Understanding Computational Thinking and Its Importance—An Overview

Paul S. Wang, Kent State University

September 15, 2020

Abstract

Computational thinking (CT) is the mental skill to apply fundamental concepts and reasoning, derived from digital computing technologies, in all areas, including day-to-day activities. To **computize** means to apply CT and put it to practice. CT is a must-have skill for people in the digital age. It ought to be as basic as literacy for society and in education systems. CT can make you more efficient and effective. It can help avoid disasters, even save lives. Example applications are given to show the impact of CT.

Keywords: Computational Thinking, Computize, Algorithm, Problem Solving, Abstraction, Preplanning, Automation, Precise Communication, Computing, Information Technology, Education, Literacy.

Why Computational Thinking?

Digital computers brought us the information revolution. People in the information age have such wonderful tools as laptops, smartphones, the Internet, and the Web. *Information Technology* (IT) brought us tremendous benefits as well as brand new challenges. For example, we can *ask Google* to answer any questions we may have on just about any subject and usually get answers instantly. The Internet spans the globe and brings all parts of the world within instant reach. Yet, it also allows hackers to steal our information or worse, to hold our computers with mission-critical data for ransom.

As modern citizens, we need to understand computing and adopt IT techniques and products effectively, and to wisely mitigate the risks brought by the information highway. That's where *computational thinking* (CT) can come into play.

Furthermore, CT encourages an additional dimension in our way of thinking, a way that can bring enormous advantages in almost any area, even in our daily lives.

What Is CT?

Computational thinking is the mental skill to apply fundamental concepts and reasoning, derived from digital computing technologies, in all areas, including day-to-day activities. CT is thinking inspired by an understanding of computation, its advantages, limitations, and potential problems. CT also encourages us to keep asking questions such as "*What if we automate this?*" "*What instructions and precautions would we need if we were asking young children to do this?*" "*How efficient is this?*" and "*What can go wrong with this?*"

CT can expand your mind, help you solve problems, increase efficiency, avoid mistakes, and anticipate pitfalls, as well as interact and communicate better with others, people or machines. CT can make you more successful and even save lives!

Who Promotes Computational Thinking?

Back in March 2006, Dr. Jeannette M. Wing published an article on computational thinking in *The Communications of ACM* [1] and boldly advocated it as a skill for everyone:

> "Computational thinking builds on the power and limits of computing processes, whether they are executed by a human or by a machine. Computational methods and models give us the courage to solve problems and design systems that no one of us would be capable of tackling alone. ... Computational thinking is a fundamental skill for everyone, not just for computer scientists. To reading, writing, and arithmetic, we should add computational thinking to every child's analytical ability. Just as the printing press facilitated the spread of the three Rs, what is appropriately incestuous about this vision is that computing and computers facilitate the spread of computational thinking."

Within the academic research community, there have been significant discussions on computational thinking, what it encompasses, and its role inside the education system.

In educational circles, there is an increasing realization of the potential importance of learning to think computationally. According to a report on computational thinking by the National Research Council of The National Academies (NRC) [2]:

> "... Computational thinking is a fundamental analytical skill that everyone, not just computer scientists, can use to help solve problems, design systems, and understand human behavior. ... Computational thinking is likely to benefit not only other scientists but also everyone else. ..."

A Powerful Way of Thinking

OK great, CT is important. But what exactly are the concepts and methodologies it provides? Here is a list of some main aspects of CT:

- ***Simplification through abstraction***—Abstraction is a technique to reduce complexity by ignoring unimportant details and focusing on what matters. For example, a driver views a car in terms of how to drive it and ignores how it works or is built. A user cares only about which mouse button to click and keys to press and generally overlooks how computers work internally.

- ***Power of automation***—Arranging tasks so they become routine and easy to perform by others. Working out a systematic procedure, an algorithm [3], for carrying out recurring tasks can significantly increase efficiency and productivity.

- ***Iteration and recursion***—Ingeniously reapplying the same successful techniques and repeatedly executing the same set of steps to solve problems.

- ***An eye and a mind for details***—Small things such as characters in uppercase versus lowercase or with an extra space can make all the difference. Any piece of data may be subject to interpretation depending on the context. You need eyes of an eagle, mind of a detective, and a careful and meticulous approach. Overlooking anything can and will lead to failure.

- ***Precision in communication***—Try telling the computer to do what you mean and not what you say `;-)`. You need to spell it out precisely and completely. Don't spare any details. Vagueness is not tolerated. And contexts must be made explicit.

- ***Logical deductions***—"Cold logic" rules. Causes will result in consequences, whether you like it or not. There is no room for wishful or emotional thinking. Don't we all wish some of this seeps into such things as our politics?

- ***Breaking out of the box***—A computer program executes code to achieve any task. Unlike humans, especially experts, it does not bring experience or expertise to bear. Coding a solution forces us to think at a dumb computer's level (as if talking to a one-year-old) and get down to basics. This way, we will naturally need to think outside any "boxes."

