Northwestern should teach Python as the first programming language

At Northwestern, the typical sequence a Computer Science student takes is learning Dr. Racket, a scripting language, then C++, an object-oriented language. Students who did not learn Python in high school or learn it on their own are faced with the challenge of having to do so in higher level classes. Before making a case for why Python should be the first language taught to incoming freshmen, I need to describe the categorization of programming languages.

The simplest way to categorize languages, is as either low-level languages, which have little abstraction from the computer's instruction set architecture (ISA) like machine code and assembly code, or high-level languages that are easily read by humans for example C++, Java, Python, and Racket.

A better categorization is as either declarative or imperative. Imperative languages provide sequential instruction to achieve a desired state while declarative languages only provide the desired state. Imperative languages are split into procedural languages, for example the C language and object-oriented languages like C++, Python, and Java. Declarative languages are split into functional, logical, and mathematical languages.

The importance of this categorization was to show that C++ and Python are both high-level Object-Oriented programming languages (OOP) and as such can both be used to teach OOP concepts. What are the benefits of teaching Python before C++?

Python is a dynamically scripted language while C++ is a statically scripted language. This means that one can declare variables in Python without specifying their data types, but this is not possible in C++. For a programming beginner that is still grappling with Boolean algebra, teaching them C++ adds to the burden by having them figure out what data types take what amount of memory space. Taking data types as a simple example, we can see that C++ has a steeper learning curve than Python.

Python has simpler syntax than C++. C++ has been in commercial use for a longer period, implying that it has more libraries suited for experienced programmers. When writing the famous 'Hello World' program in C++, one has to start with the standard iostream library; students must therefore ignore the lines of code that initialize this library till they are able to understand what features it provides. Drawing a parallel between C++ and Java, Jenkins (2015) found "Students must learn (or be told to ignore the details of) a lot of Java before they can write even the simplest program. The API is massive, as is its documentation, and intimidating for the novice".

Python scripts are interpreted while C++ scripts are compiled. This means that a Python script will execute till an error is encountered while a C++ script will simply not compile if there is an error. For debugging purposes, Python is more helpful in diagnosis sources of errors and for a beginner, they will be many at first. As the joke goes, debugging your own code is like being the detective in a movie where you're also the murderer.

If Northwestern is to retain more beginners in Computer Science department or attract non-majors to take classes, using a simpler language will be an incentive.

References

Jenkins, T. (2015). The First Language- A Case for Python? Innovation in Teaching and Learning in Information and Computer Sciences. https://doi.org/10.11120/ital.2004.03020004

Mathieu Fourment, and Michael R Gillings. “A Comparison of Common Programming Languages Used in Bioinformatics.” BMC Bioinformatics, BioMed Central, 5 Feb. 2008, bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-9-82.