# EasyCode Pillars Scope and Sequence

EasyCode Pillars is an online, interactive curriculum that teaches students coding principles using the Python programming language. In EasyCode Pillars, students use the Codesters coding environment to learn key coding principles and Python syntax through a series of activities in each lesson. Activities include small coding challenges, debugging practice and short quizzes to test understanding. Each lesson ends with a coding exercise in which students create their own project applying the skills learned. In addition to basic coding principles, students learn how to animate objects, play sounds and use mouse and keyboard input. The curriculum includes detailed lesson plans and other scaffolding to ensure teachers can teach coding concepts even if they have no previous experience with coding themselves.

**EasyCode Pillars is divided into four mini-courses. The following is an overview of the scope of sequence of each of these mini-courses:**

## Intro to Codesters

Intro to Codesters is five lessons long and provides a brief overview of the coding environment in which students will be working. It is included as part of EasyTech's Computational Thinking unit.

| CURRICULUM TITLE | DESCRIPTION | KEY TOPICS | TIME (MINUTES) |
|---|---|---|---|
| Building Your First Program | In this introductory lesson, students will learn to use the Codesters environment to create their first program by creating a scene with a background, sprite, text, movement, and interactivity. | backgrounds<br>sprites<br>click events<br>debugging | 90 |
| Codesters in Space | In this lesson, students will learn to use the (X,Y) Coordinate Plane to move a sprite around a stage that uses the background as context for creating a short animated scene. | placing objects<br>parameters<br>debugging | 90 |
| Shapes and Drawings | In this lesson, students will learn to change the variable names and parameters of shape objects to create a picture of a house. | variables<br>parameters | 90 |
| Chat with Your Sprite | In this lesson, students will create an interactive program that uses user input to control elements of the program. The sprite will appear to have a dialogue with the user. | using text<br>user input<br>variables<br>conditionals | 90 |
| Recycling Loop | In this lesson, students will use loops, lists and variables to understand looping behavior. | variables<br>lists<br>loops | 90 |

## Intro to Python - Part 1

Intro to Python Part 1 contains 20 lessons and begins to cover more programming concepts in the Python language. It covers topics such as variables, loops, if statements, lists, functions and events.

| CURRICULUM TITLE | DESCRIPTION | KEY TOPICS | TIME (MINUTES) |
|---|---|---|---|
| Dance Steps | In this initial lesson, students will begin to understand order in programming and learn to troubleshoot when the instructions within a program are out of order. | sequencing<br>debugging | 90 |
| Robot Design | In this lesson, students will understand that parameters define shapes and they will learn to define a shape using parameters. | parameters<br>strings<br>integers | 90 |
| Dialogue | In this lesson, students will learn how to use dot notation to organize the actions of multiple sprites. | dot notation<br>sequencing | 90 |
| Roll the Dice | In this lesson, students will learn how to create and apply a variable to their program. | variables | 90 |
| Star Variables | In this lesson, students will create and utilize variables to make a program more efficient. | variables<br>data types | 90 |
| My Friends | In this lesson, students will combine two values of the same data type to communicate a story. | variables<br>string concatenation | 90 |
| Math and Computation | In this lesson, students will learn to create a calculator to determine the area of a shape. | variables<br>math operators<br>converting data types | 90 |
| Bonus Lesson | In this lesson, students will show mastery of early programming standards. | (review) | 90 |
| Spiral | In this lesson, students will learn to use a *for* loop to repeat commands. | *for* loops<br>loop ranges | 90 |
| Variables and Loops | In this lesson, students will create and store temporary values as they change it through iterations in a loop. | *for* loops<br>iterating variables | 90 |
| User Input | In this lesson, students will collect and store user input to be used in a story. | user input<br>string concatenation | 90 |
| Intro to If Statements | In this lesson, students will use *if* statements to generate a story based on user input. | user input<br>*if* statements | 90 |
| Tim the Wizard | In this lesson, students will learn to randomize a response to user questions using *if* statements and random number generation. | user input<br>*if* statements<br>random number generator | 90 |
| Begin Rock, Paper, Scissors | In this lesson, students will create a program that accepts and displays user input, and shows a randomly generated computer response. | user input<br>*if* statements<br>random number generator | 90 |
| Who Won Rock, Paper Scissors? | In this lesson, students will compare user answers to randomly generated computer answers and have their programs determine a win, loss, or tie from these comparisons. | comparing user input<br>testing code | 90 |
| Lists | In this lesson, students will create and structure lists using list operations and getters. | lists | 90 |

| CURRICULUM TITLE | DESCRIPTION | KEY TOPICS | TIME (MINUTES) |
|---|---|---|---|
| Guess the Number | In this lesson, students will construct a guessing game by building a *for* loop with a counter in range. | *for* loops<br>counters<br>ranges<br>break | 90 |
| Password | In this lesson, students will learn to create a guessing game by building a loop that accesses each value in a list. | initializing variables<br>looping through a list | 90 |
| Midterm: Which Sprite | Students will learn to create a quiz program by synthesizing and applying the programming concepts mastered during Intro to Python Part 1. | creating models (review) | 90 |
| Midterm: Create Your Own Quiz | In this lesson, students will plan and build a unique quiz program synthesizing the programming concepts mastered during Intro to Python part 1. | (review) | 90 |

## Intro to Python - Part 2

Intro to Python Part 2 contains 20 lessons and continues where Part 1 left off. It covers functions, indexes, events, data types, program design and game mechanics.

