**Common and Preferred Programming Languages & Features**

**Among Senior CS Majors at Eastern**

Jillian Gonchar

**Introduction**

When creating and designing an application, there are many things that need to be taken into consideration such as what the program is expected to do, how interactive it is, and so on. Choosing the best language for a program is an even more challenging task, considering “program semantics” and the “code semantics” [1] differ greatly and cause a lot of complexity in a language. Typically, in schools, the main languages taught are Java and C++, and students need introductory level languages to help pave a pathway for later languages [2] but at the same time, more obscure ones should be included in the curriculum as well. One suggestion is to replace Java Standard Edition (J2SE) with Java Micro Edition (J2ME) in schools [3] because the issues identified by the JTF as being problematic with the Java language are less prominent in J2ME [3]. Through collecting data from Eastern’s senior computer science majors, what they prefer about languages in general, and which ones they use most, this data can be used to enhance the learning experience and help it to be more geared towards teaching students valuable skills in the field.

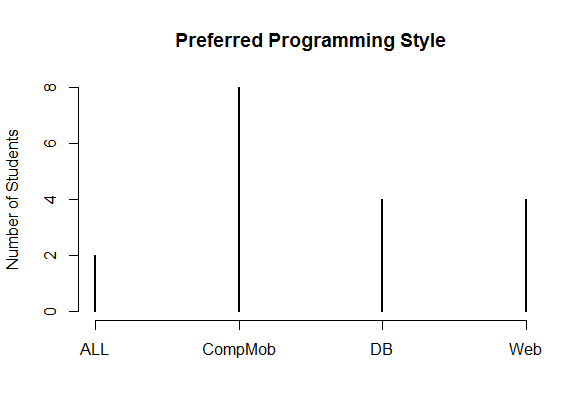
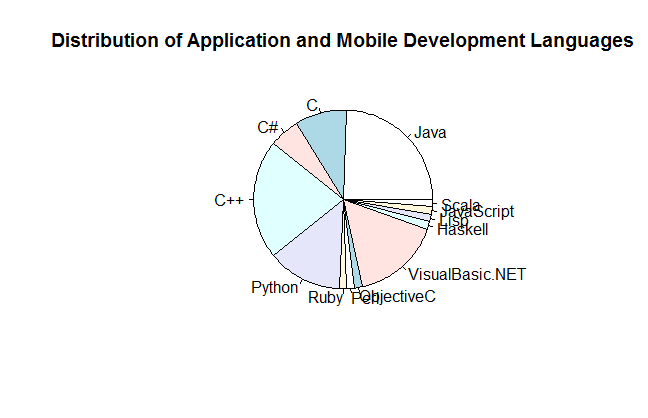
**Materials & Methods**

Before administering the surveys to students, I needed to complete the required online certification in Protecting Human Research Participants through the National Institute of Health (NIH). Once certified, I designed a survey of questions regarding programming languages and features, as well as gathering a general idea of how many languages each participant knew comprehensively, to better compare the results with the overall abilities of the class. The method of administering the survey was a voluntary paper survey handed out during class, and the collected information was inputted into an R script to analyze how the different traits and languages compare in popularity. The data was then compared to similar studies and information found online and how the results either related or differed to the preferences of computer science seniors at Eastern Connecticut State University.

