1. Scripture

D&C58:27-28

27 Verily I say, men should be anxiously engaged in a good cause, and do many things of their own free will, and bring to pass much righteousness;

28 For the power is in them, wherein they are agents unto themselves. And inasmuch as men do good they shall in nowise lose their reward.

1. Abstract

The project will consist of a new procedural programming language and interpreter required to execute the language.

* 1. "The focus of this project is input/output processing"
  2. "The application of this focus is an interpreter"

1. Background

Provide information essential to understanding your project. This includes, but is not limited to, the following:

* 1. Definitions

Interpreter – program that takes user input and generates meaningful output

Lexer – part of the interpreter that splits input into tokens

Parser – part of the interpreter that recognizes and arranges the grammar of tokens

Evaluator – part of the interpreter that executes parsed tokens as instructions

Lindy – simple procedural programming language defined specifically for this project

* 1. Why this topic is of interest (to you!)

This hits the very fundamentals of software engineering. Building software to solve problems is what the profession is about. Building the very tools in order to build the software is a very rewarding exercise. Just like studying data-structures can lead to new ideas and allow better understanding of how and when to use specific tools, studying compliers/interpretters gives an education on limitations of current technology and allow better understanding of when to use certain languages or features. This also opens the door to creating new languages specifically designed to solve problems in emerging markets or technologies, such as machine learning and biometric security.

1. Description

Project details

* 1. Program that can take an input file written in Lindy and execute appropriate instructions
  2. Executes as expected according to simple defined Lindy syntax handling general purpose programming concepts including scoped variables, literals, function calls, loops, and conditional statements.

1. Scope

This project includes the interpreter, a few sample input programs to demonstrate general purpose functionality such as basic sorting, and a document detailing basic Lindy syntax for the user.

This project does not cover advanced topics such as object orientation, functional programming, advanced data structures such as maps or stacks, or file I/O in Lindy. This project also is not an exercise in optimization or efficiency. It is only concerned with input/output processing or generating correct output for the given input.

1. Tasks and Schedule

Language development – defining features and syntax of sample Lindy language – 15 hours

Building a lexer – splitting input into independent tokens – 10 hours

Building a parser – arranging execution order of tokens – 15 hours

Input validation – enforcing correct syntax in parser with execution errors for user – 5 hours

Conditional branching – implementing if statement syntax – 15 hours

Loops – implementing loop syntax – 15 hours

Functions – implementing function calls – 15 hours

Scope – enforcing variable scope – 15 hours

Sample scripts – write a “hello world” program in Lindy and a basic sorting program in Lindy – 15 hours

Debugging language quirks – 20 hours

Documentation – formalizing basic user guide to Lindy syntax – 5 hours

Total: 145 hours

1. Applicability

This will require that I use data structures for implementation along with good programming practices to achieve an advanced product. With no previous courses focusing on input/output processing or on details or compilers or interpretters, this project will require in depth research into current and past practices in general purpose language development and execution.

1. References

<https://en.wikipedia.org/wiki/Syntax_(programming_languages)>

<https://en.wikipedia.org/wiki/Parsing#Computer_languages>

<https://en.wikipedia.org/wiki/Lexical_analysis>

<https://en.wikipedia.org/wiki/Lexical_analysis#Evaluator>

<https://ruslanspivak.com/lsbasi-part1/>

Language Implementation Patterns: Create Your Own Domain-Specific and General Programming Languages (Pragmatic Programmers) 1st Edition - Terence Parr

Writing Compilers and Interpreters: A Software Engineering Approach 3rd Edition - Ronald Mak