


<b>LESSON: Create Performance Task - practice #1</b>		<b>Time: 90-120 minutes (more time needed for wrap up)</b>
<b>Overview:</b> <p>Students have completed all the missions and several supplemental lessons. They are ready to prepare for the Create Performance Task. The program must meet a set of requirements to earn all the points. This practice will meet all but one of the requirements (no loop in the function).</p>		<b>Objectives:</b> <ul style="list-style-type: none"> <li>• I can create a meaningful list</li> <li>• I can use a list in code in a meaningful way</li> <li>• I can create a function with a parameter</li> <li>• I can use the parameter in an if statement</li> </ul>
<b>Standards:</b> <p><b>2-AP-12</b> Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.</p> <p><b>3A-AP-14</b> Use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables.</p>	<b>CSP Framework:</b> <p>Computational Thinking Practices:</p> <p>4.C Identify and correct errors in algorithms and programs, including error discovery through testing.</p> <p>6.A Collaborate in the development of solutions.</p>	<b>Create PT Requirements:</b> <ul style="list-style-type: none"> <li>• Create a list</li> <li>• Use the list in a meaningful way</li> <li>• Create a function with at least one parameter</li> <li>• The function must have sequence, selection and iteration</li> <li>• Values of the parameter must affect the section of code that is executed</li> <li>• Call the function with argument</li> </ul>
<b>Preparation:</b> <p><b>Make a copy</b> of the assignment or put it in the LMS.</p> <p><b>Prepare</b> any formative assessments you want to use in the wrap-up</p>	<b>Links:</b> <ul style="list-style-type: none"> <li>• Codespace: <a href="https://sim.firialabs.com/">https://sim.firialabs.com/</a></li> <li>• <a href="#">Assignment</a></li> <li>• <a href="#">Instructions slide deck</a></li> <li>• <a href="#">Program code - one topic</a></li> <li>• <a href="#">Program code - two topics</a></li> <li>• <a href="#">Program code w/ challenge</a></li> <li>• Daily reflection form</li> </ul>	<b>Agenda:</b> <ul style="list-style-type: none"> <li>• Warm-up (5 minutes)</li> <li>• Remix plan (20-40 minutes)</li> <li>• Remix coding (30-60 minutes)</li> <li>• Peer review (10-20 minutes)</li> <li>• Remix coding (20 minutes)</li> <li>• Wrap-up &amp; Assessment (10 minutes)</li> </ul>
<b>Vocabulary:</b> <ul style="list-style-type: none"> <li>• No new vocabulary during this lesson</li> <li>• You can review Create PT vocabulary: <b>parameter, argument, function, sequential, selection, iteration</b></li> </ul>		
<b>Assessment:</b> <ul style="list-style-type: none"> <li>• Daily reflection journal or form</li> <li>• Rubric (check-list)</li> <li>• Gallery Walk</li> </ul>		

## Teaching Guide


### Warm-up (5 minutes)

 **Discuss** – Use a discussion strategy, like journaling, working at boards, selecting random students, or a form of think-pair-share.

#### Options for Warm-Up

- Ask students about common programming errors
- Ask students what they know about, or review, functions and parameters
- Use the slides (slides 2-5) and the CollegeBoard website to introduce the create performance task and its requirements
  - [CollegeBoard website about exam](#)
  - Use the video from AP Classroom: [Create Performance Task: Overview](#)

### Practice #1 - first part with one list (20-40 minutes)

 Randomly group students into pairs for pair programming.

#### Teaching tip – Intro:

Go through slides 2 through 5, or use any other resources to introduce the Create Performance Task. This practice will include all requirements except iteration. It also uses a global variable. Both iteration and the need for global variables will be discussed in future lessons.

#### Teaching tip – Step 1:

Students should work in pairs. They will create a new project. They need to be in CodeSpace in a place that is not associated with the simulator. In the slides, the suggestion is to go to Mission 7 Objective 9. It really can be anywhere as long as it isn't an objective using the simulator.

#### Teaching tip – Step 2:

Students will plan their topic and the two lists of data for the topic. Students should plan their topic and lists on the assignment document first, and then add it to the program.

#### Teaching tip – Step 3:

Students will create functions (no parameters or variables) to introduce the program and also an ending message. If they haven't done this before, you might want to discuss why they should include an intro and ending. Print statements are used for simplicity, and since this is a practice. If students want to do a similar program for the create PT, they can use `display.show` or anything else for an intro and ending. They could also add lighting pixels or including audio. But I don't recommend doing it here so they can make that as a modification for the create PT.

#### Teaching tip – Step 4:

Students will write the code for the main program. This should be familiar code – combining the code from Mission 7 where they scrolled left and right, and the code from Mission 8 where they got a random item from a list. The difference will be that they need to use two lists with the same index instead of just one. If students need help, have them look at their previous programs. A hint for working with two lists is given on slide 12.

Don't let the students forget to test and debug as they write the code. Everything needs to be working correctly before moving forward.

## Practice #1 - second part with two lists (20-40 minutes)

### Teaching tip – Step 5:

Students will plan their second topic and the two lists of data for the topic. Students should plan their topic and lists on the assignment document first, and then add it to the program.

### Teaching tip – Step 6:

Students will create a new function using code from the infinite while loop. At this stage, the function will not have a parameter. That will be added in the next step.

The “index” variable will need to be global. This hasn’t been explicitly taught yet, so the students will just need to do it and it will be explained later.

### Teaching tip – Step 7:

Students will add print statements to the intro to print the two topics and give the user a choice (A and B)

Students will add another global variable in the main program for the “state” and use it in the main program with if statements to set the state.

Students will add a parameter to the function “state” and use it in an if statement to set local variables to the lists of the chosen topic.

Students then need to use the local list variables throughout the function. There are a lot of little things to change here.

### Teaching tip – Step 8:

Test and debug! Get everything to work as expected.

A **gallery walk** could be appropriate for students to celebrate their work and originality.

## Activity – Challenge (10 minutes)

### Teaching tip – Extension:

If students finish and have time for the challenge, it is fairly straightforward. They will add another local variable for the topic; assign the value in the if statement, and print it with the data from the lists.

Review the success criteria for completeness. Assignment is ready to turn in. Both students should include their names on the document.

To turn in the assignment, students should download their code (FILE-DOWNLOAD), which will be a text file. Then they should submit their file through Google Classroom or your LMS.

### **IMPORTANT!!**

Students should clear their CodeX by running their “Clear” program.

## Wrap-Up (10-20 minutes)

The wrap-up, if you want to use it, will have the students identify parts of the project that are requirements for the Create PT. It should be completed on the assignment document, but it can also be done as a whole class, or at the vertical white boards. Students may need help with this part, so be prepared on how you will assist the students in understanding what is being asked and how to select the correct code snippets.



Formative Assessment:

- Daily reflection journal or form
- Wrap-up questions
- Programming journal
- Exit ticket

Summative Assessment: Use the success criteria to evaluate the remix project

**SUCCESS CRITERIA:**

- Create a new program file for the Create PT Practice #1
- Import the modules needed for the program
- Create two lists on one topic
- Create functions for intro and ending
- Write code for the main program that will include if statements for four buttons
- Create two more lists for another topic
- Create a function for the code that is needed to print data from either topic
- Add a parameter with the "status" variable
- Use the parameter in an if statement
- Test and debug the program so that it runs as expected