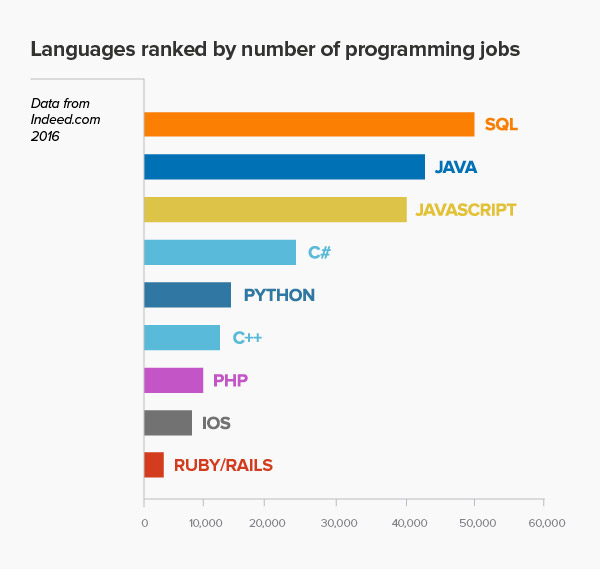
**Results**

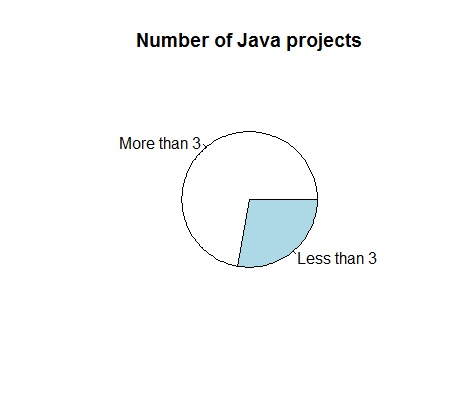
When observing the basic questions on the survey regarding programming style, overall languages known, and which languages offered by Eastern are most relevant, the results were very close to what I expected. My predictions were that most students overall would find computer and mobile application development to be the best description for their use of programming languages, and as the surveys showed, this was the most common, at 8 students listing this as their main use. The second most common was a tie between database and web development, with only 2 students choosing all three as equally representative of them (Figure 1).

(Figure 1)

In terms of application and mobile development, Java appears to be the most common among students, followed closely behind by C++, with VisualBasic.NET right behind, and Python. Less common, but still more than one student each, is C and C# (Figure 2). (Figure 2)

Compared to the outcome of popular features (being simple and concise code, readability, portability, object-oriented, solid APIs, and easy to learn), the most common languages jive pretty well with the results, as well with the most popular languages among computer science jobs (Figure 3).

Java, being the number one most used language by the class as a whole, is object-oriented, easy to learn, and has a solid API through Oracle. It is portable in most senses, however, it has less simple and concise code, and is open-sourced, which clashes a bit with the majority vote for which is more important. Java is taught in required classes at Eastern however, and in many of the electives. This is most likely a strong influencing factor as to why it’s the most commonly used because everyone has had to take CSC 231 (Programming II) which teaches Java, hence why all 18 students chose having done at least 1 project in Java (Figure 4). Java is widely considered one of the most popular languages on a global scale though, so even without the course at Eastern, it’s safe to assume that a lot of students would have pursued self-teaching or an online course in it since it’s so common.



(Figure 3)

(Figure 4)

After Java, C++ was the next most commonly used. Features of C++ that match with the class's overall preferred features are simple and concise code (only certain implementations of C++ are open-sourced), it's capable of being either procedural or object-oriented, so it fits both preferences on that matter, and is considered pretty readable. Even without much computer knowledge, some basic C++ code is pretty easy to comprehend at least from a quick glance. It's definitely considered one of the easiest to learn, hence why many schools, including Eastern, teach it as an introductory language. C++, however, isn't really portable, especially the executables, so it doesn't fit the class preference from that stand point. C++, like Java, is another language that is taught in one of the computer science major’s required courses at Eastern. Every student in the class should have already taken CSC 210 (Programming I) which teaches C++, and it’s because of this that it’s odd to find that two students wrote that they have never done a project in C++. Unlike Java, there aren’t as many elective courses that include C++ in the curriculum, but it’s still widely used in the real world and still relevant in today’s job market.

The third most common application development language is VisualBasic.NET. VB.NET is object-oriented, easy to learn, fairly readable, has solid online support, and has a version that is portable. VisualBasic.NET is a good beginner language for students as its syntax is fairly quick to pick up, even with no prior computer science knowledge, and depending on what you’re coding in, includes a lot of drag-drop capability in designing the user-interface portion of an application, unlike Java and C++ which require you to write the code for the user interface. VisualBasic.NET is taught at Eastern in CSC 249 (VisualBasic.NET), which is an elective for computer science students, but a requirement for MIS minors and a suggested elective for Business majors. Because it’s not required for computer science students, it’s not expected that many students will have taken the course, however many high school classes also use VisualBasic.NET as an introduction to computer programming, thus many students who have used it before may have even come into college with prior knowledge in it.

Python came in as the 4th most common language for application and mobile development. It is object-oriented and open-sourced, however its syntax is much less English-like and readable to those who aren't familiar with it, and not as easy to learn as Java or C++. It is, however, still considered fairly easy to learn overall. Python also has pretty solid online support. There are a few classes at Eastern that use Python as their main language, including CSC 360 (Intro to Bioinformatics), and occasionally CSC 320 (Computer Organization & Architecture) depending on who it’s taught by. Overall, not many students have taken CSC 320 using Python, and CSC 360 is an elective, therefore, of the 10 students who have used Python, it is possible that some may have taught themselves or learned it from an outside resource. Many jobs do consider Python a valuable language to know, so it would make sense to consider finding a way to incorporate some general knowledge of it into one of the introductory courses, at least for the sake of giving kids an idea of what it’s like to encourage them to pursue it on their own.

