**AP CSP CodeX**

|  |  |  |
| --- | --- | --- |
| **LESSON: PT Practice #3** | | **Time: 45 minutes** |
| **Project Goal:** Students will complete a program that meets the Create PT requirements.  **Learning Targets**   * I can create and use a list in a meaningful way. * I can create a function with a parameter, selection and iteration. * I can use the parameter in an if statement. | **Key Concepts**   * The Create PT has specific requirements for the program students create. * The parameter doesn’t have to be used in an if statement, but it should have an effect on the functionality. The easiest way to do this is to use the parameter in an if statement. | |
| **Assessment Opportunities**   * CodeBot PT Practice #3 Activity Guide * PT\_CodeBot\_Practice3 program * (optional) Writing prompts | **Success Criteria**   * Create a list * Use the list in a loop * Create a function with a parameter * Create a function that has a loop and if statement | |
| **AP CSP Framework**  **CRD-2.A** Describe the purpose of a computing innovation.  **CRD-2.C** Identify input(s) to a program.  **CRD-2.D** Identify output(s) produced by a program.  **AAP-1.D** Develop data abstraction using lists to store multiple elements.  **AAP-2.H** Write conditional statements.  **AAP-2.K** Write iteration statements.  **AAP-2.N** Write expressions that use list indexing and list procedures.  **AAP-3.A** Write statements to call procedures.  **AAP-3.C** Develop procedural abstractions to manage complexity in a program by writing procedures. | **Materials**   * CodeBot PT Practice #3 slides * CodeBot PT Practice #3 Activity Guide / Answers * Code solutions for intermediate and final program * AP CSP Student Handouts * Create PT Writing Prompts | |
| **Teacher Notes**   * This lesson will be completed on the computer, using CodeSpace for programming. * Use the Sandbox in CodeSpace for programming. This lesson is not part of a mission. * The activity guide can be distributed digitally. Space is provided for students to take notes during the programming. * Students will remix the WhatIf\_loops program to meet the requirements of the Create PT. * The best experience will come from them modifying their own code. However, we want all students to be engaged, so you can give them the original code to remix if needed. * The WhatIf\_loops program can be found in Unit 4 For Loops Practice. * Follow the slides for instructions and guidance. Additional help is provided in the Teaching Guide below. * Solution code for the final program is provided. | | |

**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. These are the same slides as the first two PT practice lessons, so it may take less time. Reviewing this many times should firmly cement the requirements.

* Slides 2-4
* Review the requirements for the Create Performance Task.
* Review how the Create PT is like a remix project.

**PT Practice 3 (30-40 minutes)**

💻 Students can work individually or with a collaborative partner.

**IMPORTANT!:** Students will use the WhatIf\_loops program from Unit 4 For Loops Practice. They need to have it completed and accessible. Alternatively, you can give students the code as a starter from the for loops lesson.

💡 **Teaching tip – Slides 5-7**

Students review Display\_parameters. Use the activity guide. Students should answer the first **three** questions before seeing slide 7. You can show it to help them with their answers, or wait until they answer the questions and let them check their answers with the slide.

💡 **Teaching tip – Slide 8**

Students brainstorm possible modifications.

💡 **Teaching tip – Slides 9-10**

Students create three lists with the information for each wheel and the delays.

💡 **Teaching tip – Slide 11**

Students review traversing multiple lists by looking at the code from PythonLists2

💡 **Teaching tip – Slides 12-13**

Students use the lists in the move() function, and modify the main program to call the function for testing. Most of the code in the main program can be deleted.

💡 **Teaching tip – Slides 14-17**

Students add a parameter to the function, and use the parameter in an if statement to assign lists for the wheels. The lists are used in the for loop.

💡 **Teaching tip – Slide 18**

Students modify the main program.

💡 **Teaching tip – Slide 20**

Students return to the activity guide and brainstorm their own remix ideas. Alternatively, you can use collaborative groups or whole class discussion with the question

💡 **Teaching tip – Slides 21-23**

Final slide that goes over information about the program they may be asked to write about in the Create PT writing prompts. The wrap-up will start to ask potential writing prompts so students can practice their responses.

✅ **IMPORTANT!!**

Students should clear their CodeBot code.

**Wrap-Up (5-10 minutes)**

The wrap-up reviews their programming process and gives potential writing prompts. The last three questions on the wrap-up are actual sample writing prompts from College Board.

If time permits, have students review each other’s responses. Or select a couple responses and anonymously display them with the class and discuss if the response answers the question.

Formative Assessment:

* Daily reflection or journal entry
* Wrap-up questions
* Completed program
* Exit ticket
* Optional – pull a writing prompt from the “Create PT WR Prompts” and have students write about their code.