**Assignment 2: Syntax, Semantics, and Memory Management**

**Introduction**

Programming languages are characterized by their own approaches for managing memory, grammar rules, and type systems. Understanding the nature of errors and how different languages handle them is essential for producing efficient, error-free programs. Syntax errors in Python, JavaScript, and C++ are examined in this article. We compare the type systems and scoping laws of Rust, Java, and C++, and we look at their memory management strategies (Anderson et al., 2024). You may learn how to build reliable code from the findings, which highlight the strengths and weaknesses of each language.

**Part 1: Analyzing Syntax and Semantics**

**Section 1: Syntax Errors Across Python, JavaScript, and C++**

This part describes the syntax errors that occur in the given Python, JavaScript, and C++ code and provides an explanation of the error messages that are produced by the compiler or interpreter (Anderson et al., 2024).

**Python Syntax Errors**

You can see the updated Python program with the syntax mistakes below:

Using 'o' instead of '0' causes a NameError to be sent out.

• A SyntaxError happens when the for loop does not include ':'.

**JavaScript Syntax Errors**

A screenshot of a computer

Description automatically generated

**Syntax Errors in C++**

**Key Errors:**

A computer screen with white text

Description automatically generated

**Section 2: Type Systems and Scope**

**Part 2: Memory Management**

**Rust Memory Management**

Rust uses ownership and borrowing to guarantee memory safety. The following code snippet shows an example of an error that could happen when trying to get a variable that has been moved.

**Java Memory Management**

The garbage collector is an automated mechanism that Java uses to manage memory. Objects are cleaned up when they are no longer referenced, as seen in the example below.

**C++ Memory Management**

Allocating and freeing memory in C++ is a manual process. Dangling pointers, which may happen when memory is not managed properly, are seen in the following example.

**Conclusion**

Based on our findings, different languages have distinct approaches to handling syntax errors, type systems, and memory management. The flexibility of dynamic type in languages like Python and JavaScript isn't without the risk of runtime errors, however. C++, on the other hand, uses static typing to prevent unexpected type mismatches (Abraham, 2020). Functional programming shines in JavaScript due to its lenient scoping limits, in contrast to the more rigid requirements of Python and C++. While Rust's ownership and borrowing mechanisms guarantee memory safety, Java's automatic memory management relies on garbage collection, and C++'s manual allocation and deallocation make memory leaks more probable. Knowing these differences can help you select the right language for your project based on its needs and performance requirements.

**References**

Anderson, C., Vandenberg, B., Hauser, C., Johansson, A., & Galloway, N. (2024). Semantic coherence dynamics in large language models through layered syntax-aware memory retention mechanism.

Abraham, W. (2020). Modality in syntax, semantics and pragmatics (Vol. 165). Cambridge University Press.