Teaching Computational Thinking Skills from the Student's Early Years Through College

Overview

Computational thinking (CT), a popular term that experienced a surge of popularity in the 2000s, refers to a broad range of psychological processes that help us as humans to find effective methods to solve a problem, even design a system, or understand human behavior, and then take advantage of computer power to automate a wide range of intellectual processes. Several people have given different definitions to CT skills. There is not one single, agreed-upon definition of computational thinking. However, the popular ones, according to the Webinar Series: *Computational Thinking for Teachers & Parents, 2017* provides the following:

- *Defining multiple layers of abstraction, understanding the relationships between the layers, and deciding which details need to be highlighted (and complementarily, which details can be ignored) in each layer when trying to understand, explain, and solve problems in different domains;*
- *Decomposing large complex tasks into manageable modular subtasks that supports parallel execution and multiple problem solvers;*
- *Iteratively developing solutions and systematically detecting and correcting errors;*
- *Analyzing the efficiency of various solutions;*
- *Reformulating seemingly difficult problems into solvable forms using reduction, transformation, recursion, and simulation.*

Why is Computation Thinking skills important for students?

There is a long list of reasons why CT skills should be taught to students at their early years and all through their University level and since CT can be easily incorporated into a variety of taught courses. However, the following provides information about some of the rationale:

CT promotes teaching students the ability to learn how to break a complex problem into smaller, more comprehensible steps that makes sense and make the task of solving a large problem less tedious and easier to solve. Essentially, CT can help make big challenges small and complex solutions simple.

CT promotes creative problem-solving ability. A problem may have several possible solutions. Creative-problem ability will train students on how to identify those possibilities and then deploy the best and smartest solution to solving the problem at hand.

CT promotes debugging skills. Debugging is essentially a process of figuring out the root causes of problem in a situation. It's a term usually used in computer programming to troubleshoot errors in a program and make the program completely bug free. This is a useful skillset to have that can be applied to solving real life problem situations in future.

Logical thinking refers to the ability of a student to think in a disciplined manner or base his/her thoughts on facts and evidence as is presented using his/her logical thinking skills. Logical thinking

skills mean incorporating logic into one's thinking process whenever analyzing a problem in order to come up with a rational solution. This is considered one of the most important skills that CT can help develop in students.

CT can promote students' ability to recognize patterns. Recognizing patterns refers to when students acquire the skills to find order to something and then analyze (follow) the pattern observed to the logical solution. Pattern matching teaches students to look for commonalities between things. Then, once students see what is the same in the problem, they also can look for differences that might lead them toward a solution. As humans, the tendency to search for patterns in things in order to make sense of them is something that we do quite often. Introducing this step and teaching them as early as possible in students' early life is considered sensible and as easy as the most natural thing to do. Thus, we should teach students to sense and continue developing pattern recognition skills from their early ages.

Conclusion:

Computational thinking is the one of the most important scientific thinking skills that students can acquire. Its research based and the development of it is capable of bringing great changes to the typical student's education. In computer science for instance, computational thinking makes it not just be confined to tool subject. Computational thinking has the potential to make student learners not to be restricted to professional knowledge only. But it can emphasize knowledge creation rather than knowledge application only. Computational thinking can provide the countless possible ways to solve problems.