| CURRICULUM TITLE | DESCRIPTION | KEY TOPICS | TIME (MINUTES) |
|---|---|---|---|
| Changing Scenes | In this lesson, students will learn to write and call simple functions without parameters. | writing functions<br>calling functions | 90 |
| Dance Moves | In this lesson, students will write functions and call them while paying attention to the order in which functions are called. | writing functions<br>calling functions | 90 |
| All Mixed Up | In this lesson, students will use indexes to reference lists and reorganize lists. | lists<br>indexes | 90 |
| Party Invitations | In this lesson, students will create functions with parameters and use them to create a format for a party invitation. | functions with parameters | 90 |
| Secret Code | In this lesson, students will create an encoder that takes in messages and outputs a coded message. | manipulating lists | 90 |
| Around the Solar System | In this lesson, students will calculate the weight of objects as a percentage of their weight on earth using returns and functions with parameters. | floats<br>functions with return values | 90 |
| Dance Off | In this lesson, students will use events to create a player-controlled dance routine. | creating events and event handlers | 90 |
| Astronaut Rescue | In this lesson, students will use collision events to define game play. They will use event handlers to call specific sprites and assign them different roles in collisions. | collisions<br>event handlers | 90 |
| Save the Moon! | In this lesson, students will learn that an interval event is called not by a user action, but by a specific amount of time passing. | global variables<br>interval events<br>keeping score | 90 |
| Whack-a-Shark | In this lesson, students will use interval events, random choice, and click events to create a Whack-a-Mole style game. | random number generator<br>win conditions | 90 |

| CURRICULUM TITLE | DESCRIPTION | KEY TOPICS | TIME (MINUTES) |
|---|---|---|---|
| Dino Dodge | In this lesson, students will create a game that uses a mouse move and interval event to control game play. They will use collisions and changing variable values to determine game length. Students set physics values to control movement of particular objects on the stage. | game physics timers | 90 |
| Refactoring | In this lesson, students will learn to revise code and make use of new structures like functions and loops. | refactoring code | 90 |
| Turtle Traffic | In this lesson, students will learn to write functions for use in events. | functions | 90 |
| Zombie Kitten Attack! | In this lesson, students will learn to create a side-scroller game where the user controls the background movement to give the appearance of constant movement. | stage physics loops | 90 |
| Rocky Road | In this lesson, students will create a game that uses probability to generate hazards and goals. They will use interval events and falling objects to give the appearance of motion. | game physics probability of events | 90 |
| Final Project - Day 1 | This is the first in a five-lesson final project in which students will design, build, test and refine a game or an interactive story. In this part, students will reflect on the concepts learned previously and brainstorm ideas for a final project in small groups. | final project: brainstorm | 90 |
| Final Project - Day 2 | In this second lesson for the final project, students will create a paper prototype of their idea for a final project and gather feedback on it. | final project: paper prototype | 90 |
| Final Project - Day 3 | In this third lesson for the final project, students will begin to create their game or interactive story in the Codesters environment. | final project: first draft | 90 |
| Final Project - Day 4 | In this fourth lesson for the final project, students will continue working on their game or interactive story. They will spend time testing their solution, reviewing another student's project, and providing feedback on the other student's project. | final project: final build and testing | 90 |
| Final Project - Day 5 | In this final lesson for the final project, students will present their game or interactive story to the class. They will also reflect on what they have learned throughout the class and project. | final project: present, play, and reflect | 90 |

## Intro to Game Design

Intro to Game Design has 10 lessons and helps students apply the skills learned in Intro to Python Part 1 to create a different type of game in each lesson. Intro to Game Design covers topics such as taking turns, moving objects, collisions, random numbers, avoiding hazards, and getting points.

| CURRICULUM TITLE | DESCRIPTION | KEY TOPICS | TIME (MINUTES) |
|---|---|---|---|
| Adventures of Star Hedgehog | In this lesson, students will use event keys, collision events, a global timer and a scoreboard to create a game where a character collects goal objects. | key events variable scope collision events timer and scoring | 90 |
| Earth Day | In this lesson, students will use lists and random choice commands to create sprites with random images to use as goals and hazards in their game. | lists of strings random number generator game hazards | 90 |

| CURRICULUM TITLE | DESCRIPTION | KEY TOPICS | TIME (MINUTES) |
|---|---|---|---|
| Feed the Fish | In this lesson, students will use the physics toolkit to create a falling-object type of game. | game physics<br>leaving the stage | 90 |
| Flappy Dino | In this lesson, students will use the physics toolkit, interval-event, and random number generators to create a flappy-bird style, perpetual motion game. | game physics | 90 |
| Alien Scroller | In this lesson, students will create an extra-long stage background to create a side scrolling type of game. | scrolling stage<br>goals and hazards<br>win conditions | 90 |
| Brick Breaker | In this lesson, students will recreate the classic brick-breaker-style game using physics, random speed settings, and a collision event. | nested loops<br>collision events | 90 |
| Basketball | In this lesson, students will create a pull-and-shoot style game using stage physics and event keys. | += iteration<br>key events | 90 |
| Pictionary | In this lesson, students will create a Pictionary-style drawing game for two players. | mouse events<br>lists<br>drawing | 90 |
| Sprite Volleyball | In this lesson, students will create a two-player volleyball/pong-style game that uses key events, stage physics and boundaries as hazards. | two-player games | 90 |
| Codebreaker | In this lesson, students will create their own version of a Mastermind-style board game. | nested loops<br>*while* loops<br>defining functions | 90 |