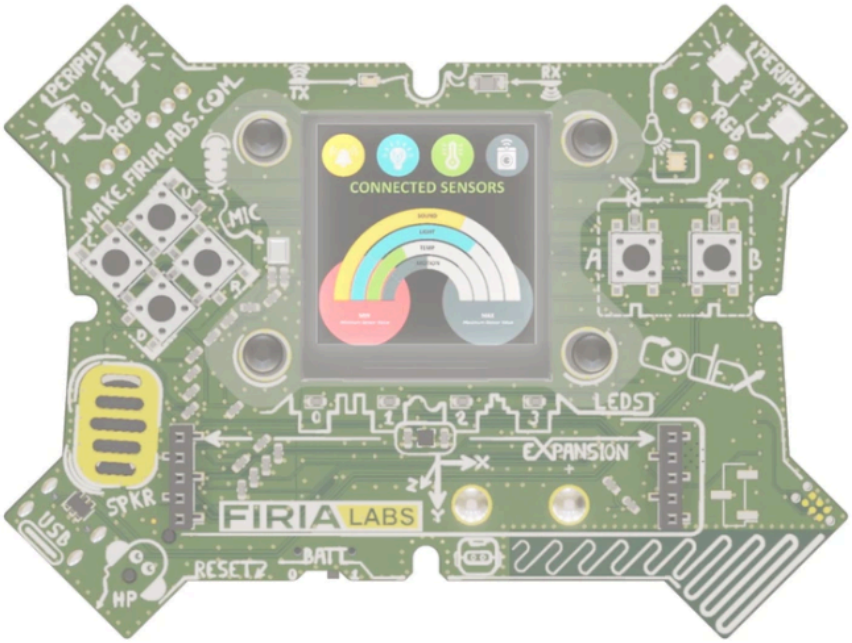
Python with CodeX – TEKS Computer Science 1 Curriculum

The curriculum for Computer Science 1 consists of programming CodeX projects and completing lessons in computer science topics. The CodeX is a specialized micro:bit with integrated speaker, accelerometer, LEDs, display screen and several buttons for input. Using a physical device for coding dramatically increases engagement and interest over traditional computer science instructional methods that focus on math problems or manipulating on-screen elements.

Our educational program creates real-world learning experiences for students. This is achieved through the use of:

- Open-ended physical hardware, used to implement meaningful projects
- Open-ended software, integrating development tools with instructional content, with the possibility of students to directly apply the tools well beyond the scope of what is covered in the curriculum
- Real music, colorful graphics and bright and colorful LEDs
- LEDs for status display of all systems, and console log for printing
- Step by step guided lessons in CodeSpace
- Python, the fastest growing major programming language used in Industry
- Big data and machine learning connections



All standards are met by completing the required missions and units. The amount of time needed to complete the curriculum is flexible. It is recommended that students spend at least 30-45 per class period.

- The CodeX missions should be completed in order.
- The additional units can be completed in any order, and at any time.
- Optional coding missions are available as time permits, but are not required to meet the standards.

Teachers have the option of completing the missions and then the additional units, or intermixing coding with additional computer science content.

Python with CodeX – Computer Science 1 Curriculum

Course Coding Projects

Mission	Outline of lesson	Standards
Computer Science Overview *5-7 class periods*	**Intro to Computer Science, CodeBot and CodeSpace** The project allows for time to get to know your students, assess their prior knowledge, and build a foundation of computer science basics. During this project you can guide your students to building a foundation of computational thinking. Dedicate some time for students to learn basic terms, such as algorithm, program and debug. You can also engage students in unplugged activities.	(1) C, D (2) A (6) A, B, C, K, M
Mission 1 / Mission 2 1-2 *class periods*	**Welcome to CodeSpace** The mission starts with an introduction to computer science. Then students log in to CodeSpace. They open the toolbox, which contains documentation, and it is used as a resource for debugging. Students will complete a worksheet or journal on the vocabulary, tools, etc. that they learn during the lesson. **Introducing CodeX** The mission starts with discussing how coding is used in many things, not just robotos. Key vocabulary words are introduced, like input, output and CPU. Students learn more about CodeSpace and also are introduced to CodeX. The lesson gives students an opportunity to transfer current knowledge to the learning of newly encountered technology.	(1) C, D (2) A (3) C (6) A, B, C
Mission 3 *1-2 class periods*	**Light Show** The mission discusses how LEDs show colors with RGB. Abstraction is introduced through the use of parameters for turning on pixels. The debugger is introduced. Variables and editor shortcuts are used for the first time. The final mission involves lighting up the four pixel LEDs in various colors.	(1) C, D (2) A (3) C (4) J, Q (6) A, D, N, O, R
Remix Mission 3 *1-2 class periods*	**Light Show Remix** For this project students will use what they have learned from mission 3 to create their own original program. Suggestions for a remix are included at the end of mission 3, or students can be creative and come up with their own ideas. Another suggestion is for students to work in teams and work on employability skills, like time management, leadership, planning, and communication. Students should seek feedback during their remix, and present their project in a variety of ways such as print, monitor display, web pages, or video.	(1) C, D, E, F (2) A, B (3) C (4) F, H, J, M, Q, V (6) A, D, N, O, R

