Course & Section: CMN 001 A18

Assignment: Persuasive Speech Outline

Name: Sriram Suresh

General Purpose: To persuade

Specific Purpose: To persuade audience to learn to code and think computationally

Target Audience: Mostly Neutral and Hostile

Attention:

Hi everyone. You may be surprised to know that you already think computationally, i.e., like a computer, from time to time. For example, when you are packing your bag to leave for a vacation, you are thinking computationally. However, you can maximize the benefits of computational thinking in your life by recognizing the instances in which you use it and gaining a deeper understanding of how to use it more often. The main way of maximizing this is by learning to code. Whenever I try to convince someone who doesn't code to start coding, I mostly get hit with the "But coding is boring and complicated, and I don't think I can work a 9-5 job staring at a screen while writing obscure looking code". However, computers can benefit you much more than just allowing you to play video games and write essays for your classes. Taking my own example, I have come to realize that the way I think about real-life situations has changed drastically ever since I started to learn to code and think computationally.

Need:

1. Often in one's day, one encounters complex situations which might require extensive and complicated thinking to get through. Due to a lack of familiarity with how to address this challenge, one might find themselves to be unable to make progress.
    a. If there was a way you could come up with a solution to your complicated problem by just following a set of steps, you might find it very helpful.
        i. For example, if you wanted to make a complicated food dish, but you had no clear idea of exactly how to make it, it might be a daunting task for you.

- Following the points that I just made, according to an article on *teachyourkidscode.com* about why computational thinking is important to teach to school kids, "Children must be taught how to think, not what to think."

2. As the stereotype goes, when most people think about coding, they probably relate it to a computer screen full of complicated looking computer instructions. However, coding is much simpler, and anyone can learn to code casually.
    a. To code professionally however, you need to be good with algorithms. An algorithm is much simpler than most of you probably think it is. In fact, you all already use algorithms in your daily life.
        i. While tying your shoelaces, for example, you are following an algorithm. You are essentially following a set of steps.
        ii. When you are given a deck of cards and are asked to separate the cards into two piles, a red pile, and a black pile, you start from the first card,

look at whether it is red or black and then put it into the red pile if it's red, and if it is not, you put it into the black pile. This is also an example of an algorithm.

Transition: Now that we have seen how some daily situations could be troublesome and complicated and that I have briefly mentioned how algorithms could help us, let us look at how computational thinking might help overcome these situations.

Satisfaction:

Solution:

1. Computational thinking is basically thinking about daily life problems as coding problems and coming up with an "algorithm" to help you solve the problem. Computational thinking also teaches you to recognize patterns in real-life situations and use the "algorithms" you know to help you make the best out of a situation.
    a. Going back to the example of cooking a complicated dish, if you just have a good recipe that tells you the exact steps you need to follow along with the ingredients and how much of each of the ingredients you need to use, you are now in a much better position to make the dish.
    - According to an article on *innovativeteachingideas.com* about why computational thinking is essential, to provide the readers with an analogy, the author says "When used in Cooking it is called a recipe. When used in Mathematics it is called an equation. When used in a basketball game we call it a play, and when used in computer science we call it coding."

2. Almost all problems can be approached with a computational thinking mindset.
    a. For example, if you are hiking and a bear decides that you look particularly tasty that day, you can do a couple of things based on the state of the bear. You could pretend to be dead, you can try to outrun it, you can try to fight it back, you can say your final prayers, and so on.

How the solution addresses the audience's needs:

3. Computational thinking helps you break down real-life problems into sub-problems and then tackle each of these sub-problems with an effective method to get optimal results.
4. Computational thinking allows you to create a list of instructions for you to follow and with this, you have a set of steps to work through a problem, rather than work aimlessly.
5. Coding has helped me communicate more effectively. Since you learn to speak to the computer using a programming language, learning to code teaches you to communicate using simple but effective terms.

- According to an article on *equip.learning.com* about how computational thinking can help students, "Computational thinking is about the process itself just as much as it is about solving the problem."
- So essentially, I am encouraging you to learn to code not for the sake of learning to code, but for the sake of developing computational thinking skills from the process of coding, skills which you could apply in real life situations.

Transition: Now that we have seen how computational thinking can help tackle complicated everyday situations, let us look at some concerns you may have about starting to code and thinking computationally.

Visualization:

1. Audience objections:
    a. Some of you may feel that it is inconvenient to pick up such a skill when you are busy with your other schoolwork especially when there is no obvious relevance to your field of study.

        Fix:

        You can start with very small steps, such as writing a computer program that takes 2 numbers as input and outputs the sum. If this doesn't excite you, you could try to motivate yourself by thinking of the things you could build by learning more about coding, such as creating video games, making your own website, etc.

    b. Now, if you were motivated to learn to code and think computationally, you may be confused and a little lost on how exactly to begin teaching yourself to do so.

        Fix:

You can look at simple tutorials on YouTube, a lot of websites offer free bootcamps and other helpful resources that will help you realize your goals, etc. If you are feeling extra ambitious, you could even do a minor in computer science.

- According to an article on *iste.org* about why computational thinking should be integrated into school curricula, the author says, “I find the most compelling reason for CT integration isn’t preparing students for the jobs of tomorrow, or even the emphasis on the computation, but rather the thinking.”

2. Starting to reason about your daily life situations with a computational thinking mindset, i.e., by recognizing patterns, breaking down problems into subproblems, etc., will help you come up with solutions to your problems easily. With more practice in using these principles, you will start solving real life problems much faster.

Action:

To start learning to code, you can start even by watching YouTube videos on the basics of coding. You should keep in mind that you are not learning to code to work a mundane 9-5 job, but instead, to start incorporating computational thinking into the way you process the decisions you make every day. In conclusion, I hope that after listening to the points I made on why you should learn to code and think computationally, you are now aware of how these principles could

help you tackle real-life situations with ease, much like a computer does, and that you will look into taking some of the steps I mentioned earlier.

Works Cited:

- Victoria, Katie. "What is Computational Thinking? Why thinking like a computer builds skills for success", *teachyourkidscode.com*, Dec 29th, 2018, https://teachyourkidscode.com/what-is-computational-thinking/
- Cummins, Kevin. "Five reasons why computational thinking is an essential tool for teachers and students.", *innovativeteachingideas.com*, December 6th, 2020, https://www.innovativeteachingideas.com/blog/five-reasons-why-computational-thinking-is-an-essential-tool-for-teachers-and-students
- McVeigh-Murphy, Anna. "What Is Computational Thinking? And Why Is It Important for Students?", *equip.learning.com*, May 22nd, 2019, https://equip.learning.com/computational-thinking
- Pinder, Nick. "Why You Should Integrate Computational Thinking Into Your Curriculum", *iste.org*, Jan 14th, 2022, https://www.iste.org/explore/computational-thinking/why-you-should-integrate-computational-thinking-your-curriculum