1<br>1.OA.1, 2.OA.1<br>1.OA.2, 2.OA.2<br>1.OA.3, 2.OA.3<br>1.MD.4   | MP.1  | MP.1   |
| CCSS-ELA<br>Standards              | 5.RF.4<br>6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7   | 5.RF.4<br>6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7   | 1.RI.3, 2.RI.3<br>1.RI.6, 2RI.6<br>1.RI.10<br>1.RF.1, 2.RF.1<br>1.RF.4, 2.RF.4<br>1.L.3, 2.L.3<br>2.L.6                                  | 1.RI.3, 2.RI.3<br>1.RI.6, 2RI.6<br>1.RI.7, 2RI.7<br>1.RI.10, 2.RI.10<br>1.RF.1, 2.RF.1<br>1.RF.4, 2.RF.4<br>1.L.3, 2.L.3<br>2.L.6   | 1.RI.3, 2.RI.3<br>1.RI.6, 2RI.6<br>1.RI.7, 2RI.7<br>1.RI.10, 2.RI.10<br>1.RF.1, 2.RF.1<br>1.RF.4, 2.RF.4<br>1.L.3, 2.L.3<br>2.L.6  | 1.RI.3, 2.RI.3<br>1.RI.6, 2RI.6<br>1.RI.7, 2RI.7<br>1.RI.10, 2.RI.10<br>1.RF.1, 2.RF.1<br>1.RF.4, 2.RF.4<br>1.L.3, 2.L.3<br>2.L.6    | 3.W.3, 4.W.3<br>3.W.4, 4.W.4<br>3.W.6, 4.W.6<br>3.L.1, 4.L.1, 3.L.2, 4.L.2<br>3.L.3, 4.L.3, 3.L.4, 4.L.4<br>3.RI.3, 4.RI.3, 3.RI.5<br>3.RI.7, 4.RI.7, 3.RF.3,<br>4.RF.3, 3.RF.4, 4.RF.4 | 1.RI.3, 2.RI.3<br>1.RI.6, 2RI.6<br>1.RI.7, 2RI.7<br>1.RI.10, 2.RI.10<br>1.RF.1, 2.RF.1<br>1.RF.4, 2.RF.4<br>1.L.3, 2.L.3<br>2.L.6  |
| CSTA<br>Standards                  | 1B-A-5-4<br>1B-A-3-7  | 1B-A-5-4<br>1B-A-3-7<br>1B-A-6-8  | 1B-A-5-4<br>1B-A-3-7<br>2-A-5-6  | 1B-A-5-4<br>1B-A-3-7<br>2-A-5-6   | 1B-A-5-4<br>1B-A-3-7   | 1B-A-5-4<br>1B-A-3-7   | 1B-A-5-4<br>1B-A-3-7  | 1B-A-5-4<br>1B-A-3-7   |
| UK National<br>Curriculum          | Keystages 1 & 2<br>Computing*   | Keystages 1 & 2<br>Computing*   | Keystages 1 & 2<br>Computing*  | Keystages 1 & 2<br>Computing*   | Keystages 1 & 2<br>Computing*  | Keystages 1 & 2<br>Computing*  | Keystages 1 & 2<br>Computing*   | Keystages 1 & 2<br>Computing*  |
| Sample<br>Application of<br>Skills | Use sequencing to solve puzzles.  | Fix the broken code to solve the puzzles.   | Demonstrate understanding of sequencing and loops.   | Use simple loops and custom sequencing to solve puzzles.  | Use loops and sequencing .   | Create a Mad Libs-style game.  | Use conditional statements to solve the puzzles.  | Use conditional statements and loops to solve the puzzles.   |

# Dragon Spells

## Scope and Sequence

# Grades 3-5

Each lesson takes about one class period to complete.

Dragon Spells is a course for students in grade 3-5 who are new to programming. It is available for free on iPads as part of the [Everyone Can Code program](#) from Apple. You can download a free [teacher guide iBook](#) from Apple. The stories, games, puzzles, and projects in this course engage students in developing computational thinking skills, as listed below from the CSTA Level 1 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 - Gem Collector   | Lesson 10 - Dragon Maker   |  |  |  |  |  |  |
|---------------------------------|--|--|--|--|--|--|--|--|
| Key Skills and Concepts         | <ul style="list-style-type: none"> <li>• Create a program that tracks a value that can change over time</li> <li>• Build programs that take input and give output</li> </ul> | <ul style="list-style-type: none"> <li>• Craft and modify the visual elements of an app (UI)</li> <li>• Change the experience that a user has on an app (UX)</li> <li>• Customize UI and UX</li> </ul> |  |  |  |  |  |  |
| CCSS-Math Standards             | MP.1<br>MP.2<br>MP.4   | MP.1   |  |  |  |  |  |  |
| CCSS-ELA Standards              | 7.RI.4, 8.RI.4<br>6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7  | 1.RI.3, 2.RI.3<br>1.RI.6, 2RI.6<br>1.RI.7, 2RI.7<br>1.RI.10, 2.RI.10<br>1.RF.1, 2.RF.1<br>1.RF.4, 2.RF.4<br>1.L.3, 2.L.3<br>2.L.6  |  |  |  |  |  |  |
| CSTA Computer Science Standards | 1B-A-5-4<br>1B-A-3-7<br>2-A-5-7  | 1B-A-5-4<br>1B-A-3-7   |  |  |  |  |  |  |
| UK National Curriculum          | Keystages 1 & 2 Computing*   | Keystages 1 & 2 Computing*   |  |  |  |  |  |  |
| Sample Application of Skills    | Create and modify variables to solve puzzles.  | Customize a dragon using intuitive UI buttons.   |  |  |  |  |  |  |

# Programming 1A

## Scope and Sequence

# Grades 1-2

Each lesson takes about one class period to complete.

Programming 1A is a course for students in grade 1 or 2 who are new to Tynker. The puzzles and projects engage students in developing computational thinking skills, as listed below from the CSTA Level 1 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 - Welcome   | Lesson 2 - Connect Code Blocks   | Lesson 3 - Recognize the Pattern   | Lesson 4 - Follow the Path  | Lesson 5 - Sequencing  | Lesson 6 - Conditional Logic   | Lesson 7 - Conditional Loops  | Lesson 8 - Draw Simple Shapes  |
|---------------------------------|--|--|--|---|--|--|---|--|
| Key Skills and Concepts         | <ul style="list-style-type: none"> <li>Learn about Tynker and what you can build with code</li> </ul>    | <ul style="list-style-type: none"> <li>Use simple events</li> <li>Use simple loops</li> </ul>            | <ul style="list-style-type: none"> <li>Use conditional loops</li> <li>Create custom sequences to solve puzzles</li> <li>Recognize patterns to create algorithms</li> </ul> | <ul style="list-style-type: none"> <li>Sequence blocks of code to create algorithms</li> </ul>                                      | <ul style="list-style-type: none"> <li>Use simple events</li> <li>Use simple loops</li> <li>Use repetition to create algorithms</li> </ul> | <ul style="list-style-type: none"> <li>Recognize patterns</li> <li>Use conditional loops</li> <li>Sequence blocks of code</li> </ul> | <ul style="list-style-type: none"> <li>Use simple events</li> <li>Recognize patterns</li> <li>Use conditional loops</li> <li>Sequence blocks of code</li> </ul> | <ul style="list-style-type: none"> <li>Sequence blocks of code</li> <li>Draw geometric shapes and angles using repetition and loops</li> </ul> |
| CCSS-Math Standards             |  | 1.OA.1, 2.OA.1<br>1.OA.2, 2.OA.2<br>1.OA.3, 2.OA.3<br>1.MD.4   | 1.OA.1, 2.OA.1<br>1.OA.2, 2.OA.2<br>1.OA.3, 2.OA.3<br>1.MD.4   | 1.OA.1, 2.OA.1<br>1.OA.2, 2.OA.2<br>1.OA.3, 2.OA.3<br>1.MD.4  | 1.OA.1, 2.OA.1<br>1.OA.2, 2.OA.2<br>1.OA.3, 2.OA.3<br>1.MD.4   | 1.OA.1, 2.OA.1<br>1.OA.2, 2.OA.2<br>1.OA.3, 2.OA.3<br>1.MD.4   | 1.OA.1, 2.OA.1<br>1.OA.2, 2.OA.2<br>1.OA.3, 2.OA.3<br>1.MD.4  | 1.OA.1, 2.OA.1<br>1.OA.2, 2.OA.2<br>1.OA.3, 2.OA.3<br>1.MD.4<br>1.GA.1, 2.GA.1<br>1.GA.2, 2.GA.2   |
| CCSS-ELA Standards              | 1.RI.3, 2.RI.3<br>1.RI.6, 2.RI.6<br>1.RI.10<br>1.RF.1, 2.RF.1<br>1.RF.4, 2.RF.4<br>1.L.3, 2.L.3<br>2.L.6 | 1.RI.3, 2.RI.3<br>1.RI.6, 2.RI.6<br>1.RI.10<br>1.RF.1, 2.RF.1<br>1.RF.4, 2.RF.4<br>1.L.3, 2.L.3<br>2.L.6 | 1.RI.3, 2.RI.3<br>1.RI.6, 2.RI.6<br>1.RI.7, 2.RI.7<br>1.RI.10, 2.RI.10<br>1.RF.1, 2.RF.1<br>1.RF.4, 2.RF.4<br>1.L.3, 2.L.3<br>2.L.6  | 1.RI.3, 2.RI.3<br>1.RI.6, 2.RI.6<br>1.RI.7, 2.RI.7<br>1.RI.10, 2.RI.10<br>1.RF.1, 2.RF.1<br>1.RF.4, 2.RF.4<br>1.L.3, 2.L.3<br>2.L.6 | 1.RI.3, 2.RI.3<br>1.RI.6, 2.RI.6<br>1.RI.7, 2.RI.7<br>1.RI.10, 2.RI.10<br>1.RF.1, 2.RF.1<br>1.RF.4, 2.RF.4<br>1.L.3, 2.L.3<br>2.L.6        | 1.RI.3, 2.RI.3<br>1.RI.6, 2.RI.6<br>1.RI.7, 2.RI.7<br>1.RI.10, 2.RI.10<br>1.RF.1, 2.RF.1<br>1.RF.4, 2.RF.4<br>1.L.3, 2.L.3<br>2.L.6  | 1.RI.3, 2.RI.3<br>1.RI.6, 2.RI.6<br>1.RI.7, 2.RI.7<br>1.RI.10, 2.RI.10<br>1.RF.1, 2.RF.1<br>1.RF.4, 2.RF.4<br>1.L.3, 2.L.3<br>2.L.6                             | 1.RI.3, 2.RI.3<br>1.RI.6, 2.RI.6<br>1.RI.7, 2.RI.7<br>1.RI.10, 2.RI.10<br>1.RF.1, 2.RF.1<br>1.RF.4, 2.RF.4<br>1.L.3, 2.L.3<br>2.L.6            |
| CSTA Computer Science Standards | L1:3.CT.1<br>L1:3.CT.4   | L1:3.CT.1<br>L1:3.CT.4<br>L1:6.CT.1<br>L1:6.CT.2<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6               | L1:3.CT.1<br>L1:3.CT.4<br>L1:6.CT.1<br>L1:6.CT.2<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6   | L1:3.CT.1<br>L1:3.CT.4<br>L1:6.CT.1<br>L1:6.CT.2<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6  | L1:3.CT.1<br>L1:3.CT.4<br>L1:6.CT.1<br>L1:6.CT.2<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6   | L1:3.CT.1<br>L1:3.CT.4<br>L1:6.CT.1<br>L1:6.CT.2<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6   | L1:3.CT.1<br>L1:3.CT.4<br>L1:6.CT.1<br>L1:6.CT.2<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6  | L1:3.CT.1<br>L1:3.CT.4<br>L1:6.CT.1<br>L1:6.CT.2<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6   |
| UK National Curriculum          | Coming Soon  | Coming Soon  | Coming Soon  | Coming Soon   | Coming Soon  | Coming Soon  | Coming Soon   | Coming Soon  |
| Sample Application of Skills    | Understand how to use visual blocks to program algorithms to solve puzzles.                              | Demonstrate understanding of sequencing and loops.   | Use sequencing and repetition to solve puzzles.  | Use sequencing and conditional loops to solve puzzles.  | Use sequencing and repetition to solve puzzles.  | Use sequencing, conditional logic, conditional loops, and repetition to solve puzzles.   | Use sequencing, conditional loops, and repetition to solve puzzles.   | Draw simple shapes and angles using sequencing, repetition, and loops.   |

# Programming 1A

# Grades 1-2

## Scope and Sequence

Each lesson takes about one class period to complete.