Mission 4 *1-3 class periods*	**Display Games** Some data types are introduced, and the need for converting from one data type to another is demonstrated. Branching with Boolean conditions is used for the first time. The final mission will show a variety of pre-made pictures on the display screen.	(1) C, D, E (2) A (3) A, B, C, D (4) A, B, D, E, H, I, J, Q, R (6) A, D, N, O, R
Mission 5 *1-2 class periods*	**Micro Musician** The mission shows how the CodeX can play any of the several audio files that are pre-loaded on the CodeX. As a warm-up activity, discuss how today's musicians use technology. Code readability is also stressed, including using blank lines and adding comments. Optional: Assign the CodeX & MP3s lesson next	(1) C, D, E (2) A (3) C (4) H, J, Q (6) A, D, N, O, R
Remix Mission 4/5 *2-4 class periods*	**Display Games / Micro Musician Remix** For this project students will use what they have learned from missions 3-5 to create their own original program. Suggestions for a remix are included at the end of missions 4 and 5, or students can be creative and come up with their own ideas. Another suggestion is for students to work in teams and work on employability skills, like time management, leadership, planning, and communication. Students should seek feedback during their remix, and present their project in a variety of ways such as print, monitor display, web pages, or video.	(1) C, D, E, F, H (2) A, B (3) A, B, C, D (4) A, B, D, E, F, G, H, I, J, M, Q, R (6) A, D, N, O, R
Mission 6 *2-3 class periods*	**Heartbeat** The mission introduces loops and the "kill switch" with a button press. The float data type is used, and shortcut keys and Boolean conditions are continued. Using the debugger is practiced. The final mission will show a heartbeat that can change speed faster or slower. It will end with a possible error! that they can go back and fix later.	(1) C, D, E (2) A (3) B, C (4) H, I, J, K, L, N, P, R, S (6) D, J, N, O, R
Mission 7 *2-3 class periods*	**Personal Billboard** The mission introduces lists and tuples. They must navigate data conversion and use Boolean conditions to help them code the correct display commands. The use of CodeTrek and the debugger is emphasized. The final mission will display a variety of images, text and color on the screen. Optional: Assign the CodeX & JPGs next	(1) C, D, E (2) A (3) A, B, C, D (4) D, H, I, J, K, L, N, P, R, S, T (6) D, F, J, L, N, O, Q, R
Remix Mission 6/7 *2-3 class periods*	**Heartbeat / Personal Billboard Remix** For this project students will use what they have learned from missions 3-7 to create their own original program. Suggestions for a remix are included at the end of mission 6 and 7, or students can be creative and come up with their own ideas. Another suggestion is for students to work in teams and work on employability skills, like time management, leadership, planning, and communication. Students should seek feedback during their remix, and present their project in a variety of ways such as print, monitor display, web pages, or video.	(1) C, D, E, F, H (2) A, B (3) A, B, C, D (4) A, B, D, E, F, G, H, I, J, K, L, M, N, P, R, S, T (6) D, F, J, L, N, O, Q, R
Mission 8	**Answer Bot** The mission is similar to the personal billboard, but it will show another way to	(1) C, D, E (2) A

