(http://csmatters.org) 2 - 17

0b10 - 0b10001

Unit Assessment

Unit 2. Developing Programming Tools

Revision Date: Jul 22, 2019
Duration: 3 50-minute sessions

Lesson Summary

Summary

To conclude the unit, students will complete a small project as well as a written assessment. The project requires students to parse text and search through lists or words to find a specific characteristic. The assessment covers integers, strings, booleans, loops, if statements, and lists.

Outcomes

- Students will synthesize concepts from the previous lessons to create their first project in PyCharm.
- Students will design a function and determine the relationship between algorithms and functions.

Overview

1. Getting Started (5 min)
2. Independent Activity (40 min)
3. Wrap Up (5 min)

Learning Objectives

CSP Objectives

- *EU CRD-1 - Incorporating multiple perspectives through collaboration improves computing innovations as they are developed.*
  - LO CRD-1.C - Demonstrate effective interpersonal skills during collaboration.
- *EU CRD-2 - Developers create and innovate using an iterative design process that is user-focused, that incorporates implementation/feedback cycles, and that leaves ample room for experimentation and risk-taking.*

    - LO CRD-2.C - Identify input(s) to a program.
    - LO CRD-2.E - Develop a program using a development process.
    - LO CRD-2.I - For errors in an algorithm or program: a. Identify the error. b. Correct the error.
    - LO CRD-2.J - Identify inputs and corresponding expected outputs or behaviors that can be used to check the correctness of an algorithm or program.

- *EU AAP-1 - To find specific solutions to generalizable problems, programmers represent and organize data in multiple ways.*
    - LO AAP-1.A - Represent a value with a variable.
    - LO AAP-1.C - Represent a list or string using a variable.

- *EU AAP-2 - The way statements are sequenced and combined in a program determines the computed result. Programs incorporate iteration and selection constructs to represent repetition and make decisions to handle varied input values.*
    - LO AAP-2.A - Express an algorithm that uses sequencing without using a programming language.
    - LO AAP-2.B - Represent a step-by-step algorithmic process using sequential code statements.
    - LO AAP-2.C - Evaluate expressions that use arithmetic operators.
    - LO AAP-2.D - Evaluate expressions that manipulate strings.
    - LO AAP-2.E - For relationships between two variables, expressions, or values: a. Write expressions using relational operators. b. Evaluate expressions that use relational operators.
    - LO AAP-2.F - For relationships between Boolean values: a. Write expressions using logical operators. b. Evaluate expressions that use logic operators.
    - LO AAP-2.G - Express an algorithm that uses selection without using a programming language.
    - LO AAP-2.H - For selection: a. Write conditional statements. b. Determine the result of conditional statements.
    - LO AAP-2.I - For nested selection: a. Write nested conditional statements. b. Determine the result of nested conditional statements.
    - LO AAP-2.J - Express an algorithm that uses iteration without using a programming language.
    - LO AAP-2.K - For iteration: a. Write iteration statements. b. Determine the result or side-effect of iteration statements.
    - LO AAP-2.L - Compare multiple algorithms to determine if they yield the same side effect or result.
    - LO AAP-2.M - For algorithms: a. Create algorithms. b. Combine and modify existing algorithms.
    - LO AAP-2.N - For list operations: a. Write expressions that use list indexing and list procedures. b. Evaluate expressions that use list indexing and list procedures.
    - LO AAP-2.O - For algorithms involving elements of a list: a. Write iteration statements to traverse a list. b. Determine the result of an algorithm that includes list traversals.

- *EU AAP-3 - Programmers break down problems into smaller and more manageable pieces. By creating procedures and leveraging parameters, programmers generalize processes that can be reused. Procedures allow programmers to draw upon existing code that has* *already been tested, allowing them to write programs more quickly and with more confidence.*

  - LO AAP-3.A - For procedure calls: a. Write statements to call procedures. b. Determine the result or effect of a procedure call.

Math Common Core Practice:

- MP1: Make sense of problems and persevere in solving them.
- MP5: Use appropriate tools strategically.
- MP6: Attend to precision.
- MP7: Look for and make use of structure.
- MP8: Look for and express regularity in repeated reasoning.

Common Core Math:

- S-ID.1-4: Summarize, represent, and interpret data on a single count or measurement variable

NGSS Practices:

- 3. Planning and carrying out investigations
- 5. Using mathematics and computational thinking
- 8. Obtaining, evaluation, and communicating information

Key Concepts

Students should synthesize concepts from the previous lessons to create their first project in PyCharm. This lesson also pushes a student to think about how to design a function, and the relationship between algorithms and functions.

Essential Questions

- How are algorithms implemented and executed on computers and computational devices?
- What kinds of problems are easy, what kinds are difficult, and what kinds are impossible to solve algorithmically?
- How are algorithms evaluated?
- How are programs developed to help people, organizations or society solve problems?
- How are programs used for creative expression, to satisfy personal curiosity or to create new knowledge?
- How do computer programs implement algorithms?
- How do people develop and test computer programs?
- Which mathematical and logical concepts are fundamental to computer programming?

What are the key elements we need to think about when designing a function?

Teacher Resources

Student computer usage for this lesson is: **required**

In the Lesson Resources folder:

- Word Play
- Word Play Rubric
- Assessment

Lesson Plan

Getting Started (5 min)

- Ask the students to work in small groups to create pseudocode for the following function:
- Create a function titled `'is_palindrome'` that inputs a word and determines whether that word is a palindrome. If it is, return `True` , otherwise return `False` .
- As a class, use the groups' pseudocode to create the function `'is_palindrome'` .
- Introduce the projects to students. They will have a total of 2 sessions to complete their project.

Independent Activity (40 min)

Students work individually on the Word Play and Assessment which are found in the lesson resource folder.

Wrap up (5 min)

Allow students to continue working to the end of class on their projects; have individual check-ins with students to make sure that they are on track and have a clear idea of what they need to complete the following day.

Options for Differentiated Instruction

Option to allow students to complete Word Play with partners to promote collaboration, then complete the written assessment individually.

Evidence of Learning

Formative Assessment

Teacher will monitor the progress of the students on each of the programs in Word Play.

Summative Assessment

Written assessment (see google drive)

Project Assessment (see google drive for project and rubric)

(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)