Kennesaw State University

Department of Computer Science

CS 4308

Concepts of Programming Languages

Section 01

Project Deliverable 2

Charles Wildner: [**cwildner@students.kennesaw.edu**](mailto:cwildner@students.kennesaw.edu)

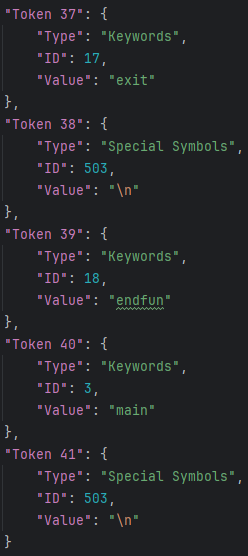
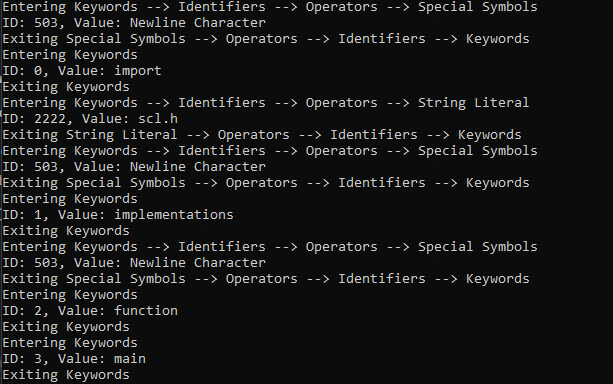
Nick Tanner: [**ntanner9@students.kennesaw.edu**](mailto:ntanner9@students.kennesaw.edu)

Rachel Anderson: [**rande147@students.kennesaw.edu**](mailto:rande147@students.kennesaw.edu)

Selam Kelil: [**skelil@students.kennesaw.edu**](mailto:skelil@students.kennesaw.edu)

Zachary Carruthers: [**zcarruth@students.kennesaw.edu**](mailto:zcarruth@students.kennesaw.edu)

**Date:** 3/24/2024

1. **Initial Problem Statement**
   1. The primary task for the second deliverable is to develop a complete parser program component of an interpreter for a specific subset of the SCL (Structured Control Language) programming language, written in BNF form, to work alongside the scanner program that was developed in the first deliverable. This involves creating a parser program that initializes the scanner with the input file string and performs the necessary computation steps on the input files to create a parse tree representation of the program. This is a crucial component for the last stage of the project, as the parser program lays the foundation for the interpreter’s ability in conjunction with the scanner program to process SCL programs accurately.
2. **Summary/Purpose**
   1. This report's purpose is to document the development process and outcomes of deliverable 2, focusing on the implementation of the parser for the SCL interpreter project. The primary objectives for deliverable 2 are:
      1. Implementing the Parser: Creating a parser based on a subset of SCL language, written in BNF, to be implemented along with the scanner previously developed in deliverable 1.
      2. Demonstrating the Parse: After the creation of the parser, it will be demonstrated to the reader that it works correctly; this will be shown as the output of the implementation in section 4.2 of this report.
   2. By completing deliverable 2, we aim to establish a complete parser for the SCL interpreter project, enabling accurate implementation and demonstration of the parser of SCL programs. This report will detail the design decisions, implementation strategies, and testing procedures employed during the development of the scanner.
3. **Solution**
   1. Parser: The parser we present here begins by first taking in the path to an SCL file. The parser then invokes the scanner code from the first deliverable which produces the JSON file that it needs to execute. With this file now in place and populated, each of the tokens is extracted and analyzed to determine its position in the grammatical hierarchy we have designed. This information is then transformed into a parse tree, with output being displayed within the console to display the hierarchy of keywords and their order within the tree.
4. **Data/Results**
   1. Excerpt from Input file:  
        
      The parser file reads in the JSON file produced by the scanner.
   2. Output result:  
      The output of the parser program, showing the ‘branch’ of the parse tree for each token.
5. **Conclusion**

In conclusion, we have created a Python program that creates a complete parser program for a subset of the SCL language, working jointly with the developed scanner from the previous deliverable. The program works by using the public functions, parse\_tokens\_from\_file(file\_path), and print\_token\_hierarchy() together with the existing scanner program to run the parser program and create the parse tree successfully. After being run the user can see the path in which the parser took while running as it went through the parse tree. All files, including parser, scanner, JSON, and SCL program files are included.

1. **References**

[1] Jose M. Garrido, “Notes On Project”, 2017. Supplied with assignment materials.