2-3 class periods	display text. Random numbers are introduced and used. Students practice debugging by making intentional errors. The mission also uses lists. The final mission will display random answers on the screen and colors on the LEDs.	(3) A, B, C, D (4) D, H, J, K, L, N, P, R, S, V (6) D, F, N, O, P, Q, R
Remix Mission 8 *2-3 class periods*	**Answer Bot Remix** For this project students will use what they have learned from missions 2-8 to create their own original program. Suggestions for a remix are included at the end of mission 8, or students can be creative and come up with their own ideas. Another suggestion is for students to work in teams and work on employability skills, like time management, leadership, planning, and communication. Students should seek feedback during their remix, and present their project in a variety of ways such as print, monitor display, web pages, or video.	(1) C, D, E, F, H (2) A, B (3) A, B, C, D (4) A, B, D, E, F, G, H, J, K, L, M, N, P, R, S, T, V (6) D, F, N, O, P, Q, R
Mission 9 *2-5 class periods*	**Game Spinner** The mission uses a built-in list for arrow images. Logical operators are introduced and used. Abstraction is emphasized, and students create their own functions with parameters. The program uses a loop with a counter condition, modifies the counter, and identifies local variables. Students also learn how to use the debug console. The final mission has the CodeX act as an authentic spinner for a game by pressing a button.	(1) C,D, E (2) A (3) B, C (4) C, D, E, H, J, K, L, N, P, R, S, T, U, V (6) D, E, F, G, I, J, N, O, Q, R
Remix Mission 9 *3-5 class periods*	**Game Spinner Remix** For this project students will use what they have learned from missions 3-9 to create their own original program. Suggestions for a remix are included at the end of mission 9, or students can be creative and come up with their own ideas. Another suggestion is for students to work in teams and work on employability skills, like time management, leadership, planning, and communication. Students should seek feedback during their remix, and present their project in a variety of ways such as print, monitor display, web pages, or video.	(1) C,D, E (2) A (3) B, C (4) A, B, C, D, E, F, G, H, J, K, L, M, N, P, R, S, T, U, V (6) D, E, F, G, I, J, N, O, Q, R
Mission 10 *2-4 class periods*	**Reaction Tester** The mission introduces students to the internal clock. Using a loop and accessing the CodeX clock, students learn how to record a reaction time. Students create their own functions and use them during this mission. The final mission is a reaction game by recording the reaction time to pressing a button after a LED is lit.	(1) C, D, E (2) A (3) A, B, C, D (4) C, D, H, I, J, K, N, O, P, R, S, T, V (6) D, F, I, J, L, N, O, P, R
Mission 11 *2-4 class periods*	**Spirit Level** The mission introduces the CodeX's accelerometer, which tracks the CodeX position in three dimensions. Students will read data from the accelerometer and store the information in a tuple. Students will also learn how to draw simple graphics, like a line and a circle. The final mission will use the display screen as a level, with a circle as the bubble.	(1) C, D, E (2) A (3) B, C (4) C, D, H, I, J, K, N, O, P, R, S, T (6) D, J, N, O, P, R
Mission 12	**Night Light** The mission introduces the light sensor embedded on the CodeX. Students	(1) C, D, E (2) A

