

# Hardware vs. Software: Understanding Basic Computing

TOPIC: BASIC KNOWLEDGE OF COMPUTING SYSTEMS (SOFTWARE)

GRADES: K-2

LESSON DURATION: 45 MINUTES

SOFT SKILLS: COMMUNICATION, COLLABORATION, PROBLEM SOLVING, GRIT

---

## Introduction:

- This lesson is designed to give students a basic understanding of the definition of software. It can be combined with the activities included in other lessons on hardware and software in the K-2 band.

---

## Learning Outcome:

- Students will define software as the instructions given to a computer to perform a task.
- Students will display knowledge of the difference between hardware and software through completion of a task involving an algorithm.

---

## Activities:

1. Now that students have basic knowledge of hardware, introduce them to the concept of software. Both are equally necessary for a computing device to function and you cannot have one without the other. If hardware are the physical devices that make a computer work, software can be described as the tasks and programs that a computer can do. An analogy that most kids would understand: hardware is the musical instrument (drum, guitar, etc.) whereas music notes to be played are the software.
2. Other common analogies: the brain=hardware whereas the mind=software.
3. To illustrate this idea, have students complete a task. Choose a task that would require students to develop an algorithm-a step by step process in which the steps must be done in a specific order for the task to be completed correctly. For example, students at this age are learning to tie their shoes. Either take them through a step by step process for tying their shoes or have them explain the steps involved in tying their shoes. The teacher might consider having a pair of sneakers and allowing the students to verbally explain each step that he/she must take to tie the shoes. Only complete the actions that students state. If they forget a step or mess up, complete the action anyway. Allow them to fail to show them that failure isn't bad. They can always start over and learn from their mistake.
4. After the activity, review with students the steps they took to get the shoes tied. What went wrong? When did they need to adjust? Could they have done something better? The teacher can take this opportunity to explain that computers operate based upon algorithms. They must be told what to do, how and when to do it, and what to do if "x" happens. Explain to students that computers and software are not perfect-nor are the people behind the programs. Overcoming

© 2018 Teach Cyber



This work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-nc-sa/4.0/).

failure and working together are two good skills to highlight. The process of tying the shoes is like the software of a computer.

5. The teacher should then have a discussion that this is what software engineers (writers) do when writing computer programs. Sometimes flaws develop and then they must figure out what went wrong. (If this is done with older students, the teacher could go on to explain that if the flaws involve security issues, then information security specialists would be involved. The teacher could ask students how these flaws are fixed—usually through patches/updates).
6. This activity could then lead to teaching students to code via “Scratch based” programming. There are a variety of resources available including <https://scratch.mit.edu/>
7. The link for programming lessons for early readers on code.org is <https://curriculum.code.org/csf-1718/coursea/>

© 2018 Teach Cyber



This work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-nc-sa/4.0/).