Field Report

Viktoriya Bardenova

Florida Gulf Coast University

Field Report

When a person hears the word “science,” usually, a computer is not what comes to his/her mind. Science studies the structure and behavior of the physical and natural world through observations and experiments. So, how do you observe and experiment computers? Well, that particular part of science, computer science, is concerned with “processing of information by means of computers.” (SE VOCAB, 2019a) The “resources required to acquire, process, store and disseminate information” (SE VOCAB, 2019b) are referred to as information technology. To process all that data (digital information), people need software to run the computer. The people who develop, operate and maintain the software are called software engineers.

There are different fields in computer science, such as theory, hardware, networking, graphics, programming languages, software engineering, systems, and AI. Computer science theory finds how fast a problem can be solved, or whether it can be solved at all. Some things take a long time. Others are impossible to solve. The field is highly mathematical, and development of algorithms is a part of it. Algorithms are step-by-step recipes to solve problems and a good algorithm can make solving time shorter. (Dale & Lewis, "Chapter 7: Problem solving and algorithms", 2016, pp.198-201)

Most non-theory areas of computer science depend on using programming languages. How to optimize the process of computing is an aspect of programming language study. In the beginning, computers used a machine language or machine code that was directly understood by the computer's central processing unit or CPU. This type of language only uses 1s and 0s (a binary system) to produce commands. Some disadvantages of machine language are that all operation codes and memory addresses have to be remembered and that it is not easy to understand by humans. (Dale & Lewis, "Chapter 6: Low-level programming languages and pseudocode", 2016, pp.154-155) Assembly language was developed to overcome some of the inconveniences of machine language. Instead of 1s and 0s, assembly language uses mnemonic codes such as ADDA, SUBA, STOP for its operations. While it is easier for humans to understand it more than machine language, assembly language is still very difficult, and people need a lot of practice to remember all the codes. It is also machine dependent, so the programmer needs to understand the hardware as well. (Dale & Lewis, "Chapter 6: Low-level programming languages and pseudocode", 2016, pp.166-168) With time, computer languages have been evolving and new, more productive high-level languages, that use English words and/or mathematical symbols rather than mnemonic codes, came along. Each instruction of a high-level language is translated into many machine-language instructions that the computer can understand. The language is independent of the machine on which it is used, meaning programs developed in a high-level language can be run on any computer. (Dale & Lewis, "Chapter 9: Functionality in high-level language", 2016, pp.300-311)

A third field of computer science is software engineering. It deals with the design, application, and maintenance (continuous development, not to wear and tear) of software. The software is a set of instructions that tells a computer how to perform a task. In the early days of computing, computer scientists produced software, and computer engineers built the hardware to host its execution. With time, the two intertwined to form software engineering. Although it is unlikely that every software engineer will have deep expertise in all aspects of computing, a general understanding is a necessity. Often, a software engineer can work at variety of fields such as computer science, programming, database administration and network administration. (Díaz-Herrera & Hilburn, 2015, page 14)

I am most interested in the fields of software engineering and programming languages since some of their tools can help with modeling and statistics. MATLAB and Python are two computer languages that I implement in my work and research in Math and science here at FGCU. As a meteorologist, I will need to use AWIPS (NOAA’s Advanced Weather Information Processing System). It is a computer processing system that “combines data from all the tools into a graphical interface that forecasters use to analyze data and prepare and issue forecasts, watches, warnings.” ("6 tools our meteorologists use to forecast the weather", 2017)

Introduction to Computer Science is a prerequisite class for a Bachelor’s in Science in Mathematics. It provides a lot of knowledge on how to utilize computers in different fields of Math. In the twenty-first century, computers are an imminent part of every-day life, and I think that this subject needs to be taught to anybody interested in any science field such as Biology, Chemistry, Math, etc. since it provides useful tools to solve problems in all of those disciplines.

References:

6 tools our meteorologists use to forecast the weather. (2017, August 14). Retrieved March 30, 2019, from https://www.noaa.gov/stories/6-tools-our-meteorologists-use-to-forecast-weather

Dale, N., & Lewis, J. (2016). Chapter 6: Chapter 6: Low-level programming languages and pseudocode. In *Computer science illuminated* (6th ed., pp. 153-195). Burlington, MA: Jones & Barlett Learning.

Dale, N., & Lewis, J. (2016). Chapter 7: Problem solving and algorithms. In *Computer science illuminated* (6th ed., pp. 197-245). Burlington, MA: Jones & Barlett Learning.

Dale, N., & Lewis, J. (2016). Chapter 9: Functionality in high-level language. In *Computer science illuminated* (6th ed., pp. 153-195). Burlington, MA: Jones & Barlett Learning.

Díaz-Herrera, J. L., & Hilburn, T. B. (Eds.). (2015, February 23). Software Engineering 2014. Retrieved March 30, 2019, from https://www.acm.org/binaries/content/assets/education/se2014.pdf

SEVOCAB. (2019a). In *Software and system engineering vocabulary.* Retrieved on March 30, 2019 from https://pascal.computer.org/sev_display/search.action;jsessionid=ec55a53303380e4bc377294e4bfa

SEVOCAB. (2019b). In *Software and system engineering vocabulary.* Retrieved on March 30, 2019 from https://pascal.computer.org/sev_display/search.action;jsessionid=ec55a53303380e4bc377294e4bfa