1-3 class periods	use math to manipulate the data from the sensor into useful information. The final mission will light up the display screen like a night light when the light sensor reads dim light.	(3) C (4) C, D, H, I, J, K, N, O, P, R, S, T (6) D, J, N, O, R
Remix Mission 10/11/12 *3-5 class periods*	**Reaction Tester / Spirit Level / Night Light Remix** For this project students will use what they have learned from all the missions to create their own original program. Suggestions for a remix are included at the end of missions 10, 11 and 12, or students can be creative and come up with their own ideas. Another suggestion is for students to work in teams and work on employability skills, like time management, leadership, planning, and communication. Students should seek feedback during their remix, and present their project in a variety of ways.	(1) C, D, E, F, H (2) A, B (3) A, B, C, D (4) A, B, C, D, E, F, G, H, I, J, K, M. N, O, P, R, S, T, V (6) D, I, J, L, N, O, R
Mission 13 *3-5 class periods*	**Sounds Fun** In this mission students will create a user-friendly graphical interface and explore the “soundlib” library for CodeX sound effects. Students will learn how to play sounds and music in the background while other code is running, make sound effects for games and user feedback, and control the pitch and loop sounds.	(1) C, D, E (2) A (3) A, B. C, D, E (4) A, B, C, D, E, F, G, H, J, K, L, M, N, O, P, R, S, T, U (6) A, D, E, G, I, N, O, P, Q, R
Mission 14 *3-5 class periods*	**Line Art** In this mission students will create beautiful visual art and learn about the magic of computer graphics with just a few lines of code and the power of the pixel. Students will learn more about the bitmap object, x & y coordinates of the display, line drawing functions, drawing an envelope, and using a for loop.	(1) C, D, E (2) A (3) B, C, E (4) A, B, C, D, F, G, H, I< J, K, M, N, O, P, R, S (6) D, G, I, N, O, R
Mission 15 *3-5 class periods*	**Handball** The mission is the first of a 2-part mission sequence to develop a retro video game of *American Handball.* Students will build a handheld gaming framework, culminating in a fun, playable 1-player game version of the classic "Pong". Buttons move a paddle side-to-side across the bottom of the screen. A ball bounces off the sides and top of the screen. Score points by hitting the ball with your paddle. You get 3 "lives" - lose those balls and it's GAME OVER!	(1) C, D, E (2) A (3) A, B, C, D, E (4) A, B, C, D, F, G, H, I, , K, L, M, N, O, P, R, S, T, U, V (6) D, E, F, G, H, I, J, L, N, O, P, Q, R
Mission 16 *3-5 class periods*	**Break Out** The mission starts where the previous Handball mission left off. The game Breakout adds 8 rows of bricks. By the end of this mission the player will be able to score points by smashing bricks! Different color bricks are worth different points.	(1) C, D, E (2) A (3) A, B, C, D, E (4) A, B, C, D, E, F, G, H, I, , K, L, M, N, O, P, R, S, T, U, V (6) D, E, F, G, H, I, J, L, N, O, P, Q, R

Additional Computer Science Lessons

Lesson	Outline of lesson	Standards
Technology and Digital Information *5-10 class periods*	This lesson will teach students about important technology operations, concepts, systems, and operations as they apply to computer science. Basic computer components, such as storage and peripheral devices, will be studied. Students will learn about different operating systems and describe the differences between an application and an operating system. Students will also review system tools, including appropriate file management. Binary numbers will be practiced, allowing students to understand how data is represented in a computer system, convert between binary and decimal numbers, and count in binary. Students will have an opportunity to discuss and give examples of the impact of computing and computing-related advancements on society. **Final Project:** Create a project that summarizes their learning or extends their learning on a specific concept. Students will then publish their project, which could include print, monitor display, web pages, or video.	(1) C, D, F, H (2) A, B (6) A, B, C, D, K, L, M, O, S
Design Process and Flowcharts *2-4 class periods*	This lesson will go through the five steps of the design process and gives context into what it is and why they will use it. It introduces flowcharts and engages students in creating their own. The main flowchart symbols are explained. In the first part of the lesson, students are given working Python code and create flowcharts from them. In the second part of the lesson, students are given flowcharts and they create working code from them. Pseudocode can also be introduced and practiced during this lesson. **Final Project:** An assessment can be given for checking the understanding of students about the design process and flowcharts. Then students can be expected to create flowcharts and/or pseudocode for their remix projects.	(1) C, D, E, F (2) A, B (4) A, B, C
Computer Science Careers *5-10 class periods*	This project will enable students to identify various employment opportunities in the computer science field. Students will compare university computer science programs and examine the role of certifications, resumes and portfolios in the computer science profession. They will seek to identify job and internship opportunities in computer science and explore career opportunities. Students will also demonstrate an understanding of legal and ethical responsibilities in a computer science career. Students will have an opportunity to discuss and give examples of the impact of computing and computing-related advancements on society. **Final Project:** Create a project that summarizes their learning or extends their learning on a specific concept. This can be a team project, allowing students to work on employability skills, like time management, leadership, planning, and communication. Students will then publish their project, which could include print, monitor display, web pages, or video.	(1) A, B, C, D, F, G, H, I (2) A, B
Digital Citizenship *5-10 class periods*	This project will teach students about digital citizenship. Students explore and understand safety, legal, cultural, and societal issues relating to the use of technology and information. Privacy and copyright laws are discussed, and students learn to ethically find digital information and cite their sources. They	(1) C, D, F, H (2) A, B (5) A, B, C, D, E

