Requirements for Language Specification

Overview

For this first part of the project, you need to create specifications for the high-level language for which you want to implement a compiler. The specifications would be written informally using code examples.

Detailed Requirements

First, your team needs to

- Create a name for the experimental programming language (e.g., `XYZ-L`)
- Decide the extension given to the programs written in the defined language, (e.g., `hello.xyz`)
- Create a name for the compiler (e.g., `XYZ-LC`)

The experimental language should at least cover the following basic features:

- Integer scalar variables
- One-dimensional arrays of integers
- Assignment statements
- Arithmetic operators (e.g., “+”, “-”, “*”, “/”)
- Relational operators (e.g., “<”, “==”, “>”, “!=”)
- While or Do-While loops
- Break statement
- If-then-else statements
- Read and write statements
- Comments
- Functions (that can take multiple scalar arguments and return a single scalar result)

Besides the above, it is basically up to the team how the language is designed.
If you are not sure if something is required, ask the instructor/TA.

For each of the above features, the team needs to show a couple of examples. For example,

Language Feature	**Code Example**
Integer scalar variables	`int x; int y;` `int sum, avg;`

Note that, the language features should be easy-to-follow, consistent, and unambiguous.
Also note that you may change your design over the development process, just inform the TA when that happens.

Submission

Please use Markdown to write down the above language features on your team’s Github repo as the `README.md`

Due

By the End of the First Lab (Sep. 26th)