Project Name	Hopeful - A First Programming Language
Author	Gillian Mullen
Supervisor	David Sinclair
Document Title	Evaluation Analysis
Summary	This project, Hopeful - A First Programming Language, will involve the development of a programming language. This language will be aimed at beginning programmers in an undergraduate setting. It will have a simple and clean syntax that will allow students to focus on the fundamentals of programming, instead of a complicated syntax.

Overview

From the outset, my intent was to create a language that had a clear and simple syntax that allows the students to focus on the fundamentals of programming. My next goal was to evaluate my now-near complete language. To help myself achieve this end, I create an evaluation scenario involving Engineering and Computing students. I structured the evaluation so that I would gain a good idea of my progress of creating this language, and how successful I have been thus far.

In this process of the evaluation, the demographic was broken into three sections. *Beginners* comprising of first year students, *Intermediate* comprising of second and third year students, and finally *Advanced* comprising of fourth year students. All of these groups are made up of both Engineering and Computing students. In my analysis, I weighed the results equally. I made this choice as all groups have experienced learning how to program. As opposed to the survey, I believe personal bias affect all groups equally, and thus I would not consider it a factor in my results.

Environment

The base evaluation was the same for all groups. All participants were given a Plain Language Statement, Language Guide, a Virtual Machine, a survey and programming exercises. There were two subsections within the group of participants, those who attend the in-person evaluation section and those who attempted the evaluation by themselves.

The Virtual Machine supplied to participants contained everything necessary to complete the Hopeful exercises. This includes the *jar* file which ran the Hopeful code, LLVM installed to run the intermediate code, and sample programs for participants to run themselves.

The evaluation itself comprised of a series of programming exercises which were to be completed in both Hopeful and a language of the participant's choice (e.g. Python, Java, etc.). The order of the exercises were swapped as to avoid bias towards a certain language. Then, after the exercises were completed, the participants were asked to answer questions about their experience programming in Hopeful, how it compared to the other language, and if they had any suggestions for the language to help improve it.

Evaluation

The following are notes taken during the in-person evaluation session. Both sets of participants were made aware that Hopeful was a work in progress at the time of the evaluation. Many flaws pointed out in the evaluation were bugs which they were made aware of, or unfinished sections of the language, for example, bugs in returning in functions and missing void return type.

Along with the aforementioned issues, several issues came up in the in-person session. These include bugs with the *Boolean* type, negative numbers, and the inequality operator. During this session, suggestions were also made which I will discuss in the next section.

Results

The results of the evaluation were overall positive. When asked to rate their experience in Hopeful on a scale of 1 to 5, 88% of participants rated their experience a 3 or above. The amount of participants rated Hopeful the same after being asked to compare Hopeful to the other language the used in the evaluation.

Many of the critiques of Hopeful were unfinished of the language, which participants were made aware before the evaluation in the Hopeful Language Guide supplied to them. Critiques include the lack of a void type for functions, the lack of negative number functionality, being unable to declare a variable and give it a variable on the same, among others. These critiques were addressed and fixed for the final version of the project.

Aside from critiques relating to the project being unfinished, the rest of the feedback was mostly positive. For example, one participants remarked that the syntax was very clean and simple, and that the language could even be used to teach “students who are even at lower level than university”. Meanwhile, a beginner programmer remarked that the language was “a slightly simplified C”, which was the language that they chose for the evaluation. The simplicity in the syntax incorporated features and styles from other languages. The syntax was well received participants. Participants were asked to rate Hopeful on a scale of 1 to 5 in terms of how well a beginner would react to Hopeful’s syntax, all participants rated Hopeful a 3 or above.

Overall, the results were positive. I believe that the overall painted here by the evaluation is a positive picture of Hopeful as a programming language for beginners. Thus, I believe this project achieved its goal of being a simple language that allows beginners to focus on learning how to program, instead of complicated syntax.

Future Work

As for future work, since this project has a limited scope and time frame, these directions would be for someone who intended to continue with the project. The main goal would be to implement more features in the language, such as arrays. However, implementing features that would help the student to learn would also be part of the goal. For example, making indentation required in the language would teach the student to create readable code. Most of this work would be taking suggestions from the evaluation that could not be implemented by myself, such as for loops and converting between types.