Programming 1A is a course for students in grade 1 or 2 who are new to Tynker. The puzzles and projects engage students in developing computational thinking skills, as listed below from the CSTA Level 1 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 - Squares, Triangles, and Staircases  | Lesson 10 - Sequencing Review   | Lesson 11 - Use Repeat Loops  | Lesson 12 - Complete Multiple Tasks in Order   |  |  |  |  |  |
|---------------------------------|--|---|---|--|--|--|--|--|--|
| Key Skills and Concepts         | <ul style="list-style-type: none"> <li>Use simple loops</li> <li>Use simple events</li> <li>Draw geometric shapes with repetition and loops</li> </ul> | <ul style="list-style-type: none"> <li>Use sequencing concepts</li> <li>Use simple loops</li> <li>Recognize patterns</li> </ul>     | <ul style="list-style-type: none"> <li>Use simple loops</li> <li>Sequence code blocks to program algorithms</li> </ul>              | <ul style="list-style-type: none"> <li>Use functions to handle sub-tasks</li> <li>Call functions</li> <li>Use conditional loops</li> </ul> |  |  |  |  |  |
| CCSS-Math Standards             | 1.OA.1, 2.OA.1<br>1.OA.2, 2.OA.2<br>1.OA.3, 2.OA.3<br>1.MD.4<br>1.GA.1, 2.GA.1<br>1.GA.2, 2.GA.2   | 1.OA.1, 2.OA.1<br>1.OA.2, 2.OA.2<br>1.OA.3, 2.OA.3<br>1.MD.4  | 1.OA.1, 2.OA.1<br>1.OA.2, 2.OA.2<br>1.OA.3, 2.OA.3<br>1.MD.4  | 1.OA.1, 2.OA.1<br>1.OA.2, 2.OA.2<br>1.OA.3, 2.OA.3<br>1.MD.4   |  |  |  |  |  |
| CCSS-ELA Standards              | 1.RI.3, 2.RI.3<br>1.RI.6, 2.RI.6<br>1.RI.7, 2.RI.7<br>1.RI.10, 2.RI.10<br>1.RF.1, 2.RF.1<br>1.RF.4, 2.RF.4<br>1.L.3, 2.L.3<br>2.L.6                    | 1.RI.3, 2.RI.3<br>1.RI.6, 2.RI.6<br>1.RI.7, 2.RI.7<br>1.RI.10, 2.RI.10<br>1.RF.1, 2.RF.1<br>1.RF.4, 2.RF.4<br>1.L.3, 2.L.3<br>2.L.6 | 1.RI.3, 2.RI.3<br>1.RI.6, 2.RI.6<br>1.RI.7, 2.RI.7<br>1.RI.10, 2.RI.10<br>1.RF.1, 2.RF.1<br>1.RF.4, 2.RF.4<br>1.L.3, 2.L.3<br>2.L.6 | 1.RI.3, 2.RI.3<br>1.RI.6, 2.RI.6<br>1.RI.7, 2.RI.7<br>1.RI.10, 2.RI.10<br>1.RF.1, 2.RF.1<br>1.RF.4, 2.RF.4<br>1.L.3, 2.L.3<br>2.L.6        |  |  |  |  |  |
| CSTA Computer Science Standards | L1:3.CT.1<br>L1:3.CT.4<br>L1:6.CT.1<br>L1:6.CT.2<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6   | L1:3.CT.1<br>L1:3.CT.4<br>L1:6.CT.1<br>L1:6.CT.2<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6  | L1:3.CT.1<br>L1:3.CT.4<br>L1:6.CT.1<br>L1:6.CT.2<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6  | L1:3.CT.1<br>L1:3.CT.4<br>L1:6.CT.1<br>L1:6.CT.2<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6   |  |  |  |  |  |
| UK National Curriculum          | Coming Soon  | Coming Soon   | Coming Soon   | Coming Soon  |  |  |  |  |  |
| Sample Application of Skills    | Draw various shapes using sequencing, repetition and loops.  | Use sequencing and repetition to solve puzzles.   | Use sequencing, repetition, and loops to solve puzzles.   | Use functions, sequencing, and loops to solve puzzles.   |  |  |  |  |  |

# Programming 1B

# Grades 1-2

## Scope and Sequence

Each lesson takes about one class period to complete.

Programming 1B is a course for students in grade 1 or 2 who are new to Tynker. The puzzles and projects engage students in developing computational thinking skills, as listed below from the CSTA Level 1 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 - Welcome   | Lesson 2 - Use Conditional Logic  | Lesson 3 - Apply Advanced Logic   | Lesson 4 - Animation and Sound  | Lesson 5 - Apply Negative Logic  | Lesson 6 - Use Nested Logic   | Lesson 7 - Use Advanced Logic   | Lesson 8 - Write a Story   |   |
|---------------------------------|--|---|---|---|--|---|---|--|---|
| Key Skills and Concepts         | <ul style="list-style-type: none"> <li>Learn about Tynker and what you can build with code</li> </ul>    | <ul style="list-style-type: none"> <li>Use simple events</li> <li>Use simple loops</li> <li>Use negative logic</li> </ul> | <ul style="list-style-type: none"> <li>Use conditional loops</li> <li>Create custom sequences to solve puzzles</li> </ul>           | <ul style="list-style-type: none"> <li>Use simple events</li> <li>Animate characters using sounds and events</li> </ul>             | <ul style="list-style-type: none"> <li>Use simple events</li> <li>Use simple loops</li> <li>Use repetition to create algorithms</li> <li>Use negative logic</li> </ul> | <ul style="list-style-type: none"> <li>Recognize patterns</li> <li>Use conditional loops</li> <li>Use nested logic</li> </ul>       | <ul style="list-style-type: none"> <li>Use simple events</li> <li>Recognize patterns</li> <li>Use conditional loops</li> <li>Sequence blocks of code</li> </ul> | <ul style="list-style-type: none"> <li>Sequence blocks of code</li> <li>Animate characters using sounds, dialogue, and events</li> </ul> |   |
| CCSS-Math Standards             |  | 1.OA.1, 2.OA.1<br>1.OA.2, 2.OA.2<br>1.OA.3, 2.OA.3<br>1.MD.4  | 1.OA.1, 2.OA.1<br>1.OA.2, 2.OA.2<br>1.OA.3, 2.OA.3<br>1.MD.4  | 1.MD.4  | 1.OA.1, 2.OA.1<br>1.OA.2, 2.OA.2<br>1.OA.3, 2.OA.3<br>1.MD.4   | 1.OA.1, 2.OA.1<br>1.OA.2, 2.OA.2<br>1.OA.3, 2.OA.3<br>1.MD.4  | 1.OA.1, 2.OA.1<br>1.OA.2, 2.OA.2<br>1.OA.3, 2.OA.3<br>1.MD.4  | 1.MD.4   |   |
| CCSS-ELA Standards              | 1.RI.3, 2.RI.3<br>1.RI.6, 2.RI.6<br>1.RI.10<br>1.RF.1, 2.RF.1<br>1.RF.4, 2.RF.4<br>1.L.3, 2.L.3<br>2.L.6 | 1.RI.3, 2.RI.3<br>1.RI.6, 2.RI.6<br>1.RI.10<br>1.RF.1, 2.RF.1<br>1.RF.4, 2.RF.4<br>1.L.3, 2.L.3<br>2.L.6                  | 1.RI.3, 2.RI.3<br>1.RI.6, 2.RI.6<br>1.RI.7, 2.RI.7<br>1.RI.10, 2.RI.10<br>1.RF.1, 2.RF.1<br>1.RF.4, 2.RF.4<br>1.L.3, 2.L.3<br>2.L.6 | 1.RI.3, 2.RI.3<br>1.RI.6, 2.RI.6<br>1.RI.7, 2.RI.7<br>1.RI.10, 2.RI.10<br>1.RF.1, 2.RF.1<br>1.RF.4, 2.RF.4<br>1.L.3, 2.L.3<br>2.L.6 | 1.RI.3, 2.RI.3<br>1.RI.6, 2.RI.6<br>1.RI.7, 2.RI.7<br>1.RI.10, 2.RI.10<br>1.RF.1, 2.RF.1<br>1.RF.4, 2.RF.4<br>1.L.3, 2.L.3<br>2.L.6                                    | 1.RI.3, 2.RI.3<br>1.RI.6, 2.RI.6<br>1.RI.7, 2.RI.7<br>1.RI.10, 2.RI.10<br>1.RF.1, 2.RF.1<br>1.RF.4, 2.RF.4<br>1.L.3, 2.L.3<br>2.L.6 | 1.RI.3, 2.RI.3<br>1.RI.6, 2.RI.6<br>1.RI.7, 2.RI.7<br>1.RI.10, 2.RI.10<br>1.RF.1, 2.RF.1<br>1.RF.4, 2.RF.4<br>1.L.3, 2.L.3<br>2.L.6                             | 1.RI.3, 2.RI.3<br>1.RI.6, 2.RI.6<br>1.RI.7, 2.RI.7<br>1.RI.10, 2.RI.10<br>1.RF.1, 2.RF.1<br>1.RF.4, 2.RF.4<br>1.L.3, 2.L.3<br>2.L.6      | 1.RI.3, 2.RI.3<br>1.RI.6, 2.RI.6<br>1.RI.7, 2.RI.7<br>1.RI.10, 2.RI.10<br>1.RF.1, 2.RF.1<br>1.RF.4, 2.RF.4<br>1.L.3, 2.L.3<br>2.L.6 |
| CSTA Computer Science Standards | L1:3.CT.1<br>L1:3.CT.4   | L1:3.CT.1<br>L1:3.CT.4<br>L1:6.CT.1<br>L1:6.CT.2<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6                                | L1:3.CT.1<br>L1:3.CT.4<br>L1:6.CT.1<br>L1:6.CT.2<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6  | L1:3.CT.1<br>L1:3.CT.4<br>L1:6.CT.1<br>L1:6.CT.2<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6  | L1:3.CT.1<br>L1:3.CT.4<br>L1:6.CT.1<br>L1:6.CT.2<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6   | L1:3.CT.1<br>L1:3.CT.4<br>L1:6.CT.1<br>L1:6.CT.2<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6  | L1:3.CT.1<br>L1:3.CT.4<br>L1:6.CT.1<br>L1:6.CT.2<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6  | L1:3.CT.1<br>L1:3.CT.4<br>L1:6.CT.1<br>L1:6.CT.2<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6   |   |
| UK National Curriculum          | Coming Soon  | Coming Soon   | Coming Soon   | Coming Soon   | Coming Soon  | Coming Soon   | Coming Soon   | Coming Soon  |   |
| Sample Application of Skills    | Understand how to use visual blocks to program algorithms to solve puzzles.                              | Use repetitive sequencing and conditional logic to solve puzzles.   | Use conditional logic and loops to solve puzzles.   | Create an animation of a character interacting with their surroundings.   | Use sequencing and repetition to solve puzzles.  | Use sequencing, conditional logic, conditional loops, and repetition to solve puzzles.  | Use sequencing, conditional loops, and repetition to solve puzzles.   | Create an animation of a character interacting with their surroundings.  |   |

# Programming 1B

# Grades 1-2

## Scope and Sequence

Each lesson takes about one class period to complete.

Programming 1B is a course for students in grade 1 or 2 who are new to Tynker. The puzzles and projects engage students in developing computational thinking skills, as listed below from the CSTA Level 1 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 -<br>Make<br>Geometric<br>Patterns   | Lesson 10 -<br>Use Arrow<br>Keys to Move  | Lesson 11 -<br>Build a Game   | Lesson 12 -<br>Sequencing<br>Review   | Lesson 13 -<br>Loops Review   | Lesson 14 -<br>Patterns<br>Review   |   |  |  |
|--|---|---|---|---|---|---|---|--|--|
| Key Skills and<br>Concepts               | <ul style="list-style-type: none"> <li>Sequence blocks of code</li> <li>Draw geometric shapes using repetition and loops</li> </ul> | <ul style="list-style-type: none"> <li>Use sequencing concepts</li> <li>Animate characters using motion, sound, and events</li> </ul> | <ul style="list-style-type: none"> <li>Use simple events and loops</li> <li>Animate characters using motion, sound, and events</li> </ul> | <ul style="list-style-type: none"> <li>Use sequencing concepts</li> <li>Use simple loops</li> <li>Recognize patterns</li> </ul>   | <ul style="list-style-type: none"> <li>Use simple loops</li> <li>Sequence code blocks to program algorithms</li> </ul>            | <ul style="list-style-type: none"> <li>Recognize patterns</li> <li>Sequence code blocks to program algorithms</li> </ul>          |   |  |  |
| CCSS-Math<br>Standards                   | 1.OA.1, 2.OA.1<br>1.OA.2, 2.OA.2<br>1.OA.3, 2.OA.3<br>1.MD.4<br>1.GA.1, 2.GA.1<br>1.GA.2, 2.GA.2                                    | 1.MD.4  | 1.MD.4  | 1.OA.1, 2.OA.1<br>1.OA.2, 2.OA.2<br>1.OA.3, 2.OA.3<br>1.MD.4  | 1.OA.1, 2.OA.1<br>1.OA.2, 2.OA.2<br>1.OA.3, 2.OA.3<br>1.MD.4  | 1.OA.1, 2.OA.1<br>1.OA.2, 2.OA.2<br>1.OA.3, 2.OA.3<br>1.MD.4  |   |  |  |
| CCSS-ELA<br>Standards                    | 1.RI.3, 2.RI.3<br>1.RI.6, 2RI.6<br>1.RI.7, 2RI.7<br>1.RI.10, 2.RI.10<br>1.RF.1, 2.RF.1<br>1.RF.4, 2.RF.4<br>1.L.3, 2.L.3<br>2.L.6   | 1.RI.3, 2.RI.3<br>1.RI.6, 2RI.6<br>1.RI.7, 2RI.7<br>1.RI.10, 2.RI.10<br>1.RF.1, 2.RF.1<br>1.RF.4, 2.RF.4<br>1.L.3, 2.L.3<br>2.L.6     | 1.RI.3, 2.RI.3<br>1.RI.6, 2RI.6<br>1.RI.7, 2RI.7<br>1.RI.10, 2.RI.10<br>1.RF.1, 2.RF.1<br>1.RF.4, 2.RF.4<br>1.L.3, 2.L.3<br>2.L.6         | 1.RI.3, 2.RI.3<br>1.RI.6, 2RI.6<br>1.RI.7, 2RI.7<br>1.RI.10, 2.RI.10<br>1.RF.1, 2.RF.1<br>1.RF.4, 2.RF.4<br>1.L.3, 2.L.3<br>2.L.6 | 1.RI.3, 2.RI.3<br>1.RI.6, 2RI.6<br>1.RI.7, 2RI.7<br>1.RI.10, 2.RI.10<br>1.RF.1, 2.RF.1<br>1.RF.4, 2.RF.4<br>1.L.3, 2.L.3<br>2.L.6 | 1.RI.3, 2.RI.3<br>1.RI.6, 2RI.6<br>1.RI.7, 2RI.7<br>1.RI.10, 2.RI.10<br>1.RF.1, 2.RF.1<br>1.RF.4, 2.RF.4<br>1.L.3, 2.L.3<br>2.L.6 | 1.RI.3, 2.RI.3<br>1.RI.6, 2RI.6<br>1.RI.7, 2RI.7<br>1.RI.10, 2.RI.10<br>1.RF.1, 2.RF.1<br>1.RF.4, 2.RF.4<br>1.L.3, 2.L.3<br>2.L.6 |  |  |
| CSTA<br>Computer<br>Science<br>Standards | L1:3.CT.1<br>L1:3.CT.4<br>L1:6.CT.1<br>L1:6.CT.2<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6  | L1:3.CT.1<br>L1:3.CT.4<br>L1:6.CT.1<br>L1:6.CT.2<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6  | L1:3.CT.1<br>L1:3.CT.4<br>L1:6.CT.1<br>L1:6.CT.2<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6  | L1:3.CT.1<br>L1:3.CT.4<br>L1:6.CT.1<br>L1:6.CT.2<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6  | L1:3.CT.1<br>L1:3.CT.4<br>L1:6.CT.1<br>L1:6.CT.2<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6  | L1:3.CT.1<br>L1:3.CT.4<br>L1:6.CT.1<br>L1:6.CT.2<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6  | L1:3.CT.1<br>L1:3.CT.4<br>L1:6.CT.1<br>L1:6.CT.2<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6  |  |  |
| UK National<br>Curriculum                | Coming Soon   | Coming Soon   | Coming Soon   | Coming Soon   | Coming Soon   | Coming Soon   |   |  |  |
| Sample<br>Application of<br>Skills       | Draw various shapes using sequencing, repetition and loops.   | Create an animation of a rocket moving through outer space.   | Create a game with moving characters interacting with one another.  | Use sequencing and repetition to solve puzzles.   | Use sequencing, repetition, and loops to solve puzzles.   | Use sequencing, repetition, patterns, and loops to solve puzzles.   |   |  |  |

# Programming 101

# Grades 3-4

## Scope and Sequence

Each lesson takes about one class period to complete.

