Rolando C. Ortiz Santiago Prof. Wilson Rivera Gallego

Victor E. Lugo Vélez ICOM4036

**Candor**

# Introduction

# It can be argued that the most integral part of programming is the understanding of and ability to create algorithms. It is an aspect of programming that can take time to get the hang of. There are many people who may have these abilities already and are simply unaware of it. These people may have the potential to be amazing programmers, but the complexity or abstract nature of a programming language can sometimes be quite intimidating. While it is not all that difficult to get the hang of a programming language with some effort and help, it can be daunting for someone with no programming experience whatsoever. Most programmers have had the experience of someone with no programming experience seeing their code and exclaiming that they simply do not understand how someone could ever understand what is happening within that wall of text. Candor aims to remove that initial apprehension from someone who knows absolutely nothing about programming. To do this, the language was designed to emulate ordinary speech. It is meant to be extremely similar to written English in order to make it as straightforward and simple as possible. Its main goal is to ease new programmers into the concept of programming and remove the barrier of a common programming language’s complexity. As it is a programming language aimed mainly at beginners, the projects meant to be made with the language are not very complex ones.

# Language Tutorial

# Candor is incredibly simple to use. Here are some basic snippets of code demonstrating its syntax:

# Addition:

# A adds B

# Subtraction:

# A subtract B

# Multiplication:

# A times B

# Division:

# A divides B

# For loop:

# for i between A and B

# *code in for loop goes here*

# endblock

# While loop:

# while A lessThan/greaterThan/sameAs B

# *code in while loop goes here*

# endblock

# If statement:

# if A lessThan/greaterThan/sameAs B

# *code in if block goes here*

# endblock

# Attribute assignment: Where “square” is an existing variable and x is its new attribute

# square s x equals 10

# Language Development

# Candor is a Python-based programming language. It is converted from its form directly into Python. To do this, Candor makes use of Python’s PLY libraries. The translation is done by way of a lexical analyzer and a syntax analyzer. Firstly, from the IDLE Python GUI, the user must run CandorLex.py to create the lexical analyzer. Within this file, PLY’s lex function is used to make a lexical analyzer. When a user writes a line of code and presses enter from IDLE, the line is analyzed by the syntax analyzer, CandorParse.py, uses the regular expressions contained within to translate the line to its Python equivalent.

# Translator Architecture:

# Lexical Analyzer = PLY

# Syntax Analyzer = PLY

# Intermediate code file = Python

# Describe the interfaces between modules.

# CandorLex module sends an array of tokens to the CandorParse module that outputs an intermediate code file.

# Software development environment used:

# Python 2.7.11 IDLE

# 

# Test Methodology:

# Each regular expression was tested individually using one line of code per test to ensure output was the expected outcome. Afterward, each regular expression was tested as part of a complete program to see if the output was the correct Python equivalent.

# 

# Conclusion

# In creating Candor, it became much more evident just how it is that a programming language works. The lexical and syntax analyzers are the basis of all language’s structures. A lot of care and planning goes into making sure a programming language works as intended. Keywords have to be selected, the syntax has to be clearly defined, among other things. All of this requires one to have a well-defined structure for the language. Even after all this planning is done, implementing both a lexical and syntax analyzer at once seems like a somewhat difficult process, but thankfully, PLY is made specifically for this kind of work. After having worked on Candor, the matter of how exactly it is that programming languages and compilers work has become much less of a mystery than it was prior to it.