- ***Anticipating problems***—Automation relies on preset conditions. All possible exceptions must be met with prearranged contingencies. Ever said "I'll take care of that later"? Because there is a chance you might forget, according to CT, you should have a contingency plan ready in case you do forget. Otherwise, you have set a trap for yourself.

These are just some of the main ideas. CT offers you many additional concepts and ways to think that can be just as, if not more, important.

Do You Computize?

In the author's textbook *From Computing to Computational Thinking* [4], a new word is defined.

Definition: **computize**, verb. To apply computational thinking. To view, consider, analyze, design, plan, work, and solve problems from a computational perspective.

We need to acquire CT and apply it everywhere, from our daily activities to projects in our professions. Computizing is CT in action. When considering, analyzing, designing, formulating, or devising a solution/answer to some specific problem, computizing becomes an important additional dimension of deliberation.

Here is an example of computizing in everyday life: *Whenever I leave one place to go to another, I need to check "Have I left anything behind?" and "Have I got everything I need for my destination?".*

Where to Apply CT?

People say "hindsight is 20/20." But, since automation must deal with all possible applications in the future, we must ask "what if" questions and take into account all conceivable scenarios and eventualities beforehand.

Let's look at a specific example. Hurricane Sandy (2012) was one of the deadliest and most destructive hurricanes in US history.

With CT at multiple levels, dare we say that many of the disasters from Sandy might have been substantially reduced?

- The New York City subway entrances and air vents are at street level. What if streets are flooded? What if flood water enters the subway?

- What if we need to fight fires in a flooded area? Do we have fire boats in addition to fire trucks? Do we have firefighters trained for boats?

- Most portable emergency power generators run on gasoline. What happens if gas runs out and gas stations are flooded?

- What if the drinking water supply stops? Can we provide emergency water from fire hydrants? In that case, can we use a mobile contraption that connects to a hydrant, purifies the water, and provides multiple faucets?

- What if emergency power generators are flooded? Should we waterproof generators in designated at-risk buildings?

- What if cell towers lose power? How hard is it to deploy airborne (drone?) cell relays in an emergency?

- What if we simulate storm damage with computer modeling and find out ahead of time what to prepare for?

So let's computize at multiple levels and do our best to get 20/20 hindsight ahead of time.

Everyday Applications

For us, we want to avoid fussy, vague, confused, wishful, emotional, impulsive, optimistic, or pessimistic thinking. We want to practice CT anytime and anywhere we can. We want to set clear goals, have a sequence of steps to achieve them (an algorithm or flowchart), anticipate problems and prepare solution plans in advance.

Take driving a car for example. What is the goal? It is to get to a destination safely. It is not enjoying the sound system, watching the scenery, or engaging in conversation, although we have nothing against any of that as long as it does not get in the way of safe driving. Texting and driving is never safe.

Stopped at a traffic light, we wait for the light to turn green. But, we may need to run the red light if an 18-wheeler is about to crash into us from behind. That means we need to be checking our rear-view mirror while waiting for the green light. When the light turns green, do we blindly rush into the intersection? What if a car is running the tail end of the yellow light or the red light?

Thus, the goal is not to obey traffic signals, but to make sure it is safe. In the United States, a car crash kills a person every 12 minutes on average. If you are thinking straight, is a car a fun machine or a dangerous one? CT can keep you focused on the goals, make you pay attention to details, plan for contingencies, and shield you from distractions. CT can save the day, and perhaps even your life!

Now let's apply CT to the task of "getting ready to drive a car" and write down an algorithm-inspired pre-drive checklist.

1. Am I ready to leave? Forgot to bring anything?

2. Walk around the car, check windows, tires, lights, back seat, and any objects and activities near the car.

3. Get in the car, foot on brake, close and lock all doors.

4. Adjust seat and steering column positions as needed. Check positions of all rear-view mirrors, buckle up.

5. Check the instrumentation panel, pay attention to the fuel level.

6. If necessary, familiarize yourself with the controls for lights, turn signals, wipers, heat/AC, and emergency signal. Make sure they are working properly.

7. Release the hand break, start the engine, shift gear.

8. Make sure the gear is in D or R as intended, then start driving.

Airlines have developed rigorous preflight checklists for safety. Incidents, sometimes fatal, happen when pilots and crew, failing computational thinking, do not follow the exact procedure. The same goes for doctors and nurses in hospitals, especially in operating rooms.

CT Success Stories

A *loop* in a program is a construct that applies the same set of steps repeatedly until a certain goal is achieved. This technique is known as *iteration* in computing.