Programming 101 is a course for students in grade 3 or 4 who are new to Tynker. The stories, games, puzzles, and projects engage students in developing computational thinking skills, as listed below from the CSTA Level 1 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 - Tynker Workshop  | Lesson 3 - Vector Drawing   | Lesson 4 - Animation  | Lesson 5 - Storytelling   | Lesson 6 - Keyboard Control   | Lesson 7 - Slideshow   | Lesson 8 - Character Creator  |
|---------------------------------|--|---|---|---|---|---|--|---|
| Key Skills and Concepts         | <ul style="list-style-type: none"> <li>Use visual code blocks to create a program</li> <li>Sequence steps</li> <li>Use loops for repetition</li> <li>Use conditional statements and branching logic</li> </ul> | <ul style="list-style-type: none"> <li>Use characters, sounds, and scenes from the Media Library</li> <li>Use simple loops</li> <li>Use delays</li> </ul> | <ul style="list-style-type: none"> <li>Use Tynker drawing tools</li> </ul>                  | <ul style="list-style-type: none"> <li>Animate characters with simple costume handling</li> </ul> | <ul style="list-style-type: none"> <li>Use speech bubbles to make characters have a conversation</li> <li>Use different kinds of speech bubbles and delays to convey meaning and emotion</li> </ul> | <ul style="list-style-type: none"> <li>Use keyboard controls to turn, point, and move characters</li> </ul> | <ul style="list-style-type: none"> <li>Create slideshow presentations in Tynker</li> </ul>     | <ul style="list-style-type: none"> <li>Build and customize an animated character</li> <li>Set character parts and use advanced animation</li> </ul> |
| CCSS-Math Standards             | -  | -   | -   | -   | -   | 4.MD.5  | -  | 4.NF.C.7  |
| CCSS-ELA Standards              | -  | 3.RI.3, 4.RI.3<br>3.RI.5<br>3.RI.7, 4.RI.7<br>3.RF.3, 4.RF.3<br>3.RF.4, 4.RF.4<br>3.L.4, 4.L.4  | -   | 3.RI.3, 4.RI.3<br>3.RI.5<br>3.RI.7, 4.RI.7<br>3.RF.3, 4.RF.3<br>3.RF.4, 4.RF.4<br>3.L.4, 4.L.4    | 3.RI.3, 4.RI.3<br>3.RI.5<br>3.RI.7, 4.RI.7<br>3.RF.3, 4.RF.3<br>3.RF.4, 4.RF.4<br>3.W.3, 4.W.3<br>3.W.4, 4.W.4<br>3.L.1, 4.L.1<br>3.L.2, 4.L.2<br>3.L.3, 4.L.3<br>3.L.4, 4.L.4                      | 3.RI.3, 4.RI.3<br>3.RI.5<br>3.RI.7, 4.RI.7<br>3.RF.3, 4.RF.3<br>3.RF.4, 4.RF.4<br>3.L.4, 4.L.4              | 3.RI.3, 4.RI.3<br>3.RI.5<br>3.RI.7, 4.RI.7<br>3.RF.3, 4.RF.3<br>3.RF.4, 4.RF.4<br>3.L.4, 4.L.4 | 3.RI.3, 4.RI.3<br>3.RI.5<br>3.RI.7, 4.RI.7<br>3.RF.3, 4.RF.3<br>3.RF.4, 4.RF.4<br>3.L.4, 4.L.4  |
| CSTA Computer Science Standards | L1:3.CT.1<br>L1:3.CT.4<br>L1:6.CT.1<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6  | L1:3.CT.1<br>L1:3.CT.4<br>L1:6.CT.1<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6<br>L1:6.CPP.8   | L1:3.CT.1<br>L1:3.CT.4<br>L1:6.CT.1<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6<br>L1:6.CPP.8 | L1:3.CT.1<br>L1:3.CT.4<br>L1:6.CT.1<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6<br>L1:6.CPP.8       | L1:3.CT.1<br>L1:3.CT.4<br>L1:6.CT.1<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6<br>L1:6.CPP.8   | L1:3.CT.1<br>L1:3.CT.4<br>L1:6.CT.1<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6<br>L1:6.CPP.8                 | L1:3.CT.1<br>L1:3.CT.4<br>L1:6.CT.1<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6<br>L1:6.CPP.8    | L1:3.CT.1<br>L1:3.CT.4<br>L1:6.CT.1<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6<br>L1:6.CPP.8   |
| UK National Curriculum          | Keystages 1 & 2 Computing*   | Keystages 1 & 2 Computing*  | Keystages 1 & 2 Computing*  | Keystages 1 & 2 Computing*  | Keystages 1 & 2 Computing*  | Keystages 1 & 2 Computing*  | Keystages 1 & 2 Computing*   | Keystages 1 & 2 Computing*  |
| Sample Application of Skills    | Create a program using simple conditionals and loops to move a character through an obstacle course.   | Create a scene and a character and sounds to accompany them.  | Use drawing tools to draw a scene and some characters.                                      | Create an animation of a bird flying across the screen.   | Create a comic-book style conversation between characters and have them tell jokes and stories.   | Create unique controls to guide a robot through a maze, or a car through city streets.                      | Create a slideshow presentation with pictures and captions..                                   | Create an Epic Quest game, with a Hero that can animate to obtain a reward and avoid an enemy.  |



# Programming 101

# Grades 3-4

## Scope and Sequence

Each lesson takes about one class period to complete.

Programming 101 is a course for students in grade 3 or 4 who are new to Tynker. The stories, games, puzzles, and projects engage students in developing computational thinking skills, as listed below from the CSTA Level 1 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 -<br>Make a<br>Birthday Card  | Lesson 10 -<br>The Music<br>Machine  | Lesson 11 -<br>Positioning<br>Actors  | Lesson 12 -<br>Pen Drawing  | Lesson 13 -<br>TynkerBlocks  | Lesson 14 -<br>Keeping Score   | Lesson 15 -<br>Adding Logic   | Lesson 16 -<br>Quiz Game   |  |
|--|--|--|---|---|--|--|---|--|--|
| Key Skills and<br>Concepts               | <ul style="list-style-type: none"> <li>Make characters and objects appear and disappear</li> <li>Apply special graphics effects</li> </ul> | <ul style="list-style-type: none"> <li>Generate music using musical notes, various instruments, and changing tempos</li> </ul> | <ul style="list-style-type: none"> <li>Move characters and objects to specific screen locations using x- and y-values</li> <li>Move characters left and right using changes in x-values, and up and down using changes in y-values</li> </ul> | <ul style="list-style-type: none"> <li>Make characters draw using the Pen commands</li> <li>Make multiple copies of characters and objects</li> </ul> | <ul style="list-style-type: none"> <li>Use simple and advanced messaging</li> <li>Use special pre-programmed TynkerBlocks</li> </ul> | <ul style="list-style-type: none"> <li>Use a special TynkerBlock to keep track of score</li> <li>Generate a random number from a range of numbers</li> <li>Make characters respond to messages they receive</li> </ul> | <ul style="list-style-type: none"> <li>Use conditional statements and comparison operators</li> </ul> | <ul style="list-style-type: none"> <li>Use more advanced conditional statements and branching logic</li> </ul> |  |
| CCSS-Math<br>Standards                   | -  | -  | 5.G.1, 5.G.2<br>6.NS.5, 6.NS.6  | -   | -  | 5.G.1, 5.G.2<br>6.NS.5, 6.NS.6   | 2.NBT.4<br>5.G.1, 5.G.2<br>6.NS.5, 6.NS.6   | -  |  |
| CCSS-ELA<br>Standards                    | 3.RI.3, 4.RI.3<br>3.RI.5<br>3.RI.7, 4.RI.7<br>3.RF.3, 4.RF.3<br>3.RF.4, 4.RF.4<br>3.L.4, 4.L.4   | 3.RI.3, 4.RI.3<br>3.RI.5<br>3.RI.7, 4.RI.7<br>3.RF.3, 4.RF.3<br>3.RF.4, 4.RF.4<br>3.L.4, 4.L.4                                 | 3.RI.3, 4.RI.3<br>3.RI.5<br>3.RI.7, 4.RI.7<br>3.RF.3, 4.RF.3<br>3.RF.4, 4.RF.4<br>3.L.4, 4.L.4  | 3.RI.3, 4.RI.3<br>3.RI.5<br>3.RI.7, 4.RI.7<br>3.RF.3, 4.RF.3<br>3.RF.4, 4.RF.4<br>3.L.4, 4.L.4  | 3.RI.3, 4.RI.3<br>3.RI.5<br>3.RI.7, 4.RI.7<br>3.RF.3, 4.RF.3<br>3.RF.4, 4.RF.4<br>3.L.4, 4.L.4                                       | 3.RI.3, 4.RI.3<br>3.RI.5<br>3.RI.7, 4.RI.7<br>3.RF.3, 4.RF.3<br>3.RF.4, 4.RF.4<br>3.L.4, 4.L.4   | 3.RI.3, 4.RI.3<br>3.RI.5<br>3.RI.7, 4.RI.7<br>3.RF.3, 4.RF.3<br>3.RF.4, 4.RF.4<br>3.L.4, 4.L.4        | 3.RI.3, 4.RI.3<br>3.RI.5<br>3.RI.7, 4.RI.7<br>3.RF.3, 4.RF.3<br>3.RF.4, 4.RF.4<br>3.L.4, 4.L.4                 | 3.W.3, 4.W.3<br>3.W.4, 4.W.4<br>3.W.6, 4.W.6<br>3.L.1, 4.L.1, 3.L.2, 4.L.2<br>3.L.3, 4.L.3, 3.L.4, 4.L.4<br>3.RI.3, 4.RI.3, 3.RI.5<br>3.RI.7, 4.RI.7, 3.RF.3, 4.RF.3, 3.RF.4, 4.RF.4 |
| CSTA<br>Computer<br>Science<br>Standards | L1:3.CT.1<br>L1:3.CT.4<br>L1:6.CT.1<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6<br>L1:6.CPP.8  | L1:3.CT.1<br>L1:3.CT.4<br>L1:6.CT.1<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6<br>L1:6.CPP.8                                    | L1:3.CT.1<br>L1:3.CT.4<br>L1:6.CT.1<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6<br>L1:6.CPP.8   | L1:3.CT.1<br>L1:3.CT.4<br>L1:6.CT.1<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6<br>L1:6.CPP.8   | L1:3.CT.1<br>L1:3.CT.4<br>L1:6.CT.1<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6<br>L1:6.CPP.8  | L1:3.CT.1<br>L1:3.CT.4<br>L1:6.CT.1<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6<br>L1:6.CPP.8  | L1:3.CT.1<br>L1:3.CT.4<br>L1:6.CT.1<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6<br>L1:6.CPP.8           | L1:3.CT.1<br>L1:3.CT.4<br>L1:6.CT.1<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6<br>L1:6.CPP.8                    | L1:3.CT.1<br>L1:3.CT.4<br>L1:6.CT.1<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6<br>L1:6.CPP.8  |
| UK National<br>Curriculum                | Keystages 1 & 2<br>Computing*  | Keystages 1 & 2<br>Computing*  | Keystages 1 & 2<br>Computing*   | Keystages 1 & 2<br>Computing*   | Keystages 1 & 2<br>Computing*  | Keystages 1 & 2<br>Computing*  | Keystages 1 & 2<br>Computing*   | Keystages 1 & 2<br>Computing*  |  |
| Sample<br>Application of<br>Skills       | Create a birthday card featuring characters that appear and disappear, and special graphic effects.  | Create a piano and a drum machine.   | Create a game with falling characters that need to be caught.   | Create an Etch-a-Sketch® style drawing machine.   | Create a basic Space Invaders style video game.  | Create a game where characters appear and disappear in random places, and score is kept.   | Create an Actor who makes different comments and changes costumes depending on the scene.             | Develop an interactive story where the user provides input on which path the story will take.                  |  |

# Programming 101

# Grades 3-4

## Scope and Sequence

Each lesson takes about one class period to complete.

