

## CS 460 – Project 1 - FAQ

**Question:** I noticed that you use cout to show what the input file is in project1.cpp.

```
cout << "Input file: " << argv[1] << endl << endl;
```

Should we remove that line before submitting?

**Answer:** You can leave this line in the Project1.cpp file.

**Question:** If we see 'modulo' as a lexeme it's a MODULO\_T. if we see 'modulo?' as a lexeme should that also prioritize the modulo and therefore have a MODULO\_T and then backup, so in the next iteration we read the '?'

**Answer:** Yes – based on the examples in the spec, if an IDKEY ends in a ? and is not one of the defined predicates, it should be evaluated without the ? and the next call to GetToken should start with the ?.

**Question:** I noticed that there are 2 tokens that don't seem to have a corresponding lexeme. TRUE\_T and FALSE\_T. What are there corresponding lexemes?

**Answer:** You can **remove TRUE\_T and FALSE\_T** – they will not be needed until we get to Project 2.

**Question:** For the .lst file, do we need to use .tell() and .seekg (unsigned int) for dynamically building the .list file? Or, can we use a vector of strings to store the error messages then print them all to the .list file at the end of the line?

**Answer:** tell() or tellp()? Neither of these methods is necessary. You should use the getline command to read a line from the input file into a string then immediately write the string to the .lst file. You will then be able to walk through the string and write error messages as the errors are detected.

**Question:** Why are there 2 separate methods for getting the token type and the lexeme? Would it be better to have one method return them both so it doesn't rely on calling the methods in a specific order?

**Answer:** This would be a good modification to the LexicalAnalyzer class. Currently we will not be making any modifications to the class. However, I will take the idea into consideration for a future project. Thank you.

**Question:** I just wanted to clarify, for the LISTOP1\_T variables (car, caddr, etc.) they need to have a space in between them in order to be considered a LISTOP1\_T token right? Based on the example input and output given in P1-2.pl460 and P1-2.p1, it seems that when they are connected together (carcdr...), they are considered an IDENT\_T token so I am pretty sure this is accurate but, I just wanted to be sure.

**Answer:** Yes. To be identified as a LISTOP\_T, the r may not be immediately followed by another letter, a digit, an underscore, or a question mark. caddr+cdr is a LISTOP\_T followed by a PLUS\_T and another LISTOP\_T. caddr1car is an IDENT\_T.

**Question:** Do we use 'getline' from our 'getToken()' function to get each line?

**Answer:** Yes. To do this, you should create a loop to bypass whitespace at the beginning of your GetToken function.

**Question:** Regarding the constructor that takes in the "fileNamePrefix" as a parameter. Are we supposed to be passing it the prefix (Lex tokenizer("P1-1");) and opening the file from our constructor?

**Answer:** Take a look at how the constructor for LexicalAnalyzer is called in SyntacticAnalyzer.cpp and how the constructor for SyntacticAnalyzer is called in Project1.cpp.

**Question:** What is the purpose of the function: string GetTokenName (token\_type token) If I gave this the token CONS\_T, would it give back the string "CONS\_T" or would it give back "key words"?

**Answer:** This function should return the string version of the name assigned to the element of the enumerated type. GetTokenName (CONS\_T) should return "CONS\_T".

**Question:** Does this language we're writing the lexer for need to have a space between identifiers? For example, is newlinecond two separate tokens or a user defined identifier:

**Answer:** newlinecond is a single user defined identifier. newline+cond would be a NEWLINE\_T followed by a PLUS\_T followed by a COND\_T.

**Question:** Is 1+2 acceptable for the language? Or, does this need to be 1 + 2

**Answer:** Both are acceptable and would result in a NUMLIT\_T followed by a PLUS\_T followed by a NUMLIT\_T.

**Question:** Is 1++2 the same as 1 + +2 or is it an error?

**Answer:** Yes, since our language does not include a ++ operator, 1 ++2 is equivalent to 1 + +2

**Question:** Can we change the numbering of the enumerating?

**Answer:** Yes, you may modify the token\_type enumerated type as you need. All new elements of the type should be added before MAX\_TOKENS. **LISTOP2\_T needs to be changed to CONS\_T.**

**Question:** What is IDKEY\_T in our case? Do we first get IDKEY\_T and from there check in our map if that specific lexeme has a different token type and if not, set it to IDENT\_T?

**Answer:** IDKEY\_T is a token that will be used internally. Once a lexeme is identified, a more specific token should be returned by the function. A map is probably the most efficient way to correlate a key word lexeme with its specific token. For example, display and DISPLAY\_T.

**Question:** They're error\_BU and error\_NBU, so then what's error\_T?

**Answer:** Like IDKEY\_T, ERROR\_BU and ERROR\_NBU should be used internally. If either type of error is found the ERROR\_T token should be returned by your GetToken function.

**Question:** Can I have a vector of chars so that getting the column for that char, to use for accessing my table? If so How would I treat the 'other' category?

**Answer:** You will need to develop a technique for associating specific characters with columns in your state table. In the RE1/RE2 example, there is a function called GetColumn that does this task. This is a simple function – but not an efficient way to solve the problem if there are a lot of possible characters in the alphabet associated with the regular expression.

```
int getColumn (char one)
{
    if (isspace (one))
        return 0;
    if (one == 'a')
        return 1;
    if (one == 'b')
        return 2;
    if (one == 'c')
        return 3;
    if (one == 'x')
        return 4;
    if (one == 'y')
        return 5;
    if (one == 'z')
        return 6;
    return 7;
}
```

**Question:** What is the functionality of reportError function?

**Answer:** The ReportError function should write a message containing the line number, position on the line, and error message passed to it to the .lst file.

**Question:** Question

**Answer:** Answer