Julibert Díaz

Angel Hernández

Anel Martínez

Naomy Morales  
CIIC4030-ICOM4036: Programming Languages

**Phase 1: Project Proposal**

**• Introduction:**

*Sphinx* will be a high level programming language similar to Python. It will be open source, cross-platform compatible, and easy to learn and use for developers. The programming language we strive for is one that is functional and that has an extended library for mathematical implementations. Our language will facilitate the implementation of data structures such as Linked Lists, ArrayLists and Maps. It will have error detection to help the user identify syntax errors and facilitate the debugging process. The purpose of our language is to create a new tool with improved specifications from other languages to optimize performance and software development. It will help users from different ages create applications for learning purposes, the work environment, or for just plain fun. We want to develop a language that is multifunctional.

**• Language Features (Python)**

* *Easy to learn and use:* It’ll be developer-friendly and a high level programming language. It aims to be easy to read and write to ensure a low level of complexity.
* *Familiar:* The language aims to be an implementation of an already existing language, therefore being in a familiar language that other users will easily get used to.
* *Free and Open Source:* Anyone can download it or access it with a computing device. It will be free of charge for the purpose of keeping it a good alternative tool for learning and for the needs of computer science students or developers around the world.
* *Easy to implement:* Due to its user friendly style and program structure, the language promises to optimize code and run efficiency with a limited amount of lines. This also includes the use of library methods that weren’t usable before like polynomial math methods, quadratic solutions, etc..
* *Cross-Platform compatibility:* Can run equally on different platforms such as Windows, Linux, Unix and Macintosh etc.
* *Functional Language:* It will have flexibility. The language aims to apply features of functionality like interfaces and lambda expressions that are generated by the translated language.
* *Error Detection:* The Language will point out errors by changing the color of the line which causes the compilation error

**• Example of a program**

##This is how comments will be made##

Because the main purpose of this language is to add an easier implementation of the math equations by expanding the math library using an already existing language, many mathematical expressions are to be added. Some examples are presented below with a description of what it should do:

math\_library\_example( ){

int x, y, z;

string poly\_result;

int quadratic\_result;

int polar\_result;

int magnitude\_result;

int cartesianX\_reslult;

int cartesianY\_result;

int summation\_result;

int factorial\_result;

poly\_result = Math.Polynomial(x,y,z);

//This would accept 3 parameters to make an equation turn polynomial

quadratic\_result = Math.Quadratic(x,y,z);

//This would search for the quadratic result of a polynomial equation

polar\_result = Math.getPolar(x,y);

//Calculates Polar result by accepting two parameters

magnitude\_result = Math.getMagnitude(x,y);

//Calculates Magnitude by accepting two parameters

cartesianX\_result = Math.getCartesianX(x,y);

//Calculates X cartesian if numbers were in polar type, accepts two parameters

cartesianY\_result = Math.getCartesianY(x,y);

//Calculates Y cartesian if numbers were in polar type, accepts two parameters

summations\_result = Math.Summation(x,y,z);

//Calculates summations

factorial\_result = Math.Factorial(x,y,z);

//Calculates Factorial

}

**• Implementation requirements and tools**

* We will use PyCharm as the IDE of choice for the project.
* For generating the scanner and parser we will use PLY.
* As reference we will use the Math library in python to develop the language.
* In PyCharm we will do the necessary intermediate code for testing that language is working as intended with all of its features using.
* Custom made Unit Tests will be designed and implemented for the language.
* To save progress we will be using GitHub repository.
* As for what tool we would use for the video we still haven't decided on one.
* Google Docs and Microsoft Word will be used as to keep track of documentation, language specs and other such files.
* Python Standard Library and the Python Language Reference for Python 3.8 will be used as documentation and implementation sources.

**• Project timeline**

|  |  |  |
| --- | --- | --- |
| **Assignment or Goals** | **Time Slot** | **Objective** |
| **Proposal** | March 2 | Deliver a possible project assignment |
| Study of the chosen reference language | March 10 | Research aspects on a existing programming language as a reference for our project |
| Regular expressions and lexical rules for lexical analyzer | March 20 | Define the lexical rules and regular expressions that will determine the syntax for our project |
| Implementation of parser | April 1 | Working parsing analyzer |
| Syntax analyzer and lexical analyzer | April 7 | Both lexical analyzer and parsing are implemented and functional |
| **Intermediate code** | Abril 8 | To be able to generate it correctly with our previous steps. |
| Finished Implementation: | April 30 | A functional programming language with all features of the expanded math library. |
| Documentation | May 1 | Deliver a commented document explaining in detail how the programming language works |
| Video Creation | May 3 | Class requisite, small descriptive video |
| **Delivery** | May 5 | Final delivery as final class requirement |