Programming 101 is a course for students in grade 3 or 4 who are new to Tynker. The stories, games, puzzles, and projects engage students in developing computational thinking skills, as listed below from the CSTA Level 1 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 17 -<br>Color Sensing   |  |  |  |  |  |  |  |
|--|--|--|--|--|--|--|--|--|
| <b>Key Skills and Concepts</b>         | <ul style="list-style-type: none"> <li>Make characters respond to touching certain colors</li> </ul> |  |  |  |  |  |  |  |
| <b>CCSS-Math Standards</b>             | 2.NBT.4<br>5.G.1<br>5.G.2<br>6.NS.5<br>6.NS.6 (not all these)  |  |  |  |  |  |  |  |
| <b>CCSS-ELA Standards</b>              | 3.RI.3, 4.RI.3<br>3.RI.5<br>3.RI.7, 4.RI.7<br>3.RF.3, 4.RF.3<br>3.RF.4, 4.RF.4<br>3.L.4, 4.L.4       |  |  |  |  |  |  |  |
| <b>CSTA Computer Science Standards</b> | L1:3.CT.1<br>L1:3.CT.4<br>L1:6.CT.1<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6<br>L1:6.CPP.8          |  |  |  |  |  |  |  |
| <b>UK National Curriculum</b>          | Keystages 1 & 2 Computing*   |  |  |  |  |  |  |  |
| <b>Sample Application of Skills</b>    | Create a game where characters need to navigate through a maze and avoid enemies.                    |  |  |  |  |  |  |  |

# Programming 102

# Grades 3-4

## Scope and Sequence

Each lesson takes about one class period to complete.

Programming 102 is a course for students in grade 3 or 4 who have completed Tynker’s Programming 101 course. The stories, games, puzzles, and projects engage students in developing computational thinking skills, as listed below from the CSTA Level 1 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 -<br>Review:<br>Interactive<br>Animation   | Lesson 2 -<br>Review:<br>Pen Drawing   | Lesson 3 -<br>Review:<br>Ghost-Catcher   | Lesson 4 -<br>Review:<br>Character<br>Creator   | Lesson 5 -<br>Show and Hide  | Lesson 6 -<br>Layers   | Lesson 7 -<br>Motion  | Lesson 8 -<br>Snowball Siege   |
|---------------------------------|---|--|--|---|--|--|---|--|
| Key Skills and Concepts         | <ul style="list-style-type: none"> <li>Review how to add Actors, add Costumes, animate, move, handle key-press events, use loops for repetition, use delays, and play sounds</li> </ul> | <ul style="list-style-type: none"> <li>Review messaging techniques and usage of the Pen drawing tools</li> </ul> | <ul style="list-style-type: none"> <li>Review detection of color and touch</li> <li>Review making characters appear and disappear</li> </ul> | <ul style="list-style-type: none"> <li>Review basic and advanced animation methods</li> <li>Review usage of conditional statements</li> </ul> | <ul style="list-style-type: none"> <li>Review moving characters and objects to random screen locations using random number generators</li> </ul> | <ul style="list-style-type: none"> <li>Layer characters and objects in front of or behind each other on the Stage</li> </ul> | <ul style="list-style-type: none"> <li>Make a character follow the cursor around the stage</li> <li>Use math operators in conditional statements</li> </ul> | <ul style="list-style-type: none"> <li>Glide smoothly to a specific location</li> <li>Control script flow</li> <li>Broadcast and receive messages</li> </ul> |
| CCSS-Math Standards             | MP.1<br>5.G.1<br>5.G.2<br>6.NS.5<br>6.NS.6  | MP.1   | MP.1   | MP.1  | MP.1<br>5.G.1<br>5.G.2<br>6.NS.5<br>6.NS.6   | MP.1   | MP.1<br>2.OA.1<br>5.G.1<br>5.G.2<br>6.NS.5<br>6.NS.6  | MP.1<br>5.G.1<br>5.G.2<br>6.NS.5<br>6.NS.6   |
| CCSS-ELA Standards              | -   | 3.RI.3, 4.RI.3<br>3.RI.5<br>3.RI.7, 4.RI.7<br>3.RF.3, 4.RF.3<br>3.RF.4, 4.RF.4<br>3.L.4, 4.L.4                   | 3.RI.3, 4.RI.3<br>3.RI.5<br>3.RI.7, 4.RI.7<br>3.RF.3, 4.RF.3<br>3.RF.4, 4.RF.4<br>3.L.4, 4.L.4   | 3.RI.3, 4.RI.3<br>3.RI.5<br>3.RI.7, 4.RI.7<br>3.RF.3, 4.RF.3<br>3.RF.4, 4.RF.4<br>3.L.4, 4.L.4  | 3.RI.3, 4.RI.3<br>3.RI.5<br>3.RI.7, 4.RI.7<br>3.RF.3, 4.RF.3<br>3.RF.4, 4.RF.4<br>3.L.4, 4.L.4   | 3.RI.3, 4.RI.3<br>3.RI.5<br>3.RI.7, 4.RI.7<br>3.RF.3, 4.RF.3<br>3.RF.4, 4.RF.4<br>3.L.4, 4.L.4                               | 3.RI.3, 4.RI.3<br>3.RI.5<br>3.RI.7, 4.RI.7<br>3.RF.3, 4.RF.3<br>3.RF.4, 4.RF.4<br>3.L.4, 4.L.4  | 3.RI.3, 4.RI.3<br>3.RI.5<br>3.RI.7, 4.RI.7<br>3.RF.3, 4.RF.3<br>3.RF.4, 4.RF.4<br>3.L.4, 4.L.4   |
| CSTA Computer Science Standards | L1:3.CT.1<br>L1:3.CT.4<br>L1:6.CT.1<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6<br>L1:6.CPP.8   | L1:3.CT.1<br>L1:3.CT.4<br>L1:6.CT.1<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6<br>L1:6.CPP.8                      | L1:3.CT.1<br>L1:3.CT.4<br>L1:6.CT.1<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6<br>L1:6.CPP.8  | L1:3.CT.1<br>L1:3.CT.4<br>L1:6.CT.1<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6<br>L1:6.CPP.8   | L1:3.CT.1<br>L1:3.CT.4<br>L1:6.CT.1<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6<br>L1:6.CPP.8  | L1:3.CT.1<br>L1:3.CT.4<br>L1:6.CT.1<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6<br>L1:6.CPP.8                                  | L1:3.CT.1<br>L1:3.CT.4<br>L1:6.CT.1<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6<br>L1:6.CPP.8   | L1:3.CT.1<br>L1:3.CT.4<br>L1:6.CT.1<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6<br>L1:6.CPP.8  |
| UK National Curriculum          | Keystages 1 & 2 Computing*  | Keystages 1 & 2 Computing*   | Keystages 1 & 2 Computing*   | Keystages 1 & 2 Computing*  | Keystages 1 & 2 Computing*   | Keystages 1 & 2 Computing*   | Keystages 1 & 2 Computing*  | Keystages 1 & 2 Computing*   |
| Sample Application of Skills    | Create an underwater adventure game involving animating and moving fish, collecting treasure, and avoiding enemies.   | Create a painting program that lets the user change the colors, shades, and sizes of the brush strokes.          | Create a ghost-catcher game where ghosts respond to touching certain colors.   | Create an adventure game where a Hero must animate to earn power-ups while avoiding enemies.  | Create a Whack-a-Mole style of game where characters appear in random locations quickly.   | Build a “Where’s Waldo?” style game with multiple characters overlapping and hiding.   | Create an obstacle course game.   | Create a snowball fight game, with different characters hiding, appearing, and disappearing when hit.  |

# Programming 102

# Grades 3-4

## Scope and Sequence

Each lesson takes about one class period to complete.

Programming 102 is a course for students in grade 3 or 4 who have completed Tynker's Programming 101 course. The stories, games, puzzles, and projects engage students in developing computational thinking skills, as listed below from the CSTA Level 1 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 -<br>Drawing<br>Patterns  | Lesson 10 -<br>Color Effects  | Lesson 11 -<br>Stamping   | Lesson 12 -<br>Star Runner   | Lesson 13 -<br>Space Breaker   | Lesson 14 -<br>2-Player Battle   | Lesson 15 -<br>Sky Train  | Lesson 16 -<br>Final Lesson  |   |
|--|--|---|---|--|--|--|---|--|---|
| Key Skills and<br>Concepts               | <ul style="list-style-type: none"> <li>Use keyboard controls to point and turn characters, and have them use various drawing effects.</li> </ul> | <ul style="list-style-type: none"> <li>Apply a variety of graphic effects to objects and characters.</li> <li>Use functions to re-use groups of code blocks.</li> </ul> | <ul style="list-style-type: none"> <li>Make multiple copies of characters and objects</li> <li>Change sizes of characters and objects by a percentage</li> <li>Draw text on the screen</li> </ul> | <ul style="list-style-type: none"> <li>Draw lines on the screen using a new method</li> </ul>  | <ul style="list-style-type: none"> <li>Make a simple timer to control a game, and reset timer</li> </ul> | <ul style="list-style-type: none"> <li>Use nested loops</li> <li>Use conditional loops</li> <li>Use "or" in a conditional statement</li> </ul> | <ul style="list-style-type: none"> <li>Use advanced motion techniques</li> <li>Call functions</li> <li>Use more math operators</li> </ul> | <ul style="list-style-type: none"> <li>Open-ended projects</li> </ul>  |   |
| CCSS-Math<br>Standards                   | MP.1<br>4.MD.5   | MP.1  | MP.1  | MP.1   | MP.1   | MP.1   | MP.1  | MP.1   |   |
| CCSS-ELA<br>Standards                    | 3.RI.3, 4.RI.3<br>3.RI.5<br>3.RI.7, 4.RI.7<br>3.RF.3, 4.RF.3<br>3.RF.4, 4.RF.4<br>3.L.4, 4.L.4   | 3.RI.3, 4.RI.3<br>3.RI.5<br>3.RI.7, 4.RI.7<br>3.RF.3, 4.RF.3<br>3.RF.4, 4.RF.4<br>3.L.4, 4.L.4  | 3.RI.3, 4.RI.3<br>3.RI.5<br>3.RI.7, 4.RI.7<br>3.RF.3, 4.RF.3<br>3.RF.4, 4.RF.4<br>3.L.4, 4.L.4  | 3.RI.3, 4.RI.3<br>3.RI.5<br>3.RI.7, 4.RI.7<br>3.RF.3, 4.RF.3<br>3.RF.4, 4.RF.4<br>3.L.4, 4.L.4 | 3.RI.3, 4.RI.3<br>3.RI.5<br>3.RI.7, 4.RI.7<br>3.RF.3, 4.RF.3<br>3.RF.4, 4.RF.4<br>3.L.4, 4.L.4           | 3.RI.3, 4.RI.3<br>3.RI.5<br>3.RI.7, 4.RI.7<br>3.RF.3, 4.RF.3<br>3.RF.4, 4.RF.4<br>3.L.4, 4.L.4   | 3.RI.3, 4.RI.3<br>3.RI.5<br>3.RI.7, 4.RI.7<br>3.RF.3, 4.RF.3<br>3.RF.4, 4.RF.4<br>3.L.4, 4.L.4  | 3.RI.3, 4.RI.3<br>3.RI.5<br>3.RI.7, 4.RI.7<br>3.RF.3, 4.RF.3<br>3.RF.4, 4.RF.4<br>3.L.4, 4.L.4                 | 3.W.3, 4.W.3, 3.W.4,<br>4.W.4, 3.W.6, 4.W.6<br>3.L.1, 4.L.1, 3.L.2,<br>4.L.2, 3.L.3, 4.L.3,<br>3.L.4, 4.L.4, 3.RI.3,<br>4.RI.3, 3.RI.5, 3.RI.7,<br>4.RI.7, 3.RF.3, 4.RF.3<br>3.RF.4, 4.RF.4 |
| CSTA<br>Computer<br>Science<br>Standards | L1:3.CT.1<br>L1:3.CT.4<br>L1:6.CT.1<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6<br>L1:6.CPP.8  | L1:3.CT.1<br>L1:3.CT.4<br>L1:6.CT.1<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6<br>L1:6.CPP.8   | L1:3.CT.1<br>L1:3.CT.4<br>L1:6.CT.1<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6<br>L1:6.CPP.8   | L1:3.CT.1<br>L1:3.CT.4<br>L1:6.CT.1<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6<br>L1:6.CPP.8    | L1:3.CT.1<br>L1:3.CT.4<br>L1:6.CT.1<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6<br>L1:6.CPP.8              | L1:3.CT.1<br>L1:3.CT.4<br>L1:6.CT.1<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6<br>L1:6.CPP.8  | L1:3.CT.1<br>L1:3.CT.4<br>L1:6.CT.1<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6<br>L1:6.CPP.8   | L1:3.CT.1<br>L1:3.CT.4<br>L1:6.CT.1<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6<br>L1:6.CPP.8                    | L1:3.CT.1<br>L1:3.CT.4<br>L1:6.CT.1<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6<br>L1:6.CPP.8   |
| UK National<br>Curriculum                | Keystages 1 & 2<br>Computing*  | Keystages 1 & 2<br>Computing*   | Keystages 1 & 2<br>Computing*   | Keystages 1 & 2<br>Computing*  | Keystages 1 & 2<br>Computing*  | Keystages 1 & 2<br>Computing*  | Keystages 1 & 2<br>Computing*   | Keystages 1 & 2<br>Computing*  |   |
| Sample<br>Application of<br>Skills       | Create interesting circular patterns among the stars by making starships turn and change colors.   | Create a constellation maker that draws lines between draggable stars and uses a variety of graphic effects.  | Fill a scene with multiple copies of characters and objects with different costumes.  | Create a game where a starship can shoot lasers at stars and enemy ships.                      | Build a Space Breaker game with a ball, paddles, and bricks.   | Develop a battle game with lasers, characters, scenes, and music.  | Create a Sky Rider with train cars that follow along one after the other, and collect treasures.  | Five open-ended projects allow students to use their creativity and apply all of the concepts they've learned. |   |

# Programming 201

# Grades 5-6

## Scope and Sequence

Each lesson takes about one class period to complete.

