

```
#ifndef mcrlex_h
#define mcrlex_h 1
//-----
// File mcrlex.h defines class Lexer.
//-----
// Author: Thomas R. Turner
// E-Mail: trturner.uco.edu
// Date: September, 2018
//-----
// Copyright September, 2018 by Thomas R. Turner
// Do not reproduce without permission from Thomas R. Turner.
//-----
//-----
// Standard C and C++ include files
//-----
#include <stdio>
#include <fstream>
#include <iostream>
//-----
//Namespaces
//-----
using namespace std;
//-----
//Function: yylex
//Function yylex is the mcrner. Function yylex returns an integer
//token code as defined above or 0 if end-of-file has been
//reached.
//-----
#ifdef __cplusplus
extern "C"
#endif
int yylex (void);
```

```
//-----  
//Token definitions  
//-----  
#define ID          1  
#define BEGAN      2  
#define END        3  
#define READ       4  
#define WRITE      5  
#define INTLIT     6  
#define COMMA      7  
#define SEMICOLON  8  
#define ASSIGN     9  
#define PLUS      10  
#define MINUS     11  
#define SLASH     12  
#define STAR      13  
#define LPAREN    14  
#define RPAREN    15  
#define ERROR     16  
//-----  
//Class Lexer defines the attributes of a Scanner  
//-----  
class Lexer {  
    int tokencode;           //Code for the most recent token found  
public:  
    Lexer(FILE* i);          //Constructor used to redirect the keyboard (stdin) to file i.  
    int Lex(void);           //Call the scanner yylex and return the code found by yylex  
    int FetchTokenCode(void); //Return the code of the most recent token  
    void StoreTokenCode(int T); //Store the token code.  
    char* FetchSpelling(void); //Return the spelling of the most recent token  
};  
#endif
```