Hip, Hip, Array: Teaching Programming for Data Science is the same as Computer Science-- Just Different.

Jan W. Buzydlowski
Holy Family University, jbuzydlowski@hfu.edu

Abstract - Teaching an introductory programming class to a data science major needs to be done in a different manner than with a computer science major. This paper will focus on the major differences between the two majors regarding the need for different topical coverage.

Index terms – data science, introductory programming.

INTRODUCTION

Computer science is somewhat about computers, but data science is mostly about data. It could be argued that neither discipline is really about science, in a traditional experimental sense, but both are computationally bound, both alone and together.

Computer science is well-defined subject area but gets broader as time passes. This makes it difficult to teach the subject in a curriculum, as there is always the need to trim and add within the 120-credit circle. This can be aided by a consistent and agreed upon tautology of the subject [1].

One area within computer science that tends to drift and shift is how to teach programming to the freshman, e.g. [2], [3], etc. The language *du jour* and the preferred paradigm changes often, but it always focuses on providing a foundation for the languages and concepts that follow. In the spirit of Papert, the language becomes "a thing to think with." [4]

Data science is new, however, and trying to find its otologic feet. One of the original definitions by Higgins is to define it as non-mathematical statistics [5], but current trends have been attempts to define it in terms of a hybrid, combining computer science and statistics. It is also lumped in with information systems concepts such as systems analysis [6]. However, the field is definitely the center of a Venn diagram with computer science and statistics, at least, as the two major circles.

In terms of statistics, its traditional study is numerical, probabilistic, and formulaic, and this subject area, too, is well defined. It is important to note, however, that another major area of that discipline, perhaps unknown to the people outside of the area, is that of data visualization. A major work by Cleveland in this area begins with the quote that "Visualization is critical to data analysis. It provides a front line of attack, revealing intricate structure in data that cannot be absorbed in any other way." [7]

Given data science's components, then, the next step is to design a curriculum. Since computer science is a component, one could argue that the languages taught should be that of the computer scientists. However, the purpose of this paper is to suggest that this may not be true. Or, rather, that the topics covered, and the approach should be different.

EXPERIENCE

Teaching a general introductory computer programming course is sometimes made more difficult with the heterogeneity of the students, in terms of interests and majors. Whereas the interests can be managed by assignment of projects of interest, and varying application problems to the different disciplines, sometimes the disparity is so great that this cannot be accomplished due to the needs of the subject areas.

In general, a computer science major is oriented towards producing a product that will be used by others. Software and systems are created so that others may access information or create data. A data science major, however, is more oriented towards using software and systems to convert data to information for themselves, and then, ultimately, to their client. Because the ultimate end of the majors' product is significantly different, it is optimal to have two different courses for them.

Holy Family University now has a revised Introduction to Programming class offered in two different semesters for the two different majors: Computer Information Systems (INFO) and Business Intelligence (BUIN). The topical coverage is different, as well as the timing of the concepts. To further differential between the two, two different textbooks are used. Finally, to differential even further, a single, yet broad, language is used: Python (the contradiction will be explained).

In terms of the topics, INFO majors cover the basics of programming in terms of sequence, selection, iteration, arrays, and functions. The programming paradigm is that of structured programming, with coverage of objects near

978-1-7281-1502-3/19/$31.00 ©2019 IEEE

the end of the course as an introduction to the following course. Algorithms and their complexity are covered and emphasized. For example, several sorting routines are covered and studied in terms of their complexity and implementation. The implementation of the algorithms is important and stressed. The ability to write functions and libraries is also stressed. Ultimately, the goal of the course is to allow the students to create a system for the final project that is of interest to them by combining all the elements of the course, and the rubric used focuses on the creation of all those elements.

The BUIN majors focus on the topics of sequence, selection, iteration, arrays and functions, as well, but with less detail. Whereas the paradigm is that of structured programming as well, arrays are used from the beginning, and functional and object-oriented programming is covered near the end of the course as an introduction to a following course. Algorithms and their complexity are covered, but in less detail. For example, sorting is covered and a single algorithm is implemented, but the use of the Python `sort()` function is emphasized. The ability to use, rather than implement, functions and libraries is stressed instead. The goal of the course is a final project which extracts and analyzes a data set that is of interest to them, and the rubric used is the application of the libraries correctly.

In terms of the textbooks, the fall semester is geared towards the computer information majors and focuses more on computer science. The book used is "How to Think Like a Computer Scientist (using Python)." [8] The spring section of the data science majors use the book: "Python Programming: An Introduction to Computer Science." [9] Although the second book talks about CS, it focuses on lists and their use.

