(http://csmatters.org) 5 - 6a

0b101 - 0b110

Create Performance Task Partial Practice 2020-2021

Unit 5. Data Manipulation

Revision Date: Jun 21, 2019
Duration: 4 50-minute sessions

Lesson Summary

Summary

Students collaboratively define, design and implement a programming project: a miniature version of the Create Performance Task. They create and share presentations and share with groups (of about six students).

Students present. the project goals, the development process they used, functional and dta abstractions they considered and used.

Outcomes

- Students will complete a miniature version of the Create Performance Task.

Overview

Session 1:

1. Presentation (5 min) Students are introduced to the Create Task.
2. Activity (10 min) Students select a small programming project to model the Create Task.
3. Activity (10 min) Students identify and describe key algorithm to use in their program.
4. Activity (10 min) Students identify a functional abstraction that uses a parameter and a data abstraction to use in their program.
5. Wrap Up (5 min) Students complete the project development plan table. In it they describe their program goals, a key algorithm they plan to implement, and the abstractions they plan to use.

Sessions 2 and 3:

1. Journal (2 min) - What is your plan for today for the development of your project?
2. Activity (40 min) - Students implement their projects.
3. Journal (5 min) - Reflection. What abstractions did you use in your project?

Session 4:

1. Getting Started (2 min) - Explain goals for the presentations.
2. Activity (30 min) - Students prepare presentations of their projects.
3. Presentations (15 min) - Students present their project to table groups.
4. Wrap Up (3 min) - Students create exit slips with any questions or comments about the Create Task.

Learning Objectives

Key Concepts

Students working in pairs practice choosing a project and planning how to implement it in a fixed time frame.

Students have just two days to plan and implement a project that uses a user created function and makes use of a list or other data structure. Since this is a practice task, teachers may provide help with algorithms, functions, data abstraction or other programming concepts. Remind students that during the actual task teachers may not provide this kind of support so as much as possible students should rely on the collaborative partners for assistance when needed.

Students may receive most or all of the credit from an incomplete project if the project demonstrates the required components.

For this practice task, teachers may want to provide program stubs. Stubs could include a suggested function and data abstraction. Functional abstraction should include a parameter that is used to make a selection in the procedure.

Teacher Resources

Lesson Plan

Session 1: Planning Day

Briefly present an overview of the Create Performance Task. Explain that students will have 12 hours to complete the Create Task later in the course and they will four sessions for the practice version we are beginning today. Explain that the Full Create Performance Task Guidelines are provided as a resource and a guide for students however students are to complete the guidelines for the practice task. Requirementst from the full task that we are not doing are struck though in the lis below. For instance, in the practice Create Task we will not be producing a video or writing responses about the video.

Full Create Performance Task Guidelines

General Requirements

You will be provided with a minimum of 12 hours of class time to complete and submit the following:

1. your complete program code;
2. a video (created independently) that displays the running of your program and demonstrates functionality you developed; and
3. your individual written responses to all the prompts in the performance task.

Submission Requirements

1. **Program Code**

Your program must demonstrate:

- output (tactile, audible, visual, or textual) based on input from:
  - the user (including user actions that trigger events); or
  - a device; or
  - a file;
- use of at least one list (or other collection type)to represent a collection of data related to the program's purpose; and
- development of at least one procedure that uses one or more parameters to accomplish the program's intended purpose, and that implements an algorithm that includes sequencing, selection, and iteration.

Include comments or acknowledgments for any part of the submitted program code that has been written by someone other than you and/or your collaborative partner(s).

Create a PDF file that contains all of your program code (including comments).

2. ~~**Video**~~

~~Your video must demonstrate your program running, including:~~

- ~~input to your program; and~~
- ~~at least one aspect of the functionality of your program; and~~
- ~~output produced by your program.~~

~~Your video:~~

- ~~must be either .mp4, .wmv, .avi, or .mov format; and~~
- ~~must not exceed 1 minute in length; and~~
- ~~must not exceed 30MB in file size.~~

~~Collaboration is not allowed during the development of your video. Your video must not contain any distinguishing information about yourself. Your video must not be narrated, but text captions are encouraged.~~

3. **Written Responses**

Submit one PDF file that includes your responses to each prompt below. Clearly label your responses 3a–3d in order. Your response to all prompts combined must not exceed 750 words, exclusive of the program code. Collaboration is not allowed when answering the written responses.

3a. Provide a written response that:

- describes the overall purpose of the program; and
- ~~describes what functionality the video illustrates; and~~
- ~~describes the input and output shown in the video.~~

3b. Capture and paste two program code segments you developed during the administration of this task which contain a list (or other collection type) being used in your program. The first program code segment must show how data has been stored in the list1. The second program code segment must show the data in the same list being processed, such as creating new data from the existing data. Then, provide a written response that:

- identifies the name of the list being processed in this response; and
- identifies what the data contained in the list is representing in your program; and
- ~~explains how the selected list manages complexity in your program code by explaining how your program code would be written differently without using this list.~~

3c. Capture and paste a procedure from your program that you developed during the administration of this task which implements an algorithm used in your program. This procedure must:

- contain and use one or more parameters that have an effect on the functionality of the procedure; and
- implement an algorithm that includes sequencing, selection and iteration.

Then, provide a written response that:

- describes what the selected procedure does and how it contributes to the overall functionality of the program; and
- explains how the algorithm implemented in the selected procedure accomplishes its task.

