Austin McCalley

Dr. Jennifer Parham-Mocello

Computer Science 160-020

29 September 2019

**Questions for the teaching assistants in CS 160**

1. If you were to take CS 160 again what is something you would do differently?
2. What is something in CS 160 that you found difficult?
3. Have you gotten involved in research at Oregon State? If so how did you get started with research?
4. What made you want to become a teaching assistant?
5. How do you look for internships for computer science?
6. What is the best way to explore campus and find things to do?
7. What were the classes you took at Oregon State which you did not expect to be fun but in the end were fun?
8. How did you set yourself up for success at Oregon State?
9. If you had to choose a different major what would you pick?
10. Do you think that studying computer science was worth it?

**How do you define computer science?**

Computer science is the study of how computer systems, programs, and algorithms work. As the University of Maryland said “Principal areas of study within Computer Science include artificial intelligence, computer systems and networks, security, database systems, human computer interaction, vision and graphics, numerical analysis, programming languages, software engineering, bioinformatics and theory of computing. [1]” All of these areas of study fall under the big umbrella of computer science which were once studied by a computer scientist who analyzed the math, and or algorithm to solve a problem. Computer science has been able to optimize different solutions to problems to make life more efficient for end users, a way in which the complexity of a problem can be defined is by big O [2]. Computer science is an ever evolving idea which is extremely dynamic based off of what limits we as a society have pushed [3]. Even with computer science ever evolving I do believe that it still holds on to the root values of computer science with studying the way which algorithms work inside of software.

**What is computational thinking?**

Computational thinking is the way in which computer scientists write software for computers to analyze. [4] This way of computational thinking which computer scientists use to write software often is used in a problem solving way; for example, when thinking of how a certain problem works its possible to build out the problem using the “three As” [5] to think about how the problem works and then convert it to code for the computer to be able to understand. Computation thinking is the decomposition of a problem to its variables which can be used in the creation of an algorithm to develop a program to solve for the solution set of the problem. Using an analytical approach of computational thinking its possible to approach real world problems and interpret them using mathematics and computer science to solve the problem [6].

**Why are you studying computer science?**

I am studying computer science because the way which computers interpret software has fascinated me. I always wanted to learn how to optimize programs and software to make them faster and push the edge of technology. With studying computer science this will allow me to be the person that I always wanted to be by making the world a better place with software.

**Do you know how to convert back and forth between binary, decimal, octal and hexadecimal?**

I studied how to convert numbers to different bases such as binary, and octal my junior year of high school; however, I have forgotten how to do so since then. I believe with a little bit of refreshing I will remember how to do it.

[1] Anon. What is Computer Science? Retrieved September 28, 2019 from https://undergrad.cs.umd.edu/what-computer-science

[2] Anon. What is Computer Science? Retrieved September 28, 2019 from https://undergrad.cs.umd.edu/what-computer-science

[3] Melissa Wen and Melissa Wen. 2013. Computer Science: an ever-evolving discipline. (November 2013). Retrieved September 28, 2019 from https://www.dailycal.org/2013/11/14/computer-science-an-ever-evolving-discipline/

[4] Anon. Computational Thinking for Educators - - Unit 1 - Introducing Computational Thinking. Retrieved September 28, 2019 from https://computationalthinkingcourse.withgoogle.com/unit

[5] Steve Humble. 2018. Creating the coding generation in primary schools: a practical guide for cross-curricular teaching, London: Routledge.

[6] Anon.Retrieved September 29, 2019 from https://doi.org/10.1098/rsta.2008.0118