Group Members:

1. Tamudashe P Ngara	A10485515
2. Kaleb Green	A10440255
3. Jasmine Seals	A10478537
4. Anthony Shahid	A10438219
5. Dimitri Shelton	A10469829
6. Stojan Stojanovski	A10498995
7. Chiemere Sylvanus	A10471675
8. Devon Williams	A10441713

Name of project: DEVELOPING A COMPILER

Abstract

The project was to develop a compiler which should allow the user to enter the file name of a source code following a specific list of grammar, then to detect errors and give error messages, and run the user's program if there is no grammar error.

Tech Stack

Backend: C++ Programming Language

Libraries used:

- iostream - C++ library
- fstream - used to read from files (we'll be reading line by line)
- string - all tokens are represented as strings
- queue - to store the tokens extracted from the program code
- set - to store the keywords allowed in the language syntax

Program Functionalities

Classes:

- Token - defines the token data type (value and token type)
- Lexer - to break down the lines of code into tokens

Functions:

- AddToken - helper function to adds a token to the queue
- ShowTokens - displays the tokens
- GetCodeFromFile - reads the file line by line and adds the lines into a string
    - returns *false* if no errors were found reading from the file
- GetTokenFromCode - breaks down the lines of code into tokens
- LexicalErrors - checks for syntax errors in the code before executing it.
    - returns *true* if errors are detected else, returns *false*
- CompileCode - compiles the given code if and only if there are no errors found

Error Checking

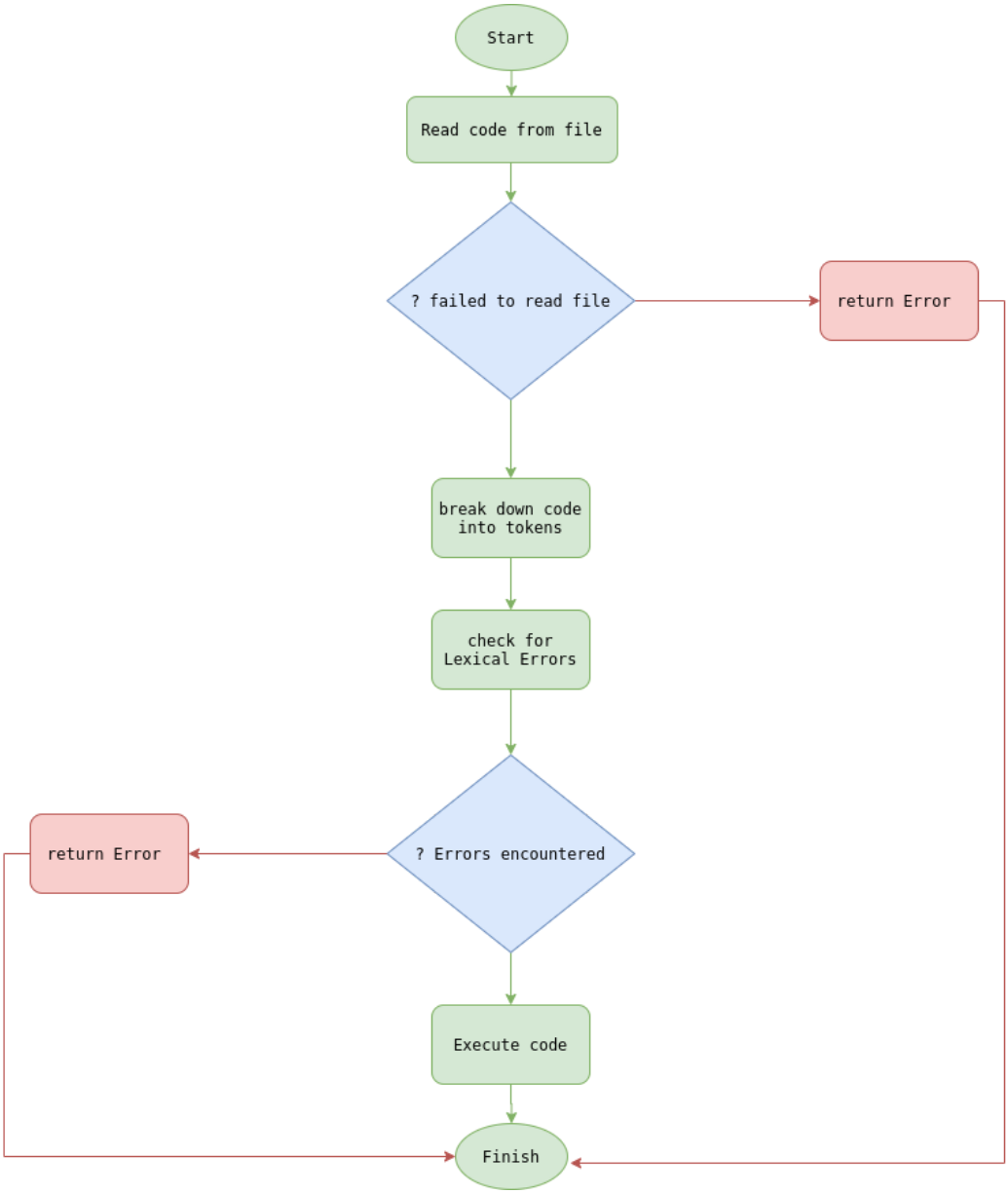
- notify user if the file path does not exist
- all lines in the code should begin with { begin; end; input; display; A; B; C }
- the first line of the code should be a "begin" identifier
- the last line of the code should be an "end" identifier
- { A; B; C } are the only variables allowed in the program
- a semicolon should be at the end of each line of code
- "display" can only display a variable or a string
- if a given file does not exist, return an error
- integers cannot be divided by zero

Language definition: Syntax

```
<program> -> begin <stmt_list> end
<stmt_list> -> <stmt>; | <stmt>; <stmt_list>
<stmt> - > <var>=<expression> | input <var> | display <var> | display "string" |
<var> -> A | B | C
<expression> -> <var> + <var> | <var>-<var> | <var>*<var> | <var>/<var> | <var>
```

Sample Program

```
display "Group 3 Compiler Code";
begin
input A;
input B;
C = A + B;
display C;
end
```


Start
Read code from file
? failed to read file
return Error
break down code into tokens
check for Lexical Errors
? Errors encountered
return Error
Execute code
Finish