MISSION 3: Time and Motion Lesson 1 (Objectives 1-5)	**Time Frame: 30-40 minutes**
Project Goal: Students will control CodeBot LEDs with specific timing and sequencing. **Learning Targets** • I can import a specific built-in function from a library. • I can use the "Step" feature of the CodeSpace debugger. • I can use a variable to make code more efficient. • I can assign a value to a variable. • I can use a variable as an argument in a function call.	**Key Concepts** • Computers execute code in sequential steps. • The CodeSpace debugger lets you *step* through the code one line at a time to understand what the computer is doing. • Built-in functions come from libraries, like botcore or time. • A function call can pass a value, or argument, to the function. • Variables can be defined to hold changing values. Variables make code more efficient and reduce duplication.
Assessment Opportunities • Mission 3 Lesson 1 Log • Submit completed program ***SequenceLEDs*** • Submit the program with extensions • Mission 3 Obj. 1-5 Review Kahoot!	**Success Criteria** ☐ Light up four user LEDs in sequence ☐ Use the "Step In" feature of the CodeSpace debugger ☐ Import the sleep function ☐ Use the sleep function to slow down code ☐ Define a variable ☐ Use a variable in a function call ☐ Create a light show with CodeBot's LEDs
Teacher Materials in Learning Portal • Mission 3 Lesson 1 Slides • Mission 3 Lesson 1 Log • Mission 3 Lesson 1 Answer Key	**Additional Resources** • Mission 3 Obj. 1-5 Review Kahoot! • ***SequenceLEDs*** sample code (learning portal) • ***SequenceLEDs_extensions*** sample code (learning portal)

Vocabulary

- **Physical computing:** Writing code (instructions) for a physical device, like CodeBot or cars
- **Editor shortcuts:** Keyboard hotkeys to write code faster; combinations of keys which complete a task
- **CPU:** The "brain" of the computer that executes your code; the Central Processing Unit
- **Debugging:** The process of understanding what the computer is actually doing and then changing the code to do what you want it to do
- **Argument:** A value that is passed to a function
- **Literal:** An actual value, like 1 or "hello" or True
- **Variable:** A name to which you assign some data, like a number; must be defined before it is used

New Python Code

`from time import sleep`	Import the ability to delay, or pause, the code
`sleep(1.0)`	Use sleep() – will sleep the amount of time in seconds
`delay = 1.0`	Define a variable (define variables near the top just under the imports)

`sleep(delay)`	Use a variable with sleep()
`leds.user_num(2, False)`	Turn off an LED

Real World Applications

You've used some fundamental computer science and robotics principles:

- Controlling LEDs with specific timing and sequencing

This code is used in cars, stage lights, espresso machines, music sequencers, and more!

Teacher Notes:

- Although you cannot copy and paste from CodeTrek, you can copy and paste in the text editor. The program for this lesson has a lot of repetition. Show students editor shortcuts, as needed, to save time.
- This lesson is a good time to discuss common programming practices, such as descriptive variable names and blank lines.
- The extensions for this lesson give students an opportunity to be creative and express themselves through a light show. If you have time for students to do an extension, it is motivational and engaging.

Extensions / Cross-Curricular:

- Make a poster or chart of Python commands.
- Use the line sensor LEDs in the light show.
- Combine user and line sensor LEDs to create a pattern, like turning on the even LEDs and leaving the odd LEDs off.
- **MATH:** Students design a timed light show and use math to determine the specific timings for each LED segment to meet the specified time.
- Supports **language arts** through reading instructions and reflection writing.

Preparing for the lesson:

- Look through the slides and workbook. Decide what materials you want to use for presenting the lesson. The slides can be converted to Google Slides. They can be projected on a large screen. The workbook (if used) can be printed or remain digital through your LMS and given to students.
- Be familiar with the mission log assignment and the questions they will answer. Prepare the assignment to give through your LMS.
- If you have a word wall, or another form of vocabulary presentation, prepare the new terms.

Lesson Tips and Tricks:

💡 **Teaching tip:**

You can use a variety of discussion strategies to get the most engagement from your students. For example, you can have students write their answers before asking anyone for an answer. You can use one of many think-pair-share methods. You can have students write their answer and share with someone, and then have other students share answers they heard from their peers. You can randomly select students to answer.

👫 **Pre-Mission Warm-up:** -- slide 2

Students can write in their log first and then share, or discuss first and then write in their log.

- Question: What code turns on a user LED?
- Question: What code turns off a user LED?
- Students can share their answers, or compare with each other. This code is used during the lesson.

💻 **Mission 3 Lesson 1 Activities:**

The Chrome browser works best, but other browsers also support CodeSpace. Each student will complete a Mission Log. Students could work in pairs through the lesson, or they can work individually.

💡 **Teaching tip: Mission Introduction** -- slides 3-4
The intro gives real-world applications for the concepts used in this lesson. It also lists the six goals for the mission. Only the first three are addressed in this lesson.

💡 **Teaching tip: Objective #1** -- slides 5-6
This step should be review – turning on four LEDs. The instructions mention using editor shortcuts. You can show students how to copy a line of code and paste it in the text editor, just having to change the user LED number for each line of code.

💡 **Teaching tip: Objective #2** -- slides 7-8
This objective shows students how to use the debugger. This can be confusing the first time. You should practice using the debugger and model it for your students.

💡 **Teaching tip: Objective #3** -- slides 9-11
Students add code to their program. They add an import statement near the top, and then three sleep() statements. Verify that they add all three sleep() statements, and not just one. If they add a fourth one after the last leds.user_num() statement, that is fine, too.

💡 **Teaching tip: Objective #4** -- slides 12-14
The variable should be defined just after the import statements. The blank line is not required but makes the code more readable. Make sure each sleep() statement uses the variable.

💡 **Teaching tip: Objective #5** -- slides 15-16
Students add to their code by turning off each LED before turning on the next LED. They can try different values for delay if they want to.

💡 **Teaching tip: Extension** -- slide 17
If you have time, students should do the extension. They need to complete Objective 5 first, so they meet the goal. Then they can continue to modify their code to create an interesting light show.

Optional: 🗝️ Mission 3 Obj 1-5 Kahoot! Review.

A review Kahoot! Is available for these five objectives. You can do the Kahoot together as a class, or assign it independently. (link above)

👫 **Post-Mission Reflection:**

The post-mission reflection asks students to think critically about programming and reflect on what they are learning.

You can use an extension or cross-curricular activity as post-mission activity.

End by collecting the Mission 3 Lesson 1 Log.

SUCCESS CRITERIA:

- ❑ Light up four user LEDs in sequence
- ❑ Use the “Step In” feature of the CodeSpace debugger
- ❑ Import the sleep function
- ❑ Use the sleep function to slow down code
- ❑ Define a variable
- ❑ Use a variable in a function call
- ❑ Create a light show with CodeBot’s LEDs