C was the 5th most common language among the class. C is not an object-oriented language, and only some implementations are open-sourced. The code itself is similar to C++ though, in which case it's pretty readable and easy to understand, also making it fairly simple to learn. Since C++ is merely an extension of C, there are a lot of similar features. The only thing separating the two in popularity among the class is age and relevance. Although some jobs may use C, C++ is more common in most places, and C isn’t really offered for any courses at Eastern, so most knowledge of this language would have to come from pursuing it on your own.

C#, the least popular language among the ones known by more than one person in the class, being a curly brace language and less English-like syntax makes it a bit more intimidating than some of the newer, and more popular languages for beginners and wide-range use. It is object-oriented and its code is generally pretty readable and easy to learn, however, it doesn't really have solid APIs to reference. It’s also a bit dated at this point and newer, more wide-range and capable languages have come along and sort of taken its place, so although it still has a purpose in application development, its relevance in the job market isn’t quite as high which would explain why Eastern doesn’t have any courses that stress learning it.

(Added to this will be analysis of the web languages and database languages, and comparing all of the results to how many languages students know overall, etc)

**Discussion**

Using this data, I think it can be considered as input into what languages might have a place in the computer science curriculum as perhaps new elective courses. Even considerations such as replacing Java Standard Edition with Java Micro Edition [3] since it has benefits to using the newer version such as many issues with the older Java version being less prominent in the newer version. In hindsight, the data collected can’t really be used as conclusive evidence that certain languages should or shouldn’t be added to the program since the sample size is so small. For future consideration, the survey could be extended to all computer science students as well as computer science minors, and staff of the department. Overall, this study can be used for a lot more than just what languages to consider adding to the program, because in job hunting you find that certain ones are more needed than others, so it could potentially be extended into the job market as well and used to help mold the way schools teach computer science languages and what features to emphasize on.

**Bibliography & References**

[1] Baniassad, Elisa, and Clayton Myers. "An Exploration of Program as a Language."

OOPSLA, 25 Oct. 2009. Web. 26 Oct. 2016. <http://delivery.acm.org/10.1145/1650000/1640132/p547-baniassad.pdf?ip=149.152.42.55&id=1640132&acc=ACTIVE%20SERVICE&key=94A5065CFB78B519%2EE540DBC41E6A94A8%2E4D4702B0C3E38B35%2E4D4702B0C3E38B35&CFID=686273639&CFTOKEN=77799430&\_\_acm\_\_=1477578437\_c4ebebf8a443a646786a8102b76d3764>.

[2] Lewis, Mark C., Douglas Blank, Kim Bruce, and Peter-Michael Osera. "Uncommon

Teaching Languages." SIGCSE, 5 Mar. 2016. Web. 27 Oct. 2016. <http://delivery.acm.org/10.1145/2850000/2844666/p492-lewis.pdf?ip=149.152.42.55&id=2844666&acc=ACTIVE%20SERVICE&key=94A5065CFB78B519%2EE540DBC41E6A94A8%2E4D4702B0C3E38B35%2E4D4702B0C3E38B35&CFID=686273639&CFTOKEN=77799430&\_\_acm\_\_=1477597347\_3c9c2b61452f60be1fd2a75549853428>.

[3] Utting, Ian. "Problems in the Initial Teaching of Programming Using Java: The Case for

Replacing J2SE with J2ME." ITiCSE, 28 June 2006. Web. 26 Oct. 2016. <http://delivery.acm.org/10.1145/1150000/1140176/p193-utting.pdf?ip=149.152.42.55&id=1140176&acc=ACTIVE%20SERVICE&key=94A5065CFB78B519%2EE540DBC41E6A94A8%2E4D4702B0C3E38B35%2E4D4702B0C3E38B35&CFID=686273639&CFTOKEN=77799430&\_\_acm\_\_=1477611329\_32837707ffe06d9d13cb4ae673c02b24>.

Figure 3:

Bouwkamp, Katie. *Languages Ranked by Number of Programming Jobs*. Digital

image. CodingDojo.com. CodingDojo, 17 Jan. 2016. Web. 14 Nov. 2016. <http://www.codingdojo.com/blog/9-most-in-demand-programming-languages-of-2016/>.