	look at acceptable use policies. They learn about keeping their information safe through strong passwords, virus detection and security. Students also analyze how electronic media can affect the reliability of information. Students will have an opportunity to discuss and give examples of the impact of computing and computing-related advancements on society. **Final Project:** Create a project that summarizes their learning or extends their learning on a specific concept. This can be a team project, allowing students to work on employability skills, like time management, leadership, planning, and communication. Students will then publish their project, which could include print, monitor display, web pages, or video.	
Fina Project *5-15 class periods*	The final project can be determined by the teacher and the interests of the students. For example, students could: ● create an original program for the CodeX, then create a video of the CodeX running the code ● Research a computer science topic not yet covered: ○ cyber security ○ how the internet works, or the internet of things ○ artificial intelligence or machine learning ○ digital data and compression ○ data science and representation ○ global impact of computing / future of computing ● Take apart and label the parts of a computer ● Create a presentation or lesson on a computer science topic and teach it to a group of students ● Create a newsletter or video about the class (recruiting tool!)	(1) C, D, E, F, H (2) A, B

Optional Coding Projects

Project	Outline of lesson	Standards
Define and Call Functions *5-8 class periods*	**Define and Call Functions** This lesson has three parts. They can be completed together or spread out during the semester. **Part 1:** User-defined functions are introduced. Students define and call functions to organize their code and reduce repetition. Functions also help the readability of code by naming sections of code that accomplish a task. Abstraction is introduced, which is an important concept in computer science. Students define and call functions in three programs from early missions (missions 3 & 4). **Part 2:** The lesson starts with a review of functions. Students look at previous programs and their functions, all without parameters. Then an example is given that requires a parameter. Throughout the lesson students learn about parameters and local variables and practice recognizing when a value needs to be a parameter or when it can be a local variable. This lesson is appropriate after mission 8 or mission 9. **Part 3:** The last part of the lesson reviews the first two parts and then discusses the need for global variables and how to use them in Python functions. It involves two examples for learning and a coding activity. It should	(1) C, D, E, F (2) A (4) C, D, E (6) E, F, G, H, I

	be taught after mission 9.	
Types of Division *1-3 class periods*	**Types of Division** This lesson specifically emphasizes the different types of division: decimal (or real), integer and modulo division. They practice with math problems and learn applications for modulo division. Other types of math problems, including problems that use built-in math functions, can also be discussed and practiced.	(4) N, O, P, V (6) R
Pixel Art *3-5 class periods*	**Creating pixel (ASCII) images** The CodeSpace editor includes images, similar to ASCII art, but with color. These are the pictures students use starting in Mission 6. Instead of using characters to create an image, students can make a grid of colors for the display pixels to create their own picture. This mission shows students how to take either ASCII art or pixel art and recreate it for the CodeX. This lesson should be given after Mission 7 and/or Mission 8. Final Project: Program the CodeX to display an ASCII (or pixel) image ● Can use and re-color an included image ● Can use an image found on the Internet and adapt for CodeX ● Can create original artwork	(1) C, D, E (2) A (3) A, B, C (4) C, J
CodeX & JPGs *2-4 class periods*	**Transforming images to JPG and using with CodeX** Students may want to display their own images or photos on the CodeX display screen. This mission teaches students the steps to resizing and compressing an image to fit CodeX display requirements. Editing software is needed for this part of the mission. Then students must upload their images to the CodeX. Finally, they use code to display the image. This lesson should be given after Mission 7 and/or Mission 8.	(1) C, D (2) A (3) B, C (4) J
CodeX & MP3s *2-4 class periods*	**Adding audio files to the CodeX** Students may want to play their own audio files in a program like the personal billboard, or to enhance a game. This mission teaches students to record their own audio file and save it in the correct file format. Additional software is needed for this part of the lesson. Then students must upload their audio files to the CodeX. Finally, they use code to play their original files. This lesson should be given after Mission 5.	(1) C, D (2) A (3) B, C (4) J
CodeX and Line Art *3-5 class periods*	**Creating an image with simple shapes** Students may want to create their own simple artwork using lines, rectangles and circles. This lesson teaches students how to draw these shapes, either as an outline or filled in. Graph paper can be used to design a simple graphic. The lesson can be extended by using variables with the simple graphic in a series of loops to create a row of images, a grid of images, an image in a random location, etc.	(1) C, D, E, F, H (2) A, B (3) B, C, E (4) A, B, C, D, F, g, H, J, M, N, P, Q, V (6) E, G, I, N, O, R