Programming 201 is a course for students in grade 5 or 6 who are new to Tynker. The stories, games, puzzle and projects engage students in developing computational thinking skills, as listed below from the CSTA Level 1 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 - Loops and Animation  | Lesson 3 - Creating a Scene   | Lesson 4 - Jumping over Obstacles   | Lesson 5 - Storytelling   | Lesson 6 - Keyboard Interaction  | Lesson 7 - Guessing Game   | Lesson 8 - Rotation  |
|---------------------------------|---|---|---|---|---|--|--|--|
| Key Skills and Concepts         | <ul style="list-style-type: none"> <li>Use blocks to create a program</li> <li>Sequencing of steps</li> <li>Use simple animation</li> <li>Use sounds</li> <li>Use loops for repetition</li> </ul> | <ul style="list-style-type: none"> <li>Create and animate characters</li> <li>Add background and sounds to a scene</li> </ul> | <ul style="list-style-type: none"> <li>Add sounds to characters</li> <li>Handle key-press events</li> <li>Animate character when clicked</li> </ul> | <ul style="list-style-type: none"> <li>Move characters up and down using changes in y-values</li> </ul> | <ul style="list-style-type: none"> <li>Use speech bubbles to make characters have a conversation</li> <li>Use different kinds of speech bubbles to convey emotion</li> <li>Use a delay to control timing</li> </ul> | <ul style="list-style-type: none"> <li>Move characters using keyboard controls and changes in x- and y-values</li> </ul> | <ul style="list-style-type: none"> <li>Use conditional statements and branching logic</li> <li>Use some math operators</li> <li>Wait for a signal</li> </ul> | <ul style="list-style-type: none"> <li>Use direction and turning</li> </ul>      |
| CCSS-Math Standards             | MP.1  | MP.1  | MP.1  | MP.1<br>5.G.1<br>6.NS.5   | MP.1  | MP.1<br>5.G.1<br>6.NS.5  | MP.1<br>5.G.1<br>5.G.2<br>6.NS.6   | MP.1<br>5.G.1<br>5.G.2<br>6.NS.6   |
| CCSS-ELA Standards              | 5.RF.4<br>6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7   | 5.RF.4<br>6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7   | 5.RF.4<br>6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7   | 5.RF.4<br>6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7   | 5.W.3, 6.W.3<br>5.W.4, 6.W.4<br>5.L.1, 6.L.1<br>5.L.2, 6.L.2<br>5.L.3, 6.L.3<br>5.RF.4<br>6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7   | 5.RF.4<br>6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7  | 5.W.3, 6.W.3<br>5.W.4, 6.W.4<br>5.L.1, 6.L.1<br>5.L.2, 6.L.2<br>5.L.3, 6.L.3<br>5.RF.4<br>6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7                                | 5.RF.4<br>6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7                                    |
| CSTA Computer Science Standards | L1:6.CT.1<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6<br>L1:6.CPP.8   | L1:6.CT.1<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6<br>L1:6.CPP.8   | L1:6.CT.1<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6<br>L1:6.CPP.8   | L1:6.CT.1<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6<br>L1:6.CPP.8                                       | L1:6.CT.1<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6<br>L1:6.CPP.8<br>L1:6.CD.1  | L1:6.CT.1<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6<br>L1:6.CPP.8<br>L1:6.CD.1   | L1:6.CT.1<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6<br>L1:6.CPP.8<br>L1:6.CD.1   | L1:6.CT.1<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6<br>L1:6.CPP.8                |
| UK National Curriculum          | Keystages 2 & 3 Computing*  | Keystages 2 & 3 Computing*  | Keystages 2 & 3 Computing*  | Keystages 2 & 3 Computing*  | Keystages 2 & 3 Computing*  | Keystages 2 & 3 Computing*   | Keystages 2 & 3 Computing*   | Keystages 2 & 3 Computing*   |
| Sample Application of Skills    | Use loops to create a basic stop-motion animation.  | Create a natural environment scene with sounds from that habitat.   | Create a slideshow on a historical event. Add actors and record your own voice for narration.   | Create a scene with multiple characters that animate and jump up and down when clicked.                 | Create a dialogue that tackles a digital citizenship issue such as cyber bullying.  | Update the natural environment scene to make various animals move and speak.   | Create an interactive quiz game to assess knowledge about any subject.   | Create a scene with multiple characters that can animate, move, flip and rotate. |

# Programming 201

# Grades 5-6

## Scope and Sequence

Each lesson takes about one class period to complete.

Programming 201 is a course for students in grade 5 or 6 who are new to Tynker. The stories, games, puzzle and projects engage students in developing computational thinking skills, as listed below from the CSTA Level 1 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 -<br>Detecting<br>Screen Bounds  | Lesson 10 -<br>Music and<br>Animation  | Lesson 11 -<br>Instruments<br>and Tempo  | Lesson 12 -<br>Broadcasting<br>and Messages  | Lesson 13 -<br>Time Limits  | Lesson 14 -<br>Message<br>Driven<br>Programming   | Lesson 15 -<br>Pop the<br>Balloon   | Lesson 16 -<br>Animation<br>with<br>Movement  |
|--|---|--|--|--|---|---|---|---|
| Key Skills and<br>Concepts               | <ul style="list-style-type: none"> <li>Use screen bounds to make characters bounce at screen edge</li> <li>Move characters and objects to random screen locations using random numbers</li> <li>Make characters and objects appear and disappear</li> </ul> | <ul style="list-style-type: none"> <li>Add background music for a character</li> </ul> | <ul style="list-style-type: none"> <li>Generate music using musical notes, various instruments, and changing tempos</li> </ul> | <ul style="list-style-type: none"> <li>Send and receive messages between characters</li> </ul> | <ul style="list-style-type: none"> <li>Use properties of other characters</li> <li>Make a simple timer to control a game</li> <li>Troubleshoot and debug a program</li> </ul> | <ul style="list-style-type: none"> <li>Make characters perform different animations based on the messages they receive</li> </ul> | <ul style="list-style-type: none"> <li>Receive a message to run a program</li> <li>Use animation techniques to simulate explosions</li> </ul> | <ul style="list-style-type: none"> <li>Send a message to make a character move, animate and make sounds at the same time</li> </ul> |
| CCSS-Math<br>Standards                   | MP.1<br>5.G.1<br>5.G.2<br>6.NS.6  | MP.1   | MP.1   | MP.1   | MP.1<br>5.G.1<br>5.G.2<br>6.NS.6  | MP.1  | MP.1<br>5.G.1<br>5.G.2<br>6.NS.6  | MP.1<br>5.G.1<br>5.G.2<br>6.NS.5<br>6.NS.6  |
| CCSS-ELA<br>Standards                    | 5.RF.4<br>6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7   | 5.RF.4<br>6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7  | 5.RF.4<br>6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7  | 5.RF.4<br>6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7  | 5.RF.4<br>6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7   | 5.RF.4<br>6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7   | 5.RF.4<br>6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7   | 5.RF.4<br>6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7   |
| CSTA<br>Computer<br>Science<br>Standards | L1:6.CT.1<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6<br>L1:6.CPP.8   | L1:6.CT.1<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6<br>L1:6.CPP.8                      | L1:6.CT.1<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6<br>L1:6.CPP.8  | L1:6.CT.1<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6<br>L1:6.CPP.8                              | L1:6.CT.1<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6<br>L1:6.CPP.8   | L1:6.CT.1<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6<br>L1:6.CPP.8   | L1:6.CT.1<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6<br>L1:6.CPP.8   | L1:6.CT.1<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6<br>L1:6.CPP.8   |
| UK National<br>Curriculum                | Keystages 2 & 3<br>Computing*   | Keystages 2 & 3<br>Computing*  | Keystages 2 & 3<br>Computing*  | Keystages 2 & 3<br>Computing*  | Keystages 2 & 3<br>Computing*   | Keystages 2 & 3<br>Computing*   | Keystages 2 & 3<br>Computing*   | Keystages 2 & 3<br>Computing*   |
| Sample<br>Application of<br>Skills       | Create a two-player space shooter game.   | Use music to create a game where robots battle to mimic the sound sequence.            | Build a traditional piano and a unique piano.  | Create an interactive musical activity involving multiple characters.                          | Create a timed two-player game.   | Program a BeatBot to perform dance moves.   | Create a balloon-popping game with explosion effects.   | Combine animation with music and movement to create a music video.  |

# Programming 201

# Grades 5-6

## Scope and Sequence

Each lesson takes about one class period to complete.

Programming 201 is a course for students in grade 5 or 6 who are new to Tynker. The stories, games, puzzle and projects engage students in developing computational thinking skills, as listed below from the CSTA Level 1 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 17 -<br>Obstacle<br>Course   |  |  |  |  |  |  |  |
|--|---|--|--|--|--|--|--|--|
| <b>Key Skills and Concepts</b>         | <ul style="list-style-type: none"> <li>• Use nested loops to repeat specific actions</li> <li>• Make characters follow other characters and cursor</li> <li>• Make characters bounce when they reach screen edge</li> </ul> |  |  |  |  |  |  |  |
| <b>CCSS-Math Standards</b>             | MP.1<br>5.G.1<br>5.G.2<br>6.NS.6  |  |  |  |  |  |  |  |
| <b>CCSS-ELA Standards</b>              | 5.RF.4<br>6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7   |  |  |  |  |  |  |  |
| <b>CSTA Computer Science Standards</b> | L1:6.CT.1<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6<br>L1:6.CPP.8   |  |  |  |  |  |  |  |
| <b>UK National Curriculum</b>          | Keystages 2 & 3 Computing*  |  |  |  |  |  |  |  |
| <b>Sample Application of Skills</b>    | <p>Create an obstacle course or a parade featuring animated characters and objects.</p> <p>Apply all concepts to build a highly interactive project.</p>  |  |  |  |  |  |  |  |

# Programming 202

# Grades 5-6

## Scope and Sequence

Each lesson takes about one class period to complete.

Programming 202 is a course for students in grade 5 or 6 who have completed Tynker's Programming 201 course. The stories, games, puzzles, and projects engage students in developing computational thinking skills, as listed below from the CSTA Level 1 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 -<br>Pen Drawing  | Lesson 2 -<br>Following the<br>Mouse   | Lesson 3 -<br>Changing Size   | Lesson 4 -<br>Changing Pen<br>Color  | Lesson 5 -<br>Detecting<br>Colors   | Lesson 6 -<br>Avoiding<br>Obstacles   | Lesson 7 -<br>Geometry  | Lesson 8 -<br>Game Effects<br>and Rules   |
|--|--|--|---|--|---|---|---|---|
| Key Skills and<br>Concepts               | <ul style="list-style-type: none"> <li>Use the pen drawing tool and change pen settings</li> <li>Point toward the mouse pointer</li> </ul> | <ul style="list-style-type: none"> <li>Make characters follow other characters and cursor</li> <li>Move characters and objects to random screen locations using random number generators</li> <li>Make objects appear and disappear</li> </ul> | <ul style="list-style-type: none"> <li>Make characters change their size, and create the illusion of perspective</li> </ul> | <ul style="list-style-type: none"> <li>Change the color and size of pen for drawing</li> </ul> | <ul style="list-style-type: none"> <li>Detect colors, touches, and screen bounds</li> </ul> | <ul style="list-style-type: none"> <li>Use keyboard controls to point and turn characters</li> <li>Apply a variety of graphic effects to objects and characters.</li> </ul> | <ul style="list-style-type: none"> <li>Draw geometric shapes</li> </ul> | <ul style="list-style-type: none"> <li>Handle advanced events</li> <li>Use conditional statements to trigger special graphic and sound effects</li> </ul> |
| CCSS-Math<br>Standards                   | MP.1   | MP.1<br>5.G.1<br>5.G.2<br>6.NS.6   | MP.1  | MP.1   | MP.1  | MP.1<br>5.G.1<br>6.NS.5   | MP.1  | MP.1  |
| CCSS-ELA<br>Standards                    | 5.RF.4<br>6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7  | 5.RF.4<br>6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7  | 5.RF.4<br>6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7   | 5.RF.4<br>6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7  | 5.RF.4<br>6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7   | 5.RF.4<br>6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7   | 5.RF.4<br>6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7                           | 5.RF.4<br>6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7   |
| CSTA<br>Computer<br>Science<br>Standards | L1:6.CT.1<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6<br>L1:6.CPP.8  | L1:6.CT.1<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6<br>L1:6.CPP.8  | L1:6.CT.1<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6<br>L1:6.CPP.8   | L1:6.CT.1<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6<br>L1:6.CPP.8                              | L1:6.CT.1<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6<br>L1:6.CPP.8                           | L1:6.CT.1<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6<br>L1:6.CPP.8   | L1:6.CT.1<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6<br>L1:6.CPP.8       | L1:6.CT.1<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6<br>L1:6.CPP.8   |
| UK National<br>Curriculum                | Keystages 2 & 3<br>Computing*  | Keystages 2 & 3<br>Computing*  | Keystages 2 & 3<br>Computing*   | Keystages 2 & 3<br>Computing*  | Keystages 2 & 3<br>Computing*   | Keystages 2 & 3<br>Computing*   | Keystages 2 & 3<br>Computing*   | Keystages 2 & 3<br>Computing*   |
| Sample<br>Application of<br>Skills       | Draw shapes and patterns using pen drawing commands.   | Build a chase game where creatures or vehicles chase after each other.   | Program a racer to jump over enemy lines to reach its goal.   | Create an Etch-a-Sketch® style drawing machine.  | Make characters and objects interact with each other by detecting touches and colors.       | Create an obstacle course   | Draw a variety of geometric shapes using a racer.                       | Create a game using ghosting, portals, sound effects, and advanced rules.   |



# Programming 202

# Grades 5-6

## Scope and Sequence

Each lesson takes about one class period to complete.

