(http://csmatters.org) 2 - 4

0b10 - 0b100

Algorithms: Pseudocode

Unit 2. Developing Programming Tools

Revision Date: Sep 08, 2019
Duration: 1 50-minute session

Lesson Summary

Pre-lesson Preparation

This is the second session on algorithms

Summary

During the second session, the students will use pseudocode to describe an algorithm.

Outcome

- Students will write pseudocode using sequencing, selection, and iteration constructs.
- Students will write pseudocode for algebra / geometry formulas.
- Students will write pseudocode for determining if a year is a leap year.

Overview

1. Getting Started (5 min)
2. Guided Activities (40 min)
    1. PowerPoint
    2. Think-Pair-Share
    3. Solutions check
3. Wrap Up (5 min)

Learning Objectives

CSP Objectives

- *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.B - Explain how a program or code segment functions.

- LO CRD-2.I - For errors in an algorithm or program: a. Identify the error. b. Correct the error.

- *EU DAT-1 - The way a computer represents data internally is different from the way the data is interpreted and displayed for the user. Programs are used to translate data into a representation more easily understood by people.*
  - LO DAT-1.A - Explain how data can be represented using bits.

- *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.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.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.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.

Math Common Core Practice:

- MP1: Make sense of problems and persevere in solving them.
- MP2: Reason abstractly and quantitatively.
- MP7: Look for and make use of structure.

Common Core Math:

- A-SSE.1-2: Interpret the structure of expressions
- A-SSE.3-4: Write expressions in equivalent forms to solve problems
- F-IF.1-3: Understand the concept of a function and use function notation
- F-IF.4-6: Interpret functions that arise in applications in terms of the context

Common Core ELA:

- RST 12.3 - Precisely follow a complex multistep procedure
- WHST 12.4 - Produce clear and coherent writing in which the development, organization, and style are appropriate to task, purpose, and audience

NGSS Practices:

- 2. Developing and using models
- 5. Using mathematics and computational thinking
- 6. Constructing explanations (for science) and designing solutions (engineering)

Key Concepts

Students will write pseudocode using sequencing, selection, and iteration constructs.

Essential Questions

- How are vastly different kinds of data, physical phenomena, and mathematical concepts represented on a computer?
- How are algorithms implemented and executed on computers and computational devices?
- How do computer programs implement algorithms?

Teacher Resources

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

Files in the Lesson Resources folder:

AlgorithmsPseudocode2.pptx : PowerPoint Slides for mini-lectures

Student Handout and Key for Matching Pennies Game

Student Handout for Rock Paper Scissors

Psuedocode Summary and Examples of common Algorithms.docx

Lesson Plan

Getting Started (5 min)

- Homework Review from Session 1 (students were assigned to write pseudocode for selected Algebra/Geometry calculations)
- Journal: Describe the algorithm of another student. Is there enough detail to allow somebody to follow the steps?

Guided Activities (40 min)

Presentation

Walk through “Selection Statements”; “Iteration / Repetition” slides from the AlgorithmsPseudocode2 file in the Lesson Resources folder. Emphasize:

1. Whenever you need to store information, it must go into a variable. So think about what variables might be needed when you are creating your algorithm

2. Program steps are executed in the order they are listed from top to bottom.

3. Selection and Iteration statements require conditionals. Identify a conditional as something that returns a True or False answer. If selects the next statement to occur by answering the conditional question as being true or false. I have in the past pointed out the True and Then both start with T so TRUE always does the THEN, wherease Else and False both end with LSE, so when the answer if FALSE, you do the ELSE.

While continues to loop as long as the conditional answer is TRUE. When the conditional answer is false, the algorithm jumps to the statement after the End While.

4. Nearly all programming languages are equivalent in terms of being able to express any algorithm. Each has its own way to use variables, conditions and repetition which are needed for a solution to almost all algorithms.

Guided Activity

During powerpoint, guide students through the Game of Matching Pennies (a student working copy and a solution key is in the Master Teacher Resource folder for this lesson).

Think-Pair-Share

Students work in pairs to create and share their pseudocode. Use the Rock Paper Scissors hand out to have student pairs psuedocode Rock Paper Scissors.

Students may create their algorithms form scratch or remix exisiting ones.

If there is time, have groups switch algorithms and critique the algorithm of the other group.

Walk through pseudocode syntax summary handout called **Pseudocode Summary and Examples of common Algorithms.docx** in Lesson Resources folder.

Students work through challenges and check their results against sample solutions.

Wrap Up (5 min)

Review slide: "Why we have leap years."

Homework:

Assign students to create pseudocode for leap years.

Options for Differentiated Instruction

Pairing of students and crossing pairs to form groups of four should be used for the set of exercises that are part of this lesson.

Evidence of Learning

Formative Assessment

Think-Pair-Share

Summative Assessment

Students will write pseudocode for algebra / geometry formulas. These will be entered into their class notes.

Students will write pseudocode for determining if a year is a leap year. This will be entered into their journals.

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