



# Algorithms: Basics

## Unit 2. Developing Programming Tools

**Revision Date:** Feb 06, 2020

**Duration:** 1 50-minute session

### Lesson Summary

#### Summary

This is the first day of a two-session lesson sequence with topics covered by mini-lectures, explorations, and practice exercises.

#### Outcomes

- Students will be able to answer the question, "What is an algorithm?"
- Students will be able to answer the question, "What structures are needed in algorithms?"
- Students will explore algorithms described in English
- Students will understand Magic Square construction and parallel algorithms

#### Overview

1. Getting Started (5 min)
2. Activities (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.A - Explain how computing innovations are improved through collaboration.
  - LO CRD-1.C - Demonstrate effective interpersonal skills during collaboration.
- *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.L - Compare multiple algorithms to determine if they yield the same side effect or result.
- *EU AAP-4 - There exist problems that computers cannot solve, and even when a computer can solve a problem, it may not be able to do so in a reasonable amount of time.*
  - LO AAP-4.A - For determining the efficiency of an algorithm: a. Explain the difference between algorithms that run in reasonable time and those that do not. b. Identify situations where a heuristic solution may be more appropriate.
- *EU CSN-2 - Parallel and distributed computing leverage multiple computers to more quickly solve complex problems or process large data sets.*
  - LO CSN-2.A - For sequential, parallel, and distributed computing: a. Compare problem solutions. b. Determine the efficiency of solutions.
  - LO CSN-2.B - Describe benefits and challenges of parallel and distributed computing.

#### 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

## Common Core ELA:

- RST 12.3 - Precisely follow a complex multistep procedure
- RST 12.4 - Determine the meaning of symbols, key terms, and other domain-specific words and phrases
- 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 provide a definition of "algorithm".

Students will identify the characteristics of describing algorithms in English, pseudocode, or a programming language.

Students will demonstrate an understanding that there are different algorithms for different situations, such as parallel processing for multiple CPUs.

## 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?

## Teacher Resources

Student computer usage for this lesson is: **none**

PowerPoint Slides for mini-lectures (AlgorithmsPseudocode1.pptx in the Lesson Resources folder)

Excel sheet summary for creating a 4-by-4 Magic Square (MagicSquare4by4.xls in the Lesson Resources folder)

(optional) 16 papers, each with a number from 1-16 on it for students to use when acting out the magic square.

## Lesson Plan

### Getting Started (5 min)

- Journal Entry Review

–Small group, then large group, review of two previous homework assignments (students were assigned to use two sources to find definitions of "algorithm" and were assigned to write down the steps involved in a daily task – creating a peanut butter and jelly sandwich; getting to school; brushing teeth; completing homework; etc...) The definitions, sources, and algorithm steps were to have been entered into their journal.

- Have students consider solving problems alone or with others. Point out that some computers have more than one processor and an algorithm can be written to do 2 or more processes in parallel. Consider some complications that might arise in the simple algorithms that have been developed by students if there were multiple CPUs.

## Guided Activities

### Activity 1

Say: In programming, a "code statement" is a part of program code that expresses an action to be carried out. Note that it is a single action. Each step of an algorithm is to be carried out on its own and exactly as written.

- Act out selected homework "steps of a daily task" to highlight the potential ambiguity of English instructions. While acting out the algorithm have students look for the three sufficient parts of any algorithm - steps done in sequence, steps selected if a condition is met and steps done that are repeated.

- Mini-lecture 1

Algorithms that appear similar differ in that they yield different side effects or results. For many tasks however, more than one algorithm has been developed that accomplish the same tasks - though they may not be equivalent in some ways. For instance, some may be simpler, some may be faster, some may be more efficient but all may still be correct in the sense of achieving the desired result.

- Review Understanding by Design (UBD)-style slide for "Algorithms and Pseudocode: Need to Understand / Important to Know or Do / Worth Being Familiar With" (slide 2 of AlgorithmsPseudocode1.pptx presentation in Lesson Resources folder).

### Activity 2

Students join groups that have developed algorithms to accomplish the same task. Students are to exchange algorithms and determine:

- if the algorithm successfully accomplishes the same task as their algorithm - or comes close
- How is the algorithm the same and how is it different from their algorithm
- If the algorithm have any side effects that differ from th side effects of their algorithm

Time permitting, the class creates a 4-by-4 Magic Square, as described in MagicSquare4by4.xls in Lesson Resources folder. Sixteen students should be chosen to represent the sixteen numbers, and physically move into the 16 spaces following the algorithm in this document. (This provides an example of a "simple" algorithm that solves a more complex problem; it also gets the students out of their seats and moving around).

Reflect: Would it be easier or harder for 2 people to work on this together? (multiple CPUs or parallel processing)

- Mini-lecture 2

-In AlgorithmsPseudocode1.pptx (presentation in Lesson Resources folder), walk through "Main Ideas"; "Representing Algorithms"; and "Sequential Algorithms" slides, including pseudocode circle example.

### Wrap Up (5 min)

- Review concept of "algorithm"
- Review concept of multiprocessor algorithms that use parallel processing.
- Homework: Assign students to write pseudocode for selected algebraic computations.

## Options for Differentiated Instruction

"In your words" pair/share as the concept of what an algorithm is (and what is not an algorithm -- e.g. "sort the numbers") is developed.

Class-wide development of the graphical organizer should be facilitated with scaffolding to support students who are having difficulty with the concept.

## Evidence of Learning

## Formative Assessment

The following "Checks for Understanding" could be used to guide the students towards the two learning objectives.

Objective: Students will be expected to learn to provide a definition of "algorithm".

1. Example/Non-Example: As the class develops a consensus definition of "algorithm," a list of daily living tasks and technical tasks will be discussed to see if their description can match the definition.
2. Exit Ticket: As students leave the room they will hand the teacher a definition of "algorithm", in their own words, plus an example. The teacher will use these to guide the next day's instruction.

Objective: Students will learn to identify the characteristics of describing algorithms in English, pseudocode, or in a programming language.

1. Graphic Organizer: As the characteristics of English, pseudocode, and a programming language are compared and contrasted, a table will be created on the board that captures the dimensions of what the students discuss. The students will score the result as to how well it helps their understanding of the concept. When it is "very good," we will capture it for their use.

## Summative Assessment

Students will research formal definitions of algorithms. These will be entered into their journals and pair-shared with a peer.

Students will write out the sequence of steps in one or more daily living tasks, such as "brushing their teeth" or "building a peanut butter and jelly sandwich." These will be entered into their journals and pair-shared with a peer. Selected solutions will be "acted out" in the classroom.



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



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



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

*Authored by:* CS Matters in Maryland

*Website:* [csmatters.org](http://csmatters.org) (<http://csmatters.org>)

*Email:* [csmattersinmaryland@gmail.com](mailto:csmattersinmaryland@gmail.com) (<mailto:csmattersinmaryland@gmail.com>)

This work is licensed under a  
Creative Commons Attribution-ShareAlike 3.0 United States License (<http://creativecommons.org/licenses/by-sa/3.0/us/>)  
by University of Maryland, Baltimore County (<http://umbc.edu>) and University of Maryland, College Park (<http://umd.edu>).