In terms of the language, the seeming contradiction of using a single language for the two different areas is resolved by the many different programming paradigms that the Python language supports. It is possible to write code in the language that is functional, imperative, and object-oriented—all as the same time. What makes it particularly useful for data science are the libraries that are supported by a vast open-source community. Finally, the data structures that are native to the language: dictionaries, sets, and, especially, lists are particularly useful and allows for the segregation of the two disciplines.

It is this last element, then, lists (or arrays), which this author thinks ultimately differentiates the two majors. Whereas computer science is concerned with single value variables to maintain state and arrays for data structures, data science thinks of data, i.e., more than one value—and often many—as a single entity, and this is best represented as a list. As a list, the native functions within Python, such as `sort()`, `find()`, `sum()`, etc., best serve the typical use cases of a data scientist. The addition of the functional programming components within Python for lists, and list comprehensions, such as `map()` and `filter()`, also serve the problems data science students seek to solve. Finally, although multi-dimensional lists within Python are difficult to implement, and the processing of same is not completely efficient, the NumPy libraries, an easy add-in to Python, solve both of those problems.

After the first course in programming, the second half of the courses are Advanced Programming for the INFO majors and Statistical Computing and Visualization for the BUIN majors. This is where the two majors part company.

The Advanced Programming follows that of the traditional CS2 course: objects and data structures. The Statistical Computing is a survey of statistical programming languages, but does feature text and screen scraping, using the libraries, NLTK, Beautiful Soup, LMXL, as well as sentiment analysis, using TextBlob, and Tweepy, data mining, using SciPy and NumPy, and data visualization, using MatPlotLib. Again, the INFO major's goal is to create software for others, whereas the BUIN major's focus is on correctly using others' libraries to write programs for themselves.

A third course in the sequence, Parallel Programming, important to both majors is in the works and will be the subject of another paper. This seems to be the point at where the two disciplines meet again.

The remaining courses for the data science major further reflect the needs of current employer's desires as indicated in the want ads: ETL, SQL, statistical analysis in R, etc. These requests indicate that employers are interested in employees who are facile in frameworks as classified, in the '90s lingo, of fourth-generational languages. This further emphasizes that data science majors need different computing skills than that of their computer science classmates.

Finally, the remaining coursework of the Business Intelligence major, outside of the data science curriculum, is that of business administration. It is generally thought that data science, particularly at the undergraduate level, needs a core competency to accompany the curriculum, especially business acumen, e.g., [10].

CONCLUSION

Data science is the merging of computer science and statistics; however, others suggest that it is non-mathematic statistics. This paper argues that it is all three. Furthermore, some believe that data science majors do not need to program, but the popular literature indicates that this may be a mistake. This paper suggests that if you teach programming to data science majors as you do to computer science majors that this, too, may be a mistake. In terms of Papert's *motif* of gears, the two majors need different things to think with as they need to think of different things.

REFERENCES

[1] Lillian N. Cassel, Siva Kumar Inguva, and Jan Buzydlowski, An Ontology of All of Computing: An Update on Challenges and Approaches, *Artificial Intelligence: Methodology, Systems, and Applications, 14th International Conference, AIMSA 2010, Varna, Bulgaria, September 8-10. 2010. Proceedings.* 2010.

[2] Stephen Cooper, Wanda Dann, and Randy Pausch. 2003. Teaching objects-first in introductory computer science. SIGCSE Bull. 35, 1 (January 2003), 191-195. DOI: https://doi.org/10.1145/792548.611966.

[3] S. Joosten, K. Van Den Berg, and G. Van Der Hoeven, "Teaching functional programming to first-year students," *Journal of Functional Programming*, vol. 3, no. 1, pp. 49–65, 1993.

[4] Seymour Papert. 1980. Mindstorms: Children, Computers, and Powerful Ideas. Basic Books, Inc., New York, NY, USA.

[5] Higgins, J. (1999). Nonmathematical Statistics: A New Direction for the Undergraduate Discipline. *The American Statistician,53*(1), 1-6. doi:10.2307/2685641.

[6] J. Buzydlowski and J. Pomykalski, Comparing and Contrasting Systems Analysis Methodologies with Data Analytic Frameworks, *Proceedings of the Northeastern Association of Business, Economics and Technology Conference (NABET)*, 2016.

[7] William S. Cleveland. 1993. Visualizing Data. Hobart Press.

[8] A. Downey, J. Elkner, and C. Meyers, 2002. How to Think Like a Computer Scientest: Learning with Python. Green Tea Press.

[9] John Zelle. 2010. Python Programming: An Introduction to Computer Science, 3rd Edition. Franklin, Beedle & Associates.

[10] 9 Must-have skills you need to become a Data Scientist, updated, 2018. https://www.kdnuggets.com/2018/05/simplilearn-9-must-have-skills-data-scientist.html. Web. Accessed March 3, 2019.

AUTHOR INFORMATION

Jan W. Buzydlowski, Assistant Professor, Department of Informatics, Holy Family University.