Compiler Construction

Term Project

Instruction:

- This project carries the absolute **20 marks** of your theory subject having the **15 marks of implementation** and **5 marks of viva**.
- The performance of each group member will be evaluated based on his/her viva.
- The deadline to submit the project is Sunday 19th Dec 2021. Late submission will not be accepted.
- In case of plagiarism you will get zero.

Problem Statement:

You have to construct the Mini Complier for the "C++" language. It is case in-sensitive language with the given specification.

Language Description:

Keyword(case insensitive):	Same as C++
Identifiers:	Same as C++
Integer Literals:	Same as C++
Character, and string literals:	Same as C++
Operators:	<,>, <=,>=, =, ==, *, +, /,-,!=
punctuations:	[,{,(,),},] ,;,(,)
Comments:	single line and multiline just same as in C++ language
Delimiters:	Space, newline, tab

Language Specification:

Structure	Description
Program block	void main() {statements }
Declaration statement	int a; int a=5;

	int a, b;
	int a=5, b;
	int a=b+c, d=5;

Assignment statement	Operator (+, -, *, /)
	a=b+c;
	a=10*5-b;

Bool Expression	Operator (>,<,==,!=,>=,<=) 5<6 a>=10 10!=5
Repetition statement	while(boolExp) { statements }

Decision structure	• One-way
	if(boolExp) { statements }
	• Two-way
	If(boolExp) {statements }
	else { statements }
	if(boolExp) {statements }
	else If(boolExp) {statements }
	else { statements }

Input/output statement	**Input:** in>>a; cin>>a,b; cin>>a,b,c; **Output:** out<< a; cout<<a,b; cout<<a,b,c;
Function statement	**void fname(parameter_list) { statements}**

Class Definition	class Name {
	Assign Statements;
	Function Statements;
	};

Tasks

1. Implement the lexical analysis phase. [3]
2. Implement the syntax analysis phase. [5]
3. Implement symbol table manager (Register Variables). [2]
4. Apply Semantic action to evaluate data type, arithmetic and boolean expressions. [3]
5. Report error in case of error. [2]

Note:

You can implement the lexical and syntax analysis phase by using automated tool LEX and YACC respectively or you can implement it by using programming language (C++, C#, java).

Sample Input:	**Sample output:**
a source file "source.txt" containing the source code of C++ is input.	It writes all the tokens <TokenType, Lemexe> in separate file "token.txt". It writes all the Identifier in separate file "Identifier.txt". If the code is successfully parsed show the given message "Code Compiled Successfully" If code carry error try to resolve it by using error recovery technique and if it is unable to resolve the display the message "Code cannot be compiled " and also display the type and line number of error.