Programming 202 is a course for students in grade 5 or 6 who have completed Tynker's Programming 201 course. The stories, games, puzzles, and projects engage students in developing computational thinking skills, as listed below from the CSTA Level 1 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 -<br>The Physics Engine   | Lesson 10 -<br>Gravity and Bouncing   | Lesson 11 -<br>Static Platforms  | Lesson 12 -<br>Basketball   | Lesson 13 -<br>Impulses  | Lesson 14 -<br>Projectiles   | Lesson 15 -<br>Timers  | Lesson 16 -<br>Asteroid Pong   |
|---------------------------------|--|---|--|---|--|--|--|--|
| Key Skills and Concepts         | <ul style="list-style-type: none"> <li>• Activate and use the physics engine</li> </ul>  | <ul style="list-style-type: none"> <li>• Apply gravity and restitution to characters and objects</li> </ul>             | <ul style="list-style-type: none"> <li>• Apply active and static properties to characters and objects</li> </ul>                             | <ul style="list-style-type: none"> <li>• Apply physics and animation skills and concepts</li> </ul> | <ul style="list-style-type: none"> <li>• Apply impulses to projectiles to affect how fast they move</li> </ul> | <ul style="list-style-type: none"> <li>• Apply expert physics commands to shoot projectiles at multiple targets</li> </ul> | <ul style="list-style-type: none"> <li>• Use timers and keyboard controls to increase interactivity and control over actors</li> <li>• Apply mathematical operators in conditional statements</li> </ul> | <ul style="list-style-type: none"> <li>• Use multiplayer animations that switch from one animation to another</li> <li>• Broadcast messages to all characters</li> </ul> |
| CCSS-Math Standards             | MP.1   | MP.1  | MP.1   | MP.1  | MP.1   | MP.1   | MP.1   | MP.1   |
| CCSS-ELA Standards              | 5.RF.4<br>6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7  | 5.RF.4<br>6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7   | 5.RF.4<br>6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7  | 5.RF.4<br>6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7   | 5.RF.4<br>6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7  | 5.RF.4<br>6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7  | 5.RF.4<br>6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7  | 5.RF.4<br>6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7  |
| CSTA Computer Science Standards | L1:6.CT.1<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6<br>L1:6.CPP.8                        | L1:6.CT.1<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6<br>L1:6.CPP.8   | L1:6.CT.1<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6<br>L1:6.CPP.8  | L1:6.CT.1<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6<br>L1:6.CPP.8                                   | L1:6.CT.1<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6<br>L1:6.CPP.8  | L1:6.CT.1<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6<br>L1:6.CPP.8  | L1:6.CT.1<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6<br>L1:6.CPP.8  | L1:6.CT.1<br>L1:6.CPP.1<br>L1:6.CPP.5<br>L1:6.CPP.6<br>L1:6.CPP.8  |
| UK National Curriculum          | Keystages 2 & 3 Computing*   | Keystages 2 & 3 Computing*  | Keystages 2 & 3 Computing*   | Keystages 2 & 3 Computing*  | Keystages 2 & 3 Computing*   | Keystages 2 & 3 Computing*   | Keystages 2 & 3 Computing*   | Keystages 2 & 3 Computing*   |
| Sample Application of Skills    | Create a structure using multiple platforms and blocks, where the laws of physics apply. | Create a simulation where the direction of gravity can be changed with keyboard controls. Create a bug volleyball game. | Build a game with balls or winged creatures that can be shot out of a cannon to knock down a structure that might have hooved animals in it. | Make a fully-functional basketball game using the physics engine.                                   | Create a game involving shooting cannonballs at moving targets.  | Develop a cannon simulation where a variety of projectiles can be shot at multiple targets.                                | Create a cannonball shooting game with more finely-tuned controls.   | Create a timed multiplayer game involving collisions and other physics concepts.   |

# Programming 301

# Grades 7-8

## Scope and Sequence

Each lesson takes about one class period to complete.

Programming 301 is a course for students in grade 7 or 8 who are new to Tynker. The stories, games, puzzles, and projects engage students in developing computational thinking skills, as listed below from the CSTA Level 2 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 - Animated Motion   | Lesson 3 - Actor Positioning  | Lesson 4 - Motion and Tracking  | Lesson 5 - Conditional Loops  | Lesson 6 - Show and Hide   | Lesson 7 - Actor Properties   | Lesson 8 - Nested Loops   |
|---------------------------------|---|--|---|---|---|--|---|---|
| Key Skills and Concepts         | <ul style="list-style-type: none"> <li>Use blocks to create a program</li> <li>Sequencing of steps</li> <li>Use simple animation</li> <li>Use sounds</li> <li>Use loops for repetition</li> </ul> | <ul style="list-style-type: none"> <li>Add a background and characters to a scene</li> <li>Add sounds to a background and to characters</li> </ul> | <ul style="list-style-type: none"> <li>Show/hide characters, and make them appear in a new position using x- and y-coordinates</li> </ul> | <ul style="list-style-type: none"> <li>Move characters using keyboard</li> <li>Detect conditions</li> <li>Handle key-press events</li> <li>Make Actors perform specific actions when they touch others</li> </ul> | <ul style="list-style-type: none"> <li>Use functions to handle sub-tasks</li> <li>Use conditional loops</li> <li>Use animation</li> </ul> | <ul style="list-style-type: none"> <li>Handle advanced events</li> <li>Use advanced motion techniques</li> </ul> | <ul style="list-style-type: none"> <li>Use direction and turning</li> <li>Use properties of characters</li> <li>Use math operators</li> <li>Use conditional statements and branching logic</li> </ul> | <ul style="list-style-type: none"> <li>Use nested loops to give multiple lives to characters</li> <li>Move characters and objects to various screen locations using random number generators</li> </ul> |
| CCSS-Math Standards             | MP.1  | MP.1   | MP.1  | MP.1  | MP.1  | MP.1   | MP.1  | MP.1  |
| CCSS-ELA Standards              | 7.RI.4, 8.RI.4<br>6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7   | 7.RI.4, 8.RI.4<br>6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7  | 7.RI.4, 8.RI.4<br>6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7   | 7.RI.4, 8.RI.4<br>6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7   | 7.RI.4, 8.RI.4<br>6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7   | 7.RI.4, 8.RI.4<br>6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7  | 7.RI.4, 8.RI.4<br>6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7   | 7.RI.4, 8.RI.4<br>6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7   |
| CSTA Computer Science Standards | L2:CT.1<br>L2:CT.6<br>L2:CPP.4<br>L2:CPP.5<br>L2:CPP.8<br>L2:CD.1   | L2:CT.1<br>L2:CT.6<br>L2:CPP.4<br>L2:CPP.5<br>L2:CPP.8<br>L2:CD.1  | L2:CT.1<br>L2:CT.6<br>L2:CPP.4<br>L2:CPP.5<br>L2:CPP.8<br>L2:CD.1   | L2:CT.1<br>L2:CT.6<br>L2:CPP.4<br>L2:CPP.5<br>L2:CPP.8<br>L2:CD.1   | L2:CT.1<br>L2:CT.6<br>L2:CPP.4<br>L2:CPP.5<br>L2:CPP.8<br>L2:CT.12<br>L2:CD.1   | L2:CT.1<br>L2:CT.6<br>L2:CT.12<br>L2:CPP.4<br>L2:CPP.5<br>L2:CPP.8<br>L2:CD.1                                    | L2:CT.1<br>L2:CT.6<br>L2:CT.12<br>L2:CPP.4<br>L2:CPP.5<br>L2:CPP.8<br>L2:CD.1   | L2:CT.1<br>L2:CT.6<br>L2:CT.12<br>L2:CPP.4<br>L2:CPP.5<br>L2:CPP.8<br>L2:CD.1   |
| UK National Curriculum          | Keystages 2 & 3 Computing*  | Keystages 2 & 3 Computing*   | Keystages 2 & 3 Computing*  | Keystages 2 & 3 Computing*  | Keystages 2 & 3 Computing*  | Keystages 2 & 3 Computing*   | Keystages 2 & 3 Computing*  | Keystages 2 & 3 Computing*  |
| Sample Application of Skills    | Create a game with a character, a background, music, motion, enemies and treasures.   | Create a monster mash dance party.   | Create a hide and seek game.  | Create a castle maze.   | Create game with new animation techniques.  | Create wizards vs. zombies game.   | Create a fireball tennis game.  | Create a hero vs. enemy game.   |

# Programming 301

# Grades 7-8

## Scope and Sequence

Each lesson takes about one class period to complete.

Programming 301 is a course for students in grade 7 or 8 who are new to Tynker. The stories, games, puzzles, and projects engage students in developing computational thinking skills, as listed below from the CSTA Level 2 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 - Messaging  | Lesson 10 - Start Screen and Controls   | Lesson 11 - Shoot Projectiles   | Lesson 12 - Parallax Scrolling   | Lesson 13 - Cloning   | Lesson 14 - Variables   | Lesson 15 - Power-Ups and Effects   | Lesson 16 - Boss Battle   |
|---------------------------------|---|---|---|--|---|---|---|---|
| Key Skills and Concepts         | <ul style="list-style-type: none"> <li>Send, receive, and broadcast messages</li> <li>Use conditional wait</li> <li>Create Win and Game Over screens</li> </ul> | <ul style="list-style-type: none"> <li>Create a start screen with a start button and a broadcasted message</li> <li>Create button using Art Studio</li> <li>Set the rotation style of characters</li> </ul> | <ul style="list-style-type: none"> <li>Use further types of conditional loops</li> <li>Use screen bounds</li> <li>Make characters bounce when they reach screen edge</li> </ul> | <ul style="list-style-type: none"> <li>Use and change layers</li> <li>Send character/object to front or back</li> <li>Use two characters to create a scrolling background with parallax effect</li> <li>Use advanced animation techniques</li> </ul> | <ul style="list-style-type: none"> <li>Cloning characters/objects</li> <li>Make clones behave in specific ways</li> </ul> | <ul style="list-style-type: none"> <li>Add and change variables</li> <li>Use local and global variables, like health and score</li> </ul> | <ul style="list-style-type: none"> <li>Use graphic effects like changing color</li> <li>Use true/false variables</li> <li>Use a variable timer</li> </ul> | <ul style="list-style-type: none"> <li>Use more advanced motion techniques</li> </ul> |
| CCSS-Math Standards             | MP.1  | MP.1  | MP.1  | MP.1   | MP.1  | MP.1<br>MP.2<br>MP.4  | MP.1<br>MP.2<br>MP.4  | MP.1<br>MP.2<br>MP.4  |
| CCSS-ELA Standards              | 7.RI.4, 8.RI.4<br>6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7   | 7.RI.4, 8.RI.4<br>6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7   | 7.RI.4, 8.RI.4<br>6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7   | 7.RI.4, 8.RI.4<br>6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7  | 7.RI.4, 8.RI.4<br>6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7   | 7.RI.4, 8.RI.4<br>6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7   | 7.RI.4, 8.RI.4<br>6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7   | 7.RI.4, 8.RI.4<br>6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7                                 |
| CSTA Computer Science Standards | L2:CT.1<br>L2:CT.6<br>L2:CT.12<br>L2:CPP.4<br>L2:CPP.5<br>L2:CPP.8<br>L2:CD.1   | L2:CT.1<br>L2:CT.6<br>L2:CT.12<br>L2:CPP.4<br>L2:CPP.5<br>L2:CPP.8<br>L2:CD.1   | L2:CT.1<br>L2:CT.6<br>L2:CT.12<br>L2:CPP.4<br>L2:CPP.5<br>L2:CPP.8<br>L2:CD.1   | L2:CT.1<br>L2:CT.6<br>L2:CT.12<br>L2:CPP.4<br>L2:CPP.5<br>L2:CPP.8<br>L2:CD.1  | L2:CT.1<br>L2:CT.6<br>L2:CT.12<br>L2:CPP.4<br>L2:CPP.5<br>L2:CPP.8<br>L2:CD.1   | L2:CT.1<br>L2:CT.6<br>L2:CT.12<br>L2:CPP.4<br>L2:CPP.5<br>L2:CPP.8<br>L2:CD.1   | L2:CT.1<br>L2:CT.6<br>L2:CT.12<br>L2:CPP.4<br>L2:CPP.5<br>L2:CPP.8<br>L2:CD.1   | L2:CT.1<br>L2:CT.6<br>L2:CT.12<br>L2:CPP.4<br>L2:CPP.5<br>L2:CPP.8<br>L2:CD.1         |
| UK National Curriculum          | Keystages 2 & 3 Computing*  | Keystages 2 & 3 Computing*  | Keystages 2 & 3 Computing*  | Keystages 2 & 3 Computing*   | Keystages 2 & 3 Computing*  | Keystages 2 & 3 Computing*  | Keystages 2 & 3 Computing*  | Keystages 2 & 3 Computing*  |
| Sample Application of Skills    | Build an adventure game.  | Create a top-down arcade game.  | Create a game where heroes avoid enemies.   | Create a game with a scrolling background with the parallax effect.  | Create a 2-player airplane battle game.   | Create a snowball fight game.   | Create a game involving character health and a power-up.  | Create a hero vs. enemy game with advanced motion techniques.                         |

# Programming 301

Grades 7-8

## Scope and Sequence

Each lesson takes about one class period to complete.

Programming 201 is a course for students in grade 5 or 6 who are new to Tynker. The stories, games, puzzle and projects engage students in developing computational thinking skills, as listed below from the CSTA Level 1 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 17 -<br>Finishing<br>Touch   |  |  |  |  |  |  |  |
|--|---|--|--|--|--|--|--|--|
| <b>Key Skills and Concepts</b>         | <ul style="list-style-type: none"> <li>Store letters or words as value of a variable</li> </ul> |  |  |  |  |  |  |  |
| <b>CCSS-Math Standards</b>             | MP.1<br>MP.2<br>MP.4  |  |  |  |  |  |  |  |
| <b>CCSS-ELA Standards</b>              | 7.RI.4, 8.RI.4<br>6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7   |  |  |  |  |  |  |  |
| <b>CSTA Computer Science Standards</b> | L2:CT.1<br>L2:CT.6<br>L2:CT.12<br>L2:CPP.4<br>L2:CPP.5<br>L2:CPP.8<br>L2:CD.1                   |  |  |  |  |  |  |  |
| <b>UK National Curriculum</b>          | Keystages 2 & 3 Computing*  |  |  |  |  |  |  |  |
| <b>Sample Application of Skills</b>    | Finish top-down arcade game with advanced variable and motion techniques.                       |  |  |  |  |  |  |  |

# Programming 302

# Grades 7-8

## Scope and Sequence

Each lesson takes about one class period to complete.