3d. Provide a written response that:

- describe ~~two~~ calls to the selected procedure identified in written response 3c. ~~Each call must pass different arguments that cause a different segment of code in the algorithm to execute; and~~
- ~~describes what condition(s) is being tested by each call to the procedure; and~~
- identifies the result of each call.

Practice Create Performance Task Guidelines

For this practice task, students will complete simpler project and a two to three minute presentation about it, rather than a video and a report. Students work in partners to select and develop projects.

After completing the project, students will create a presentation about it. The presentation should address:.

1. Program Goals
2. Key algorithm - Describe a selection the algorithm makes and describe how the algorithm uses iteration.
3. Functional abstraction - Describe at least two different tasks the function may perform.
4. Data abstraction - Describe the values that will be stored and how they will be used by the program.

Projects are chosen by the students. If they wish, projects may be based on the following labs from *How to Think Like a Computer Scientist*. Before selecting a project students should consider how they might use a data structure (list) in their project.

How to Think Like a Computer Scientist Labs

- Astronomy Animation (http://interactivepython.org/runestone/static/thinkcspy/Labs/astronomylab.html)
- Turtle Racing Lab (http://interactivepython.org/runestone/static/thinkcspy/Labs/lab03_01.html)
- Drawing a Circle (http://interactivepython.org/runestone/static/thinkcspy/Labs/lab04_01.html)
- Lessons from a Triangle (http://interactivepython.org/runestone/static/thinkcspy/Labs/lab04_01a.html)
- Finally a Circle (http://interactivepython.org/runestone/static/thinkcspy/Labs/lab04_01a.html#finally-a-circle)
- Counting Letters (http://interactivepython.org/runestone/static/thinkcspy/Labs/lab12_01.html)
- Letter Count Histogram (http://interactivepython.org/runestone/static/thinkcspy/Labs/lab12_02.html)
- Approximating the Value of Pi (http://interactivepython.org/runestone/static/thinkcspy/Labs/montepi.html)
- Python Beyond the Browser (http://interactivepython.org/runestone/static/thinkcspy/Labs/pythonshell.html)
- Experimenting With the 3n+1 Sequence (http://interactivepython.org/runestone/static/thinkcspy/Labs/sequencelab.html)
- Plotting a sine Wave (http://interactivepython.org/runestone/static/thinkcspy/Labs/sinlab.html)

Students choose and plan a project with partners and complete the project development table.

Project Development Plan	Team Members:
Project Name	
Project Goals	

Key algorithm Describe a selection the algorithm makes and describe how the algorithm uses iteration.	
Functional abstraction Describe at least two different tasks the function may perform.	
Data abstraction Describe the values that will be stored and how they will be used by the program.	

Closing (3 min)

Students submit the Project Development Plan.

Students reflect on their project and making journal entry of goals for tomorrow.

Session 2 & 3: Implementation Days

Warm up (3 min)

Students complete a brief journal entry planning their team and individual work for today.

Work Time (45 min)

Students work to implement and test projects. Teachers may evaluate student performance based on student journal entries and their observations of student effort in implementing the project. In the Create Performance task, students should direct their questions with their partners and or consult their class notes and examples for help. Teachers may only help resolve technical issues (hardware or software) or provide clarification of the project requirements. In this practice task, students may receive limited help in developing the project.

Closing (2 min)

Students submit the key algorithm code developed so far.

Students make a journal entry about the progress they made and their goals for tomorrow.

Day 4: Presentation Day

Each student prepares a presentation. Partners may collaborate in the report presentation but each partner must prepare their own report.

Getting Started:

Explain the presentation requirements.

Format: A two to three minute presentation with partners taking turns in the presentation,

Required Contents:

Partner Names

Purpose

Data Abstraction

Function in the project

Code showing creation and use.

Functional Abstraction

Function in the project

Code showing definition and calls

Parameter it uses and alternative paths through the procedure

Key Algorithm

Purpose of the algorithm

Code and how the algorithm works (pseudocode can be used)

Use of selection and iteration

Presentation Preparation (30 min)

Presentations (15 min)

Students present their project to table groups. Time the presentations so that they do not exceed 3 minutes. Students discuss with table groups what they like about the project, what they learned and any questions or comments they have.

Closing (2 min)

Students submit their presentations and program code. Students create exit slips with any questions or comments they have about the Create Task.

Evidence of Learning

Formative Assessment

For the practice task, if student projects are too big or too small in scope, teachers should provide guidance.

For the practice task, project descriptions and pseudocode/algorithm description for each proposed project should be assessed. Assessment can be done by collaborative partners first.

If partners have concerns, they should be brought to the teacher.

Summative Assessment

The project should be assessed using the latest rubric provided by the College Board.

The latest rubric (updated as of November 2018) is in the lesson folder. The practice project does not ask students to create a video so row 1 should net be used. The assessment can be the basis of a summative grade using the suggested scale below.

Complete presentation submitted 25%

Code with abstractions submitted 25%

Rubric Points (5) 5 * number of rubric points up to 50%

(http://www.umbc.edu/)


(http://www.umd.edu/)


(http://www.nsf.gov/)

Authored by: CS Matters in Maryland
Website: csmatters.org (http://csmatters.org)
Email: csmattersinmaryland@gmail.com (mailto:csmattersinmaryland@gmail.com)