

## Implementation

- Use C language to implement the stage 1 modules.
- Use of any other high level language or lexer/ parser generator packages is NOT allowed.
- Test your code with the test cases given in the language specification document.
- Generate more test cases and verify the correctness of your code.
- You will be given more test cases later.
- An appropriate interface requirement will be provided to you as you are through with the ground work. Instead of starting coding right immediately, spend time designing the structure of your compiler code.

## Interface Requirements

### Test Cases

[testcase1.txt](#) : Syntactically Correct \* [ posted on March 8, 2013. 7:30 a.m.]  
[testcase2.txt](#) : Syntactically Correct \* [ posted on March 8, 2013, 7:30 a.m.]

\* Report any discrepancy or syntactic error found in these test cases immediately.  
More testcases with or without syntax errors are likely to come up today and tomorrow.

### **ERROR REPORTING** [ posted on March 8, 2013]

**ERRORS** to be notified along with the associated message as your lexer and parser execute.

#### *Lexical Errors*

*ERROR\_1 : Identifier at line <print the line no here>is longer than the prescribed length of 20 characters*

*ERROR\_2: Unknown Symbol <print the symbol here> at line <line no here>*

*ERROR\_3: Unknown pattern <print the lexeme seen so far which does not match any pattern>*

*ERROR\_4: Specify<Any other as per your perception>*

#### Syntax Errors

*ERROR\_5: The token <specify> for lexeme<specify> does not match at line <specify no>. The expected token here is <specify the expected token here>*