Programming 302 is a course for students in grade 7 or 8 who have completed Tynker's Programming 301 course. The stories, games, puzzles, and projects engage students in developing computational thinking skills, as listed below from the CSTA Level 2 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 - Object Stacking   | Lesson 2 - Applying Impulses  | Lesson 3 - Line Animation   | Lesson 4 - Physics Properties   | Lesson 5 - Ricochet   | Lesson 6 - Explosions   | Lesson 7 - Changing Direction   | Lesson 8 - Gravity Sling  |
|---------------------------------|--|---|---|---|---|---|---|---|
| Key Skills and Concepts         | <ul style="list-style-type: none"> <li>Activate the physics engine and use expert physics commands</li> <li>Apply gravity to characters and objects</li> </ul> | <ul style="list-style-type: none"> <li>Apply impulses to projectiles to affect how fast they move</li> </ul>    | <ul style="list-style-type: none"> <li>Draw geometric shapes, using positions of the cursor, characters, and objects</li> </ul> | <ul style="list-style-type: none"> <li>Create moving targets</li> <li>Control behavior and graphic effects for objects after they've been hit by projectiles</li> </ul> | <ul style="list-style-type: none"> <li>Apply density and restitution to characters and objects</li> </ul>       | <ul style="list-style-type: none"> <li>Use simple variables</li> <li>Use conditional loops</li> </ul>                 | <ul style="list-style-type: none"> <li>Change direction of projectiles after launch</li> </ul>                  | <ul style="list-style-type: none"> <li>Apply gravity and special visual effects to simulate the behavior of a black hole</li> </ul> |
| CCSS-Math Standards             | MP.1   | MP.1  | MP.1  | MP.1  | MP.1  | MP.1<br>7.EE.4  | MP.1<br>7.EE.4  | MP.1<br>7.EE.4  |
| CCSS-ELA Standards              | 6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7  | 6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7   | 6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7   | 6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7   | 6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7   | 6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7   | 6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7   | 6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7   |
| CSTA Computer Science Standards | L2:CT.1<br>L2:CT.3<br>L2:CT.4<br>L2:CT.6<br>L2:CT.12<br>L2:CPP.3<br>L2:CPP.4<br>L2:CPP.5<br>L2:CPP.8<br>L2:CD.1  | L2:CT.1<br>L2:CT.3<br>L2:CT.4<br>L2:CT.6<br>L2:CT.12<br>L2:CPP.3<br>L2:CPP.4<br>L2:CPP.5<br>L2:CPP.8<br>L2:CD.1 | L2:CT.1<br>L2:CT.3<br>L2:CT.4<br>L2:CT.6<br>L2:CT.12<br>L2:CPP.3<br>L2:CPP.4<br>L2:CPP.5<br>L2:CPP.8<br>L2:CD.1                 | L2:CT.1<br>L2:CT.3<br>L2:CT.4<br>L2:CT.6<br>L2:CT.12<br>L2:CPP.3<br>L2:CPP.4<br>L2:CPP.5<br>L2:CPP.8<br>L2:CD.1   | L2:CT.1<br>L2:CT.3<br>L2:CT.4<br>L2:CT.6<br>L2:CT.12<br>L2:CPP.3<br>L2:CPP.4<br>L2:CPP.5<br>L2:CPP.8<br>L2:CD.1 | L2:CT.1<br>L2:CT.3<br>L2:CT.4<br>L2:CT.6<br>L2:CT.12<br>L2:CPP.3<br>L2:CPP.4<br>L2:CPP.5<br>L2:CPP.8<br>L2:CD.1       | L2:CT.1<br>L2:CT.3<br>L2:CT.4<br>L2:CT.6<br>L2:CT.12<br>L2:CPP.3<br>L2:CPP.4<br>L2:CPP.5<br>L2:CPP.8<br>L2:CD.1 | L2:CT.1<br>L2:CT.3<br>L2:CT.4<br>L2:CT.6<br>L2:CT.12<br>L2:CPP.3<br>L2:CPP.4<br>L2:CPP.5<br>L2:CPP.8<br>L2:CD.1                     |
| UK National Curriculum          | Keystages 2 & 3 Computing*   | Keystages 2 & 3 Computing*  | Keystages 2 & 3 Computing*  | Keystages 2 & 3 Computing*  | Keystages 2 & 3 Computing*  | Keystages 2 & 3 Computing*  | Keystages 2 & 3 Computing*  | Keystages 2 & 3 Computing*  |
| Sample Application of Skills    | Create a skyscraper with multiple platforms.   | Develop a game where characters and objects can knock a structure over.   | Create a Gravity Drawing Tool and Gravity Sling.  | Build a shooting gallery game with asteroids and aliens.  | Create a gravity maze where balls bounce off walls and gravity can be changed.                                  | Build a Bug Soccer game where insects kick an object around. Create an explosion that can knock out a space platform. | Design a space slingshot game with specially programmable projectiles.  | Create a black hole that sucks up objects that stray too near.  |

# Programming 302

# Grades 7-8

## Scope and Sequence

Each lesson takes about one class period to complete.

Programming 302 is a course for students in grade 7 or 8 who have completed Tynker's Programming 301 course. The stories, games, puzzles, and projects engage students in developing computational thinking skills, as listed below from the CSTA Level 2 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 -<br>Linear Velocity  | Lesson 10 -<br>Double Jump   | Lesson 11 -<br>Receiving<br>Values  | Lesson 12 -<br>Lists  | Lesson 13 -<br>Laser Tennis   | Lesson 14 -<br>Enemy AI   | Lesson 15 -<br>Leaderboards   | Lesson 16 -<br>Platform<br>Movement  |
|--|--|--|---|---|---|---|---|--|
| Key Skills and<br>Concepts               | <ul style="list-style-type: none"> <li>Add and use functions to simplify programming</li> <li>Apply linear velocity</li> </ul> | <ul style="list-style-type: none"> <li>Detect collisions, apply impulses, and run pre-programmed animation sequences so a character will jump twice</li> </ul> | <ul style="list-style-type: none"> <li>Check and change property values of a character or object</li> </ul>     | <ul style="list-style-type: none"> <li>Add and manage lists</li> </ul>  | <ul style="list-style-type: none"> <li>Check and use list inventory</li> </ul>                                  | <ul style="list-style-type: none"> <li>Apply list operations</li> </ul>   | <ul style="list-style-type: none"> <li>Interact with user to collect name data</li> <li>Program a leaderboard to track high scores in a game</li> </ul> | <ul style="list-style-type: none"> <li>Apply velocity vectors to make platforms move</li> <li>Apply further list operations</li> </ul> |
| CCSS-Math<br>Standards                   | MP.1<br>7.EE.4   | MP.1<br>7.EE.4   | MP.1<br>7.EE.4  | MP.1<br>7.EE.4  | MP.1<br>7.EE.4  | MP.1<br>7.EE.4  | MP.1<br>7.EE.4  | MP.1<br>7.EE.4   |
| CCSS-ELA<br>Standards                    | 6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7  | 6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7  | 6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7   | 6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7   | 6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7   | 6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7   | 6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7   | 6-8.RST.3<br>6-8.RST.4<br>6-8.RST.7  |
| CSTA<br>Computer<br>Science<br>Standards | L2:CT.1<br>L2:CT.3<br>L2:CT.4<br>L2:CT.6<br>L2:CT.12<br>L2:CPP.3<br>L2:CPP.4<br>L2:CPP.5<br>L2:CPP.8<br>L2:CD.1                | L2:CT.1<br>L2:CT.3<br>L2:CT.4<br>L2:CT.6<br>L2:CT.12<br>L2:CPP.3<br>L2:CPP.4<br>L2:CPP.5<br>L2:CPP.8<br>L2:CD.1  | L2:CT.1<br>L2:CT.3<br>L2:CT.4<br>L2:CT.6<br>L2:CT.12<br>L2:CPP.3<br>L2:CPP.4<br>L2:CPP.5<br>L2:CPP.8<br>L2:CD.1 | L2:CT.1<br>L2:CT.3<br>L2:CT.4<br>L2:CT.6<br>L2:CT.12<br>L2:CPP.3<br>L2:CPP.4<br>L2:CPP.5<br>L2:CPP.8<br>L2:CD.1 | L2:CT.1<br>L2:CT.3<br>L2:CT.4<br>L2:CT.6<br>L2:CT.12<br>L2:CPP.3<br>L2:CPP.4<br>L2:CPP.5<br>L2:CPP.8<br>L2:CD.1 | L2:CT.1<br>L2:CT.3<br>L2:CT.4<br>L2:CT.6<br>L2:CT.12<br>L2:CPP.3<br>L2:CPP.4<br>L2:CPP.5<br>L2:CPP.8<br>L2:CD.1 | L2:CT.1<br>L2:CT.3<br>L2:CT.4<br>L2:CT.6<br>L2:CT.12<br>L2:CPP.3<br>L2:CPP.4<br>L2:CPP.5<br>L2:CPP.8<br>L2:CD.1   | L2:CT.1<br>L2:CT.3<br>L2:CT.4<br>L2:CT.6<br>L2:CT.12<br>L2:CPP.3<br>L2:CPP.4<br>L2:CPP.5<br>L2:CPP.8<br>L2:CD.1                        |
| UK National<br>Curriculum                | Keystages 2 & 3<br>Computing*  | Keystages 2 & 3<br>Computing*  | Keystages 2 & 3<br>Computing*   | Keystages 2 & 3<br>Computing*   | Keystages 2 & 3<br>Computing*   | Keystages 2 & 3<br>Computing*   | Keystages 2 & 3<br>Computing*   | Keystages 2 & 3<br>Computing*  |
| Sample<br>Application of<br>Skills       | Develop the physics in a platformer game so a character can run off a platform and drop down.                                  | Create a side-scroller game with moving platforms and double-jumping characters.   | Program the "health" and "damage" properties for characters in a game where robots are attacking.               | Create a game where the hero collects power-ups while avoiding laser-shooting enemies.                          | Make a laser tennis game using a list data structure.   | Create an enemy robot that changes position based on data stored in a randomized list of values.                | Use global variables to implement a leaderboard for high scores in a game.  | Build a more complex platform game with power-ups, special hero moves, and platforms that appear to fly by the screen.                 |

# JavaScript 1

## Scope and Sequence

Grades 7+

Each lesson takes about one class period to complete.

JavaScript 1 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 -<br>The Basics  | Lesson 2 -<br>Loops and<br>Patterns   | Lesson 3 -<br>Conditional<br>Logic   | Lesson 4 -<br>Conditional<br>Loops  | Lesson 5 -<br>Variables   | Lesson 6 -<br>Expressions  | Lesson 7 -<br>Using the<br>Canvas   | Lesson 8 -<br>User<br>Interaction  |   |
|--|---|---|--|---|---|--|---|--|---|
| Key Skills and<br>Concepts               | <ul style="list-style-type: none"> <li>Create custom sequences to solve puzzles</li> <li>Use function calls</li> <li>Use JavaScript syntax, naming conventions, and comments</li> </ul> | <ul style="list-style-type: none"> <li>Use simple, nested, and “for” loops</li> <li>Use arithmetic operators</li> <li>Recognize patterns</li> </ul> | <ul style="list-style-type: none"> <li>Distinguish between assignment, comparison, and logical operators</li> <li>Use conditional logic to program algorithms</li> </ul> | <ul style="list-style-type: none"> <li>Use “while” and “do-while” loops</li> <li>Use conditional loops to solve puzzles</li> <li>Distinguish between “for” and “while” loops</li> </ul> | <ul style="list-style-type: none"> <li>Use variable declarations and assignments</li> <li>Use and define strings</li> <li>Use variables to solve puzzles</li> </ul> | <ul style="list-style-type: none"> <li>Write and understand JavaScript expressions</li> <li>Use operator precedence to evaluate an expression</li> </ul> | <ul style="list-style-type: none"> <li>Use canvas elements, sprites, and layering</li> <li>Create game using HTML and JavaScript</li> <li>Create timed loops that execute code with a time delay</li> </ul> | <ul style="list-style-type: none"> <li>Set up and handle keyboard and mouse events</li> <li>Find the correct key code to handle a specific keyboard input</li> </ul> |   |
| CCSS-Math<br>Standards                   | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3   | 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>HSA.REI.1   | 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>HSA.REI.1  | 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>HSA.REI.1   | 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  | 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   | 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>HSA.REI.1   | -  |   |
| CCSS-ELA<br>Standards                    | 9.RI.3, 10.RI.3<br>9.RI.6, 10.RI.6<br>9.RI.10, 10.RI.10<br>9.L.3, 10.L.3<br>9.L.6, 10.L.6   | 9.RI.3, 10.RI.3<br>9.RI.6, 10.RI.6<br>9.RI.10, 10.RI.10<br>9.L.3, 10.L.3<br>9.L.6, 10.L.6   | 9.RI.3, 10.RI.3<br>9.RI.6, 10.RI.6<br>9.RI.10, 10.RI.10<br>9.L.3, 10.L.3<br>9.L.6, 10.L.6  | 9.RI.3, 10.RI.3<br>9.RI.6, 10.RI.6<br>9.RI.10, 10.RI.10<br>9.L.3, 10.L.3<br>9.L.6, 10.L.6   | 9.RI.3, 10.RI.3<br>9.RI.6, 10.RI.6<br>9.RI.10, 10.RI.10<br>9.L.3, 10.L.3<br>9.L.6, 10.L.6   | 9.RI.3, 10.RI.3<br>9.RI.6, 10.RI.6<br>9.RI.10, 10.RI.10<br>9.L.3, 10.L.3<br>9.L.6, 10.L.6  | 9.RI.3, 10.RI.3<br>9.RI.6, 10.RI.6<br>9.RI.10, 10.RI.10<br>9.L.3, 10.L.3<br>9.L.6, 10.L.6   | 9.RI.3, 10.RI.3<br>9.RI.6, 10.RI.6<br>9.RI.10, 10.RI.10<br>9.L.3, 10.L.3<br>9.L.6, 10.L.6  | 9.RI.3, 10.RI.3<br>9.RI.6, 10.RI.6<br>9.RI.10, 10.RI.10<br>9.L.3, 10.L.3<br>9.L.6, 10.L.6 |
| CSTA<br>Computer<br>Science<br>Standards | L3:CT.1<br>L3:CT.3<br>L3:CPP.2<br>L3:CPP.4<br>L3:CPP.6  | L3:CT.1<br>L3:CT.3<br>L3:CPP.2<br>L3:CPP.4<br>L3:CPP.6  | L3:CT.1<br>L3:CT.3<br>L3:CPP.2<br>L3:CPP.4<br>L3:CPP.6   | L3:CT.1<br>L3:CT.3<br>L3:CPP.2<br>L3:CPP.4<br>L3:CPP.6  | L3:CT.1<br>L3:CT.3<br>L3:CPP.2<br>L3:CPP.4<br>L3:CPP.6  | L3:CT.1<br>L3:CT.3<br>L3:CPP.2<br>L3:CPP.4<br>L3:CPP.6   | L3:CT.1<br>L3:CT.3<br>L3:CPP.2<br>L3:CPP.4<br>L3:CPP.6  | L3:CT.1<br>L3:CT.3<br>L3:CPP.2<br>L3:CPP.4<br>L3:CPP.6   |   |
| UK National<br>Curriculum                | Keystages 3 & 4<br>Computing*   | Keystages 3 & 4<br>Computing*   | Keystages 3 & 4<br>Computing*  | Keystages 3 & 4<br>Computing*   | Keystages 3 & 4<br>Computing*   | Keystages 3 & 4<br>Computing*  | Keystages 3 & 4<br>Computing*   | Keystages 3 & 4<br>Computing*  |   |
| Sample<br>Application of<br>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.   |   |

# JavaScript 1

Grades 7+

## Scope and Sequence

Each lesson takes about one class period to complete.

