# Lesson 1.02 — Algorithms & Computational Thinking

## Overview

**Objectives — *Students will be able to...***

- **Define** algorithms, programs, hardware, software, and operating systems.
- **Describe** the relationships between these concepts and components.

**Assessments — *Students will...***

- **Write** sample algorithms
- **Assemble and debug** a program that directs the instructor to make a sandwich

**Homework — *Students will...***

- **Read** BJP 1.2

## Materials & Prep

- **Projector and computer OR whiteboard and marker**
- **Food items** for peanut butter and jelly sandwich
- **Utensils** for sandwich assembly (spoon for jelly, knife for spreading)
- **Wet wipes or water** to clean hands

## Pacing Guide

| Section | Total Time |
|---|---:|
| Bell-work and attendance | 5min |
| Introduction to vocabulary | 10min |
| Explaining activity, assigning pairs | 2min |
| Activity Round One | 10min |
| Activity Round Two | 10–15min |
| Full-class discussion of debugging methods | 5min |
| HW distribution & exit tickets | 2min |

## Procedure

### Bell-work and Attendance [5 minutes]

### Introduction to Vocabulary [10 minutes]

In this lesson, you will be using yourself as the metaphor to introduce fundamental vocabulary in CS. As a hook, you should leave all of the food prep materials conspicuously laid out in the front of the classroom, without any explanation.

1. You should begin your lesson with an informal, whole-group conversation about humans being the first "computers." This is a great opportunity to use history, women's accomplishments in CS, and humor to offer up and drill vocabulary without onerous repetition. Some suggestions for your opening discussion:
   - Computer was originally a job description, and was first used in 1613 to describe someone who performs mathematical calculations.
   - Early computers were mostly women:

     http://www.officemuseum.com/IMagesWWW/Early_1920s_Veterans_Bureau_Calculating_WWI_Vet_Bonuses_I
   - Human computers have many similarities to mechanical computers. See how many of these your students can predict with minimal prompting:
     - Brain = CPU and processing

    - Input = information (data) from sensing the environment
    - Sensors = eyes, ears, nose, mouth, fingers/skin
    - Output = behavior, action
  - You can model errors and exceptions by dramatically narrating and acting out a confusing set of directions and the resultant mistake. Depending on your personal style, this can be a pratfall, embarrassing social blunder (real or imagined), or simple spelling mistake (input: sounding out a word with silent letters, error/output: the misspelled word)

2. Introduce the definition of an **algorithm**, and invite the students to write an algorithm for you, the computer, to make a peanut butter and jelly sandwich.
   - An algorithm may be defined as "a process or set of rules to be followed in calculations or other problem-solving operations"

### Explaining Activity and Assigning Pairs (2 minutes)

Give student pairs 5 minutes to write a peanut-butter-and-jelly algorithm. Specify that they should write a **complete set of directions** to describe the process of making the sandwich.

### Activity Round 1 (10 minutes)

1. Ask for a student volunteer to come to the front of the classroom to "be Java."
2. Randomly choose a student's algorithm for your first try, and narrate as "Java" picks up the "program" to read to you (the computer). Point out that when Java reads code, its compiler translates the code you type in so the computer can read it in binary. Bonus points if you can get the volunteer to read directions in a robotic voice.
3. Ask your student volunteer reads the directions aloud, following instructions literally. Repeat the instruction out loud as you model the action. Usually students forget to tell you to open the bag of bread, etc. Ham up the errors, and stop "executing the program" when it becomes clear that the algorithm won't result in a functional PB&J. Point out that this is an error or exception—your program does the same thing when it gets instructions that don't make sense.
4. Repeat this with 1 or 2 other algorithms, then graciously agree to let the class try again. Make sure to use the phrase "debugging" and have the students use it as well.

### Activity Round 2 (10–15 minutes)

1. Give students another 5 minutes to correct their algorithms. If students are on-task and really getting passionate about the job, give them a few extra minutes.
2. Have "Java" select another "program" from your students. As you execute the program again, ask the class what represents **hardware**, **software**, **input**, **output**, **processing**, and the **program**.

### Full-class Discussion of Debugging Methods (5 minutes)

1. Ask students to share the different ways they debugged their code/program/algorithm. (Changing program content, switching algorithm order, etc.)
2. Give students with successful algorithms the various PB&Js (failed and otherwise).

### Homework Distribution and Exit Tickets (2 minutes)

Distribute homework and have students complete exit tickets.

## Accommodation and Differentiation

Before delivering this lesson, you should check with the classroom teacher to make sure none of the students have a peanut allergy. If a student does have an allergy, find out how severe the allergy is (you can opt to make this student a jelly sandwich, but if the allergy is severe, you should switch the demo to another food item).

In certain classrooms, peanut butter and jelly might not be a familiar food item. In these cases, it is best to do some research first to figure out what snack will be familiar enough to all your students that they can recommend an algorithm for preparing it. Some items (such as Navajo fry bread) might require additional planning. For maximum engagement, try to select a snack that:

- Requires 3 or more ingredients
- Requires assembly or preparation (such as peeling, dicing, *etc.*)
- Uses ingredients that are jarred or wrapped
- Are generally considered palatable to most students

In ELL classrooms, you should pair students to ease the writing/composition burden of this activity. For advanced students, invite them to write algorithms for other activities, such as getting to school on time.

## Teacher Prior CS Knowledge

- Algorithms are one of the fundamental concepts in computer science. Algorithms are fundamental to solving problems in computer science.
- The lesson plan uses analogy of the student as the compiler. When writing a Java program the following is the sequence of files and translations that are carried out:
    1. Programmer uses an editor to create Java program and saves in .java file
    2. Java compiler (javac program) takes .java file and complies code to bytecode and saves in .class file.
    3. Java virtual machine (java program) runs .class file by interpreting the bytecode for a specific machine based on the operating system and hardware.
- The lesson plan assumes the teacher has prior knowledge of the components of a physical computer and can relate them to a person. If you are not familiar with the components of a computer system:
    - CPU central processing unit – brains of the computer
    - Disk drive/SSD – long term storage
    - RAM random access memory – short term storage
    - Keyboard, mouse, touch screen – input devices
    - Display, sound, vibration – output devices

## Teaching Tips

- Tips for Lecturing: http://csteachingtips.org/tips-for-lecturing
- Tips for Introducing Computer Science: http://csteachingtips.org/Tips-for-introducing-computing
- The lesson plan has the teacher be the "computer" that follows the instructions read aloud by the student. You may be inclined to have a student be the computer. However, students are unpredictable and you may or may not get the outcomes you desire. Even though this activity is meant to be playful, student silliness can get out of hand when building their P&J sandwich. If you do choose to have students be the computer, setup norms like no throwing food and they must clean up after themselves.

Other algorithms you can have students write include tying shoe laces, brushing teeth, opening a locker.

## Misconceptions

Sometimes students think algorithms and computer programs are synonymous. While they are related, they are not synonymous. Humans have been using algorithms to solve problems way before computer ever existed. The invention of the computer created a platform for algorithms created that could be carried out by a machine. So a computer program is a tool used by people to expresses algorithms that can be executed by a computer.

## Forum discussion

Lesson 1.02 Algorithms Computational Thinking (TEALS Discourse account required)