Compiler Construction / Compiler Design
Assignment Stage 1
Format

Name and ID of Team Members:

A team can be of 2 or 4 members (A group of three is not desired. If you are a team of 3, still you need to show 2 implementation of compiler)

Group 1	Group 2
Member 1:	Member 3:
Member 2:	Member 4:

Check list for submission:

1: Language features:-

Note that you are not giving exact copy of C Language. Try to use the knowledge of PPL Course to select the constructs of your programming language.

If you are designing language for a particular domain, not all the features required in the domain need to be implemented. A sub set of the programming language need to be designed and to be implemented. Do not make it too simple and too complex. Make sure that you can convert your features to assembly code in the later phases.

Example: If you have a data type called Matrix, you need not implement all the operators for this matrix. One arithmetic operation (+ or * or inverse) and one relation operation (< or > or ==) is enough.
Choose your constructs carefully.

Minimum / Desired features required in your language are:

- Keywords and Identifiers.
- Data types: some primitive data types, some derived data types, Arrays, structures.
- Operators and operation defined on each data type.
- Function:- function identifiers, nested functions ,arguments, parameters, return type, function call........etc.
  Note: There is nothing like inbuilt function. All the functions, either inbuilt or user defined functions will follow same syntax. Compiler will treat both as same. For compiler designer, it does not matter who is writing the functions.
- Scope rules: (Global, Local, Nested, block level......etc)
- Conditional statement
- iterative statements
- I/O operations.
- Expression
- Assignment statement (simple, mixed mode or multiple assignment ...etc)

2. Lexical units : List all the token used in your programming language.

Fill the table below.

Pattern	**Token**	**Purpose**
Example: =	TK_AssignOP	Assignment Operator
[1-9][0-0]*	TK_Num	Integer Number
While	TK_While	Keyword while

Make sure you will not miss any constructs including {"," , ";", ")" , "(" ....etc}

3. LL(1) grammar

4. Test cases

Give at least 5 different test cases representing different features of your programming language.

5. Derivation for your test cases :