JavaScript 1 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 -<br>Game Design   | Lesson 10 -<br>Snake   | Lesson 11 -<br>Breakout  | Lesson 12 -<br>Pong  | Lesson 13 -<br>Final Game  |  |  |  |  |
|---------------------------------|---|--|--|--|--|--|--|--|--|
| Key Skills and Concepts         | <ul style="list-style-type: none"> <li>Use game loops, win/loss conditions, and keeping score functionality</li> <li>Use collision detection</li> </ul> | <ul style="list-style-type: none"> <li>Use HTML and JavaScript syntax and functions</li> <li>Design and create a game on the canvas</li> </ul> | <ul style="list-style-type: none"> <li>Use HTML and JavaScript syntax and functions</li> <li>Design and create a game on the canvas</li> </ul> | <ul style="list-style-type: none"> <li>Use HTML and JavaScript syntax and functions</li> <li>Design and create a game on the canvas</li> </ul> | <ul style="list-style-type: none"> <li>Use HTML and JavaScript syntax and functions</li> <li>Design and create a game on the canvas</li> </ul> | <ul style="list-style-type: none"> <li>Use HTML and JavaScript syntax and functions</li> <li>Design and create a game on the canvas</li> </ul> |  |  |  |
| 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>HSA.REI.1   | 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>HSA.REI.1  | 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>HSA.REI.1  | 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>HSA.REI.1  | 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>HSA.REI.1  | 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>HSA.REI.1  |  |  |  |
| CCSS-ELA Standards              | 9.RI.3, 10.RI.3<br>9.RI.6, 10.RI.6<br>9.RI.10, 10.RI.10<br>9.L.3, 10.L.3<br>9.L.6, 10.L.6   | 9.RI.3, 10.RI.3<br>9.RI.6, 10.RI.6<br>9.RI.10, 10.RI.10<br>9.L.3, 10.L.3<br>9.L.6, 10.L.6  | 9.RI.3, 10.RI.3<br>9.RI.6, 10.RI.6<br>9.RI.10, 10.RI.10<br>9.L.3, 10.L.3<br>9.L.6, 10.L.6  | 9.RI.3, 10.RI.3<br>9.RI.6, 10.RI.6<br>9.RI.10, 10.RI.10<br>9.L.3, 10.L.3<br>9.L.6, 10.L.6  | 9.RI.3, 10.RI.3<br>9.RI.6, 10.RI.6<br>9.RI.10, 10.RI.10<br>9.L.3, 10.L.3<br>9.L.6, 10.L.6  | 9.RI.3, 10.RI.3<br>9.RI.6, 10.RI.6<br>9.RI.10, 10.RI.10<br>9.L.3, 10.L.3<br>9.L.6, 10.L.6  |  |  |  |
| CSTA Computer Science Standards | L3:CT.1<br>L3:CT.3<br>L3:CPP.2<br>L3:CPP.4<br>L3:CPP.6  | L3:CT.1<br>L3:CT.3<br>L3:CPP.2<br>L3:CPP.4<br>L3:CPP.6   | L3:CT.1<br>L3:CT.3<br>L3:CPP.2<br>L3:CPP.4<br>L3:CPP.6   | L3:CT.1<br>L3:CT.3<br>L3:CPP.2<br>L3:CPP.4<br>L3:CPP.6   | L3:CT.1<br>L3:CT.3<br>L3:CPP.2<br>L3:CPP.4<br>L3:CPP.6   | L3:CT.1<br>L3:CT.3<br>L3:CPP.2<br>L3:CPP.4<br>L3:CPP.6   |  |  |  |
| 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".   |  |  |  |  |



# Python 1

# Grades 7+

## Scope and Sequence

Each lesson takes about one class period to complete.

Python 1 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 -<br>The Basics  | Lesson 2 -<br>Loops and<br>Patterns   | Lesson 3 -<br>Conditional<br>Logic   | Lesson 4 -<br>Conditional<br>Loops   | Lesson 5 -<br>Variables   | Lesson 6 -<br>Expressions  | Lesson 7 -<br>Turtle Tool   | Lesson 8 -<br>User<br>Interaction  |   |
|--|---|---|--|--|---|--|---|--|---|
| Key Skills and<br>Concepts               | <ul style="list-style-type: none"> <li>• Create custom sequences to solve puzzles</li> <li>• Use function calls</li> <li>• Use Python syntax, naming conventions, and comments</li> </ul> | <ul style="list-style-type: none"> <li>• Use simple, nested, and “for” loops</li> <li>• Use arithmetic operators</li> <li>• Recognize patterns</li> </ul> | <ul style="list-style-type: none"> <li>• Distinguish between assignment, comparison, and logical operators</li> <li>• Use conditional logic to program algorithms</li> </ul> | <ul style="list-style-type: none"> <li>• Use “while” and “do-while” loops</li> <li>• Use conditional loops to solve puzzles</li> <li>• Distinguishing between “for” and “while” loops</li> </ul> | <ul style="list-style-type: none"> <li>• Use variable declarations and assignments</li> <li>• Use and define strings</li> <li>• Use variables to solve puzzles</li> </ul> | <ul style="list-style-type: none"> <li>• Write and understand Python expressions</li> <li>• Use operator precedence to evaluate an expression</li> </ul> | <ul style="list-style-type: none"> <li>• Use turtle graphics</li> <li>• Create animations using turtle graphics and Python</li> </ul> | <ul style="list-style-type: none"> <li>• Set up and handle keyboard and mouse events</li> <li>• Find the correct key code to handle a specific keyboard input</li> </ul> |   |
| CCSS-Math<br>Standards                   | HSN.Q.A.1<br>HSN.Q.A.2<br>HSN.Q.A.3   | 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>HSA.REI.1   | 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>HSA.REI.1  | 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>HSA.REI.1  | 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  | 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   | 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>HSA.REI.1   | -  |   |
| CCSS-ELA<br>Standards                    | 9.RI.3, 10.RI.3<br>9.RI.6, 10.RI.6<br>9.RI.10, 10.RI.10<br>9.L.3, 10.L.3<br>9.L.6, 10.L.6   | 9.RI.3, 10.RI.3<br>9.RI.6, 10.RI.6<br>9.RI.10, 10.RI.10<br>9.L.3, 10.L.3<br>9.L.6, 10.L.6   | 9.RI.3, 10.RI.3<br>9.RI.6, 10.RI.6<br>9.RI.10, 10.RI.10<br>9.L.3, 10.L.3<br>9.L.6, 10.L.6  | 9.RI.3, 10.RI.3<br>9.RI.6, 10.RI.6<br>9.RI.10, 10.RI.10<br>9.L.3, 10.L.3<br>9.L.6, 10.L.6  | 9.RI.3, 10.RI.3<br>9.RI.6, 10.RI.6<br>9.RI.10, 10.RI.10<br>9.L.3, 10.L.3<br>9.L.6, 10.L.6   | 9.RI.3, 10.RI.3<br>9.RI.6, 10.RI.6<br>9.RI.10, 10.RI.10<br>9.L.3, 10.L.3<br>9.L.6, 10.L.6  | 9.RI.3, 10.RI.3<br>9.RI.6, 10.RI.6<br>9.RI.10, 10.RI.10<br>9.L.3, 10.L.3<br>9.L.6, 10.L.6   | 9.RI.3, 10.RI.3<br>9.RI.6, 10.RI.6<br>9.RI.10, 10.RI.10<br>9.L.3, 10.L.3<br>9.L.6, 10.L.6  | 9.RI.3, 10.RI.3<br>9.RI.6, 10.RI.6<br>9.RI.10, 10.RI.10<br>9.L.3, 10.L.3<br>9.L.6, 10.L.6 |
| CSTA<br>Computer<br>Science<br>Standards | L3:CT.1<br>L3:CT.3<br>L3:CPP.2<br>L3:CPP.4<br>L3:CPP.6  | L3:CT.1<br>L3:CT.3<br>L3:CPP.2<br>L3:CPP.4<br>L3:CPP.6  | L3:CT.1<br>L3:CT.3<br>L3:CPP.2<br>L3:CPP.4<br>L3:CPP.6   | L3:CT.1<br>L3:CT.3<br>L3:CPP.2<br>L3:CPP.4<br>L3:CPP.6   | L3:CT.1<br>L3:CT.3<br>L3:CPP.2<br>L3:CPP.4<br>L3:CPP.6  | L3:CT.1<br>L3:CT.3<br>L3:CPP.2<br>L3:CPP.4<br>L3:CPP.6   | L3:CT.1<br>L3:CT.3<br>L3:CPP.2<br>L3:CPP.4<br>L3:CPP.6  | L3:CT.1<br>L3:CT.3<br>L3:CPP.2<br>L3:CPP.4<br>L3:CPP.6   |   |
| UK National<br>Curriculum                | Keystages 3 & 4<br>Computing*   | Keystages 3 & 4<br>Computing*   | Keystages 3 & 4<br>Computing*  | Keystages 3 & 4<br>Computing*  | Keystages 3 & 4<br>Computing*   | Keystages 3 & 4<br>Computing*  | Keystages 3 & 4<br>Computing*   | Keystages 3 & 4<br>Computing*  |   |
| Sample<br>Application of<br>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.   |   |

# Python 1

Grades 7+

## Scope and Sequence

Each lesson takes about one class period to complete.

Python 1 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 -<br>Game Design   | Lesson 10 -<br>Snake   | Lesson 11 -<br>Connect 4   | Lesson 12 -<br>Tetris  | Lesson 13 -<br>Final Game  |  |  |  |  |
|---------------------------------|---|--|--|--|--|--|--|--|--|
| Key Skills and Concepts         | <ul style="list-style-type: none"> <li>Use game loops, win/loss conditions, and keeping score functionality</li> <li>Use collision detection</li> </ul> | <ul style="list-style-type: none"> <li>Use turtle graphics, Python syntax, and functions</li> <li>Design and implement a game</li> </ul> | <ul style="list-style-type: none"> <li>Use turtle graphics, Python syntax, and functions</li> <li>Design and implement a game</li> </ul> | <ul style="list-style-type: none"> <li>Use turtle graphics, Python syntax, and functions</li> <li>Design and implement a game</li> </ul> | <ul style="list-style-type: none"> <li>Use turtle graphics, Python syntax, and functions</li> <li>Design and implement a game</li> </ul> | <ul style="list-style-type: none"> <li>Use turtle graphics, Python syntax, and functions</li> <li>Design and implement a game</li> </ul> |  |  |  |
| 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>HSA.REI.1   | 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>HSA.REI.1  | 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>HSA.REI.1  | 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>HSA.REI.1  | 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>HSA.REI.1  | 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>HSA.REI.1  |  |  |  |
| CCSS-ELA Standards              | 9.RI.3, 10.RI.3<br>9.RI.6, 10.RI.6<br>9.RI.10, 10.RI.10<br>9.L.3, 10.L.3<br>9.L.6, 10.L.6   | 9.RI.3, 10.RI.3<br>9.RI.6, 10.RI.6<br>9.RI.10, 10.RI.10<br>9.L.3, 10.L.3<br>9.L.6, 10.L.6  | 9.RI.3, 10.RI.3<br>9.RI.6, 10.RI.6<br>9.RI.10, 10.RI.10<br>9.L.3, 10.L.3<br>9.L.6, 10.L.6  | 9.RI.3, 10.RI.3<br>9.RI.6, 10.RI.6<br>9.RI.10, 10.RI.10<br>9.L.3, 10.L.3<br>9.L.6, 10.L.6  | 9.RI.3, 10.RI.3<br>9.RI.6, 10.RI.6<br>9.RI.10, 10.RI.10<br>9.L.3, 10.L.3<br>9.L.6, 10.L.6  | 9.RI.3, 10.RI.3<br>9.RI.6, 10.RI.6<br>9.RI.10, 10.RI.10<br>9.L.3, 10.L.3<br>9.L.6, 10.L.6  |  |  |  |
| CSTA Computer Science Standards | L3:CT.1<br>L3:CT.3<br>L3:CPP.2<br>L3:CPP.4<br>L3:CPP.6  | L3:CT.1<br>L3:CT.3<br>L3:CPP.2<br>L3:CPP.4<br>L3:CPP.6   | L3:CT.1<br>L3:CT.3<br>L3:CPP.2<br>L3:CPP.4<br>L3:CPP.6   | L3:CT.1<br>L3:CT.3<br>L3:CPP.2<br>L3:CPP.4<br>L3:CPP.6   | L3:CT.1<br>L3:CT.3<br>L3:CPP.2<br>L3:CPP.4<br>L3:CPP.6   | L3:CT.1<br>L3:CT.3<br>L3:CPP.2<br>L3:CPP.4<br>L3:CPP.6   |  |  |  |
| 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