**1. Introduction**

// What is this project about? A high-level introduction

// Why you did this? Motivation?

// Summary of the project, what you achieved?

// Introduction to sections that follow.

**2. Background**

// What is a compiler? What is its purpose? What are its

// components?. How do these components work together?

// Some real­life examples.

A compiler is a program that takes a text file written in some language and creates byte-code for the written program that can be executed on some target machine.

The purpose of compilers is to allow programmers to develop using higher-level, more powerful languages that are to read and write than lower-level languages (I.E.: c or assembly).

Compilers are partitioned into 4 components to reduce coupling and simplify the compilation process. These 4 components are:

* Scanner
* Parser
* Symbol Table
* Code Generation

**Scanner**

The scanner reads the text file provided to the compiler to be into bye-code. It performs a lexical analysis of the in order to create a series of symbols that correspond to individual segments of the code. A symbol may represent a / method name, an operation like '+' or '-', open and parentheses / brackets, etc. This series of symbols the code in the compiled text file, and is passed to the Parser to.

**Parser**

The parser performs syntactical analysis on the symbols provided

to it by the scanner. These symbols are put into a parse tree

which defines the structure of the written program. No meaning

attached to the parse tree yet; only the structure of the program

is determined by the parser.

**3. Methods and Discussion**

// a. ANTLR Setup

ANTLR is a program which generates a scanner and parser for the user given a config file (of the file format .g4). G4-files contain 2 sections: the lexer and the parser. This program is provided by ANTLR in the form of a jar named 'antlr-4.7.1-complete.jar'. This

jar contains multiple programs, including one that we used to generate a lexer and a parser.

As part of the process of setting up ANTLR, we followed a tutorial on using ANTLR provided by ANTLR at www.antlr.org. The tutorial provided the following .g4-file named Hello.g4:

grammar Hello;

r : 'hello' ID ;

ID : [a-z]+ ;

WS : [ \t\r\n]+ -> skip ;

After downloading and placing ANTLR's 'antlr-4.7.1-complete.jar' file-path into our environment variables, we were able to invoke the 'org.antlr.v4.Tool' program in this jar with the Hello.g4 file and an arbitrary string as input. This program scanned the string using the scanner defined in Hello.g4, and it determined if it was syntactically correct according to Hello.g4's grammar.

**4. Conclusion and Future Work**