Iteration of a process has led to the invention of the *polymerase chain reaction* (PCR) [5], a technique in molecular biology to generate thousands to millions of copies of a particular DNA sequence. Developed by Dr. Kary Mullis in 1983, PCR is now indispensable in medical and biological research and applications, including DNA testing and genetic fingerprinting. The impact of automated PCR is huge and far-reaching. Mullis was awarded the 1993 Nobel Prize in Chemistry for his part in the invention of PCR.

In recounting his invention, Dr. Mullis wrote in his book *Dancing Naked in the Mind Field* [6]:

> "*I knew computer programming, and from that I understood the power of a reiterative mathematical procedure. That's where you apply some process to a starting number to obtain a new number, and then you apply the same process to the new number, and so on. If the process is multiplication by two, then the result of many cycles is an exponential increase in the value of the original number: 2 becomes 4 becomes 8 becomes 16 becomes 32 and so on.*
>
> *If I could arrange for a short synthetic piece of DNA to find a particular sequence and then start a process whereby that sequence would reproduce itself over and over, then I would be close to solving my problem.*"

At the time of the invention, the "polymerase" and other related DNA duplication techniques were already known. It was the "chain reaction" part that was missing. Well, we have Dr. Mullis and his computational thinking to thank for the invention. And what a significant invention! *The New York Times* described it as "highly original and significant, virtually dividing biology into the two epochs of before P.C.R. and after P.C.R."
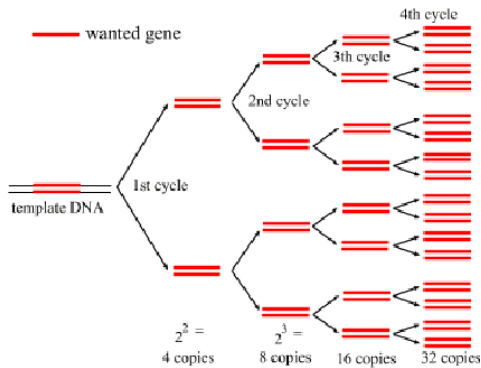
Another success story can be found in the fight against the COVID-19 pandemic which is still on going in many parts of the world as we write this article.

After discovering a new pneumonia-type disease at the end of December 2019, Chinese scientists released a draft genome of the newly discovered coronavirus [7] on January 11, 2020. With global scientific cooperation, coordination by WHO, worldwide studies of the virus (later formally named SARS-CoV-2) began rapidly, including source tracing and vaccine development.

One particularly effective way of controlling the spread of COVID-19 involves fast and wide testing, contact tracing, wearing face masks, social distancing, and issuing personal "health QR codes", notably in China, that can be used from smartphones. The health codes are particularly effective in allowing movement of people and safe resumption of economic activities.

The convergence of modern public health, microbiology, digital communication, and information technology [8], along with the application of CT gives humans their first chance in history to fight back effectively against such a pandemic and to save countless lives. A significant side effect of the pandemic is that people are going online for learning, teaching, shopping, conferencing etc., on a scale not seen before. This will change our post-pandemic lives forever.

With CT, society as a whole could have done much more online without

being pushed by a forceful event like the COVID-19 pandemic. But why not?

More Must Be Done

In this brief overview, it is impossible to cover the many aspects of Computational Thinking and how it can help individuals, organizations, and society as a whole. The author has a blog with regular posts on CT [9] covering many additional aspects of CT interesting and useful for everyone.

We must redouble our efforts to spread the word, to promote CT, to introduce CT into curricula at many levels, to hasten the day when CT is common sense.

For now, it suffices to say that, as history demonstrates time and again, a society, that is better educated in the next dominating technology and that can merge a new way of thinking into their endeavors, will have a significant competitive edge over others.

References

[1] Jeannette M. Wing. *Computational Thinking*, Communications Of The ACM, Vol. 49, No. 3, March 2006

[2] National Research Council. *Report of a Workshop on the Scope and Nature of Computational Thinking*, The National Academies Press, 2010

[3] Donald E. Knuth. *Art of Computer Programming, Volume 1: Fundamental Algorithms*, Addison-Wesley Professional; 3rd Edition, July 17, 1997

[4] Paul S. Wang. *From Computing to Computational Thinking*, CRC Press, 2015

[5] National Human Genome Research Institute (NIH). *Polymerase Chain Reaction (PCR) Fact Sheet*, www.genome.gov.

[6] Kary Mullis. *Dancing Naked in the Mind Field*, Vintage; Reprint Edition, January 4, 2000

[7] Jon Cohen. *Chinese researchers reveal draft genome of virus implicated in Wuhan pneumonia outbreak*, Science Magazine, January 11, 2020

[8] Sera Whitelaw, Mamas A Mamas, Eric Topol, Harriette G C Van Spall. *Applications of digital technology in COVID-19 pandemic planning and response*, The Lancet Viewpoint, Volume 2, ISSUE 8, June 29, 2020

[9] Paul S. Wang. *Computational Thinking Blog*, `computize.org/ctblog`