What is Computational Thinking and Why is it So Important?

As our global economy grows more focused on information and less focused on manufacturing, computational thinking is becoming an essential skill. Computational thinking is a powerful aid to mastery of virtually every academic subject, no matter what it is.

that Robots have been established as a useful tool to teach computational thinking to students, as young as 6 years old.

Computational thinking is a new development in problem solving that is changing the way students learn and how employees overcome challenges in the workplace. Jeannette Wing, a professor of computer science at Carnegie Mellon, introduced the term "computational thinking " in March 2006 in an article in a leading information technology journal.

> ***"Computational thinking involves solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science."***
>
> ***- Jeannette Wing -***

In her article, Wing described computational thinking as "computational methods and models that give us the courage to solve problems and design systems that no one of us would be capable of tackling alone." She also said that computational thinking "involves solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science."

In short, computational thinking is a process that makes the toughest challenges more manageable. And, while computational thinking draws on concepts fundamental to computer science, it can be used to solve a wide range of problems in a number of different subject areas and industries.

That's why computational thinking is such an important skill for today's students to learn.

What Exactly is Computational Thinking?

Wing's original article sets out in layman's terms some key points on what computational thinking means. Wing defines computational thinking as:

Not programming: Computational thinking requires thinking at "multiple levels of abstraction" which is a deeper skill than facility with a programming language.

Not memorization: Computational thinking is a problem-solving method that draws on human imagination and creativity; it is not that humans are trying to retain and manipulate more information, but find ways to grasp and work with complex ideas.

Complementary to mathematical thinking: Computing devices are limited by their foundational sciences. Computer scientists are forced to think computationally when they want to expand their ideas beyond the physical world.

A method for developing ideas, not artifacts: Computational thinking enables more effective interaction with the people and the world around us. Computational thinking is a about process: a process for working on problems, whether they involve computer science, social science, arts, humanities, or everyday life.

Today's Students Need Computational Thinking for Tomorrow's World

Copyright 2019 UCode All Rights Reserved.

The world and the global economy are changing around us. The manufacturing economy of the 20th century has given way to an age of information, automation, and algorithms. In this technology-saturated 21st century, the ability to work with information is prize above the skills to create, distribute and sell goods

> *"The 21st century and its advances in technology have created an information economy – one which requires us to solve problems far more abstract and complex than those of the manufacturing economy."*

- Doctors need to cope with drug resistant disease strains and new disease transmission vectors.
- Farmers, engineers and architects must respond to unprecedented population and climate pressures.
- Professionals in business, government, and academia will need tools to confront vast amounts of data – and the implications of so much data having been collected.
- Designers and artists will increasingly express their vision through virtual and networked media, changing the way we experience life.

Students who learn to think computationally will be the ones who participate in these new developments – and they will be the ones who enjoy steady employment during the ongoing and tumultuous transition from manufacturing economy to algorithmic economy.

How Does Computational Thinking Help Your Student?

To summarize, students who learn how to think computationally will have three critical strengths to succeed in the 21st century economy:

Problem-solving: computational thinking means breaking down overwhelming and complex challenges, into solvable problems.

Creativity: computational thinkers can research and organize, new information, and create new theories to understand and apply it to issues and projects of all kinds.

Autonomy and confidence: Computational thinkers are able to turn attention to the information and communicate about the problem and its possible solutions. They will feel comfortable working in groups as well as confident when forced to take on a challenge independently.

Regardless of your student's field of interest in school and career pursuit, computational thinking skills will support their success.

Google's Education arm has identified Computational Thinking as a powerful accelerator for student development, stating, *"CT is essential to the development of computer applications, but it can also be used to support problem solving across all disciplines, including math, science, and the humanities. Students who learn CT across the curriculum can begin to see a relationship between subjects as well as between school and life outside of the classroom."*

In some specific examples :

Literature: Computational thinking can help students break down and analyze poems with regard to structure, tone, meter, imagery and more.

Economics: Computational thinking can help students identify patterns and cycles that affect

Copyright 2019 UCode All Rights Reserved.

the rise and fall of a nation's economy.

Mathematics: Computational thinking can help students develop a reflexive understanding of difficult concepts, such as the rules for factoring second-order polynomials.

Chemistry: Computational thinking can help students visualize the rules that govern chemical bond and interactions.

Robotics Helps Students Develop Computational Thinking Skills

A strong connection exists between robotics and the development of computational thinking abilities. MIT Professor Seymour Papert first discovered this connection in the 1960s when he taught students to program a robotic turtle to make specific moves and take certain actions.[2]

Papert called the robotic turtles "objects(s) to think with." His students would watch the robotic turtle encounter an obstacle. The students would then imagine how they would navigate around the obstacle. And, finally, the students would apply that solution to the program, which then allowed the robotic turtle to take the steps needed to overcome the obstacle.

Papert's students created these solutions years before Jeannette Wing first wrote about computational thinking. But this process of encountering a problem, imagining the solution, and the creating a program to implement that solution is a perfect summary of how computational thinking works.

Students as young as 6 years old, can learn from exercises that connect robotics and computational thinking. Something as simple as describing the actions of a robot as it demonstrates its range of motion can serve as a beginning in computational thinking. Once this simple introduction is made, students can take on more and more complex exercises to expand their abilities for inquiry, observation, and creative problem solving.

Help Your Student Learn to Think Computationally

Computational thinking prepares today;'s students for success in countless fields. Computer science, of course, but also social sciences, arts, and all disciplines requiring science, math, and engineering.

While schools are beginning to recognize the impact of the algorithmic economy, the demands of the 20th-century curriculum leave little room to add in lessons that are specific to computational thinking. Teaching it well ultimately calls for an interdisciplinary and scaffolded curriculum; that effort could take many more years to develop let alone implement.

Where will your son or daughter develop the computational thinking skills needed to excel in school and thrive in his or her career to come? UCode offers a fully developed coding and robotics curriculum system, beginning with RoboFun for students 6-8, and continuing with a full scaffolded Python Plus+ curriculum for learners 9 to 14,

Both programs introduce the ideas and concepts of coding, programming, robotics and computational thinking. And the hands-on, experiential nature of both these programs makes the learning process fun – in fact, our students aren't even aware of the essential skills and abilities they develop while taking part. They only know that they're solving problems and achieving success alongside their peers and friends.

Help your student take a first step toward thinking computationally when you purchase a PythonPlus+ Trial Membership or a book of RoboFun lab passes.

[1] "What Is Computational Thinking?" Computational Thinking for Educators. [2] Kamada, Toshiyuki, Hiroyuki Aoki, Shuji Kurebayashi, and Yoshikazu Yamamoto. "Development of an Educational System to Control Robots for All Students." Lecture Notes in Computer Science Informatics Education – Supporting Computational Thinking(n.d.63

Copyright 2019 UCode All Rights Reserved.

Copyright 2019 UCode All Rights Reserved.