# Programming Languages Explanation

Paul Mauviel

paul.mauviel@snhu.edu

Southern New Hampshire University

CS-210: Programming Languages

Eric Gregori

1. There are many benefits to using C++ in a coding project. Among them is the general speed at which C++ executes. A C++ application has the benefit of being able to optimize for low-level assembly code which makes it an ideal candidate for low-latency applications. Additionally, C++ provides functionality for creating elegant source code organization and flexible pay-for-what-you-need data structures. For large, long-lived, development projects this facilitates maintenance and scalability while reducing the cognitive complexity of learning a new code base.   
    This flexibility does come at a cost. Programming in C++ is riddled with pitfalls that are much easier to fall into than in more managed languages. Specifically, it’s much easier to bungle memory management and leak resources leading to stability and performance issues. Additionally, C++ is generally considered ­slower for a software developer to develop for due to the wider range of concerns of which they must be cognizant.
2. Similarly, Python also comes with many benefits and several drawbacks when used in a coding project. To begin, Python’s syntax and built-in libraries are very easy to use and cover a wide range of use-cases out of the box. Manipulating files and data is much more straightforward in Python with its built-in open function and list-comprehension facilities (Timmins, n.d.) Additionally, having been widely adopted by mathematicians and data scientists, there is a wide array of Python packages available for performing sophisticated operations on data and performing visualization (Bierly, 2021.)  
    This ease of use comes at a cost when looking at Python from the lens of a developer of large-scale software projects. The dynamic-typing, lack of encapsulation, and other language-level features that come standard in other programming languages makes Python less easily maintained, more difficult to debug, and more difficult to jump into unfamiliar projects. Additionally, when thinking of a long-term engineering strategy for a company – becoming a Python shop may not be in the company’s best interests. Due to unique nature of Python, you run the risk of being too specialized and having a set of engineers that find it difficult to adapt to more complex or low-level programming languages (Disadvantages of Python - GeeksforGeeks, 2019.) That being said, it is likely that incorporating Python within your tools chain either at the repository or CI/CD level will prove beneficial.
3. There are many use-cases where multiple languages can be effectively combined in a project. At a high-level, this description can be applied to a standard full-stack application that will often use 2 languages at minimum: a back-end language such as C#, Java, or C++ and the front-end language, JavaScript. At a slightly deeper level, JavaScript bindings have been being used effectively within native desktop applications using frameworks such as Electron. These can allow for front-end web developers to create the user interface for native desktop applications using the same knowledge they’ve spent years mastering.   
    Moving beyond simple user interface integrations, game engines and similar content creation system often have a separate scripting language embedded into their application in order to provide user-friendly APIs and hide complexity. Lucas Meijer (2010), a developer at Unity Technologies, said: “Unity is written in C++, with the following exceptions: we expose a .NET api so that you don't have to go trough the pain of writing your game in c++, but you can write it in Javascript or c# or boo.”   
    As another example, at IMVU, where I am currently on the engine team, we build everything primarily in C++ but also provide bindings into the engine in JavaScript for the web client, Objective-C for the iOS applications, and Java for the Android Application.   
    Recently gaining traction, cryptocurrencies and blockchains are often implemented in using multiple programming languages as well. The Ethereum blockchain for example, has components of C++, Python, Ruby, Go, JavaScript, and Rust. The smart contracts that run within the Ethereum Virtual Machine are often written in Solidity and operate much like a scripting language in a game engine (Gupta, 2018.)  
    There are many, many more use-cases but integrating multiple languages is core for both enabling user-generated content and for enhancing (or enabling) platform compatibility.

**References**

Bierly, M. (2021, October 22). *10 Python Data Visualization Libraries for Any Field | Mode*. Mode.com. https://mode.com/blog/python-data-visualization-libraries/

Timmins, J. (n.d.). *When to Use a List Comprehension in Python – Real Python*. Realpython.com. Retrieved February 14, 2022, from https://realpython.com/list-comprehension-python/

*Disadvantages of Python - GeeksforGeeks*. (2019, January 14). GeeksforGeeks. https://www.geeksforgeeks.org/disadvantages-of-python/

Meijer, L. (2010, January 10). *Is Unity Engine written in Mono/C#? or C++ - Unity Answers*. Answers.unity.com. http://answers.unity.com/answers/9695/view.html

Gupta, K. (2018, June 15). *What Programming Language is Ethereum Written In?* Developers, Designers & Freelancers – FreelancingGig. https://www.freelancinggig.com/blog/2018/06/15/what-programming-language-is-ethereum-written-in/