CSCE 121 Assignment #5

due October 16 to eCampus

Demo in the lab by 10/09

Objectives: *This assignment gives you some experience with designing, writing, and testing C++ programs that use fundamental computations, user defined types, exceptions, similarity and differences between arrays and STL vectors.*

All work should be pre-graded in the labs and turned in to eCampus by the deadline. For programs, you need to turn in only the source code (not object code or executable code). Your code will be tested using a g++ compiler. Therefore, you are welcome to develop your program in Visual Studio or Xcode, but make sure your code also compiles using g++ and runs on a Linux machine.

1. (10 points) Create a text file, called README, in which you provide:

   - Your First Name, Last Name, UIN, Section Number, User Name, E-mail address
   - State the Aggie Honor statement:

     I certify that I have listed all the sources that I used to develop the solutions and code to the submitted work.

     On my honor as an Aggie, I have neither given nor received any unauthorized help on this academic work.

     Your Name Date
   - List any resources used such as webpages (provide URL). Do not mention the textbook and discussions with the Instructor, TA, or Peer Teachers.
   - List any known problems with the assignment you are turning in. For example, if you know your code does not run correctly, state that. This should be a short explanation.
   - Provide list of problems
   - Provide a short description or pseudocode for the program.
   - Write how you tested your program(s) for correctness and how you used exceptions for the error handling.
   - Submit to eCampus an electronic version of the README file along with your C++ code for the problems listed below and a hard copy to your TA.

2. (30 points) **Demo in the lab by (10/09)** Write a C++ program that takes arithmetic expressions as input separated by a semicolon (`;`) and terminated by the character `q`, and parses it into tokens. The arithmetic expression does not need to be correct here because we do not evaluate it. Use a loop to read input characters (one character at a time).

(a) Input numbers are integers in the range 0 to 9 (so they are characters as well).

(b) Skip any white spaces in input expressions.

(c) The lecture notes or the text "*Programming Principles and Practice Using C++*" by B. Stroustrup will help you with understanding of the token concept. Save the tokens from the parsing stage in an array and in a vector of type `Token`.

Note that type `Token` is not provided by C++ language like `int` or `double` types but a programmer has the possibility to introduce such a type to simplify a solution to a problem.

(d) This is a list of tokens: | `(` | `)` | `+` | `-` | `*` | `/` | `digits` | `;` | `q` | where a semicolon (`;`) is used for separating expressions, and the letter q is used to terminate input reading. Here is an example of input expressions (it does not need to be in one line):

```
2+3; (3 + 2) * 4 / 5; 6; q
```

(e) Print the tokens from the array/vector (one per line) and display the token's members kind and value if it exists.

(f) Example of a partial output corresponding to the above input (do not output `value` for non-numbers):

```
kind: n
value: 2
kind: +
kind: n
value: 3
kind: ;
kind: (
...
```

`struct Token` should consist of these members:

```
char kind; // what kind of a token
int value; // for numbers only: a value
// constructors:
Token(char ch) : kind(ch), value(0) {}
Token(char ch, int val) : kind(ch), value(val) {}
```

The member `kind` is simply equal to the token, except for numbers, where `kind` is equal to the character `'n'`, and then the value is equal to the face value of a digit (value is not used for other tokens).

3. (30 points) **Programing assignment 5 part 1.** Write the first version of the calculator program based on the type `Token` introduced above in the lab portion.

(a) Write the exception class `Bad_Token`

(b) Write the function `get_token()` which returns tokens corresponding to input expression. It is not a member of the struct `Token`. If it cannot recognize a token then the exception `Bad_Token` should be thrown.

(c) Numbers can be now any nonnegative integers, not only one-digit numbers.

(d) Write the functions based on the grammar for the arithmetic expression: `expression`, `term`, `primary` to evaluate the input expression.

(e) Explain in the README file why this version of the calculator program is not correct. Justify what is wrong and why. How should it be fixed?

(f) The output (if it exists) should consist of an input expression, the equal sign, and its result in one line of each arithmetic expression. Examples:

```
2+3 = 5
(1+2)*12 = 36
(2*3+1)/3 = 2
```

4. (30 points) **Programing assignment 5 part 2.** Write the second version of the calculator program based on type `Token` and `Token_stream` introduced above in the lab portion, and the functions based on the grammar for the arithmetic expression (`expression`, `term`, `primary`).

   The user-defined class `Token_stream` consists of the private and public members:

```
public: // public part
Token get_token(); //identifies tokens; throws Bad_Token() exception
void putback(Token t); // put a token back to the buffer
Token_stream() : full(false), buffer(0) {} //constructor
private: // private part
bool full; // is there a token in the buffer?
Token buffer; // to keep a token
```

   - Note that the function `get_token()` is now a member function of this class and it uses the buffer to read tokens. That is, first it tries to get a token from the buffer, and if there is no token in the buffer then it reads a token from input (using `cin`).
   - The output should consist of an input expression, the equal sign, and its result in one line of each arithmetic expression. Examples:

```
2+3 = 5
(1+2)*12 = 36
(2*3+1)/3 = 2
```

   - Explain in the README file why the second version of the calculator program is correct and provide answers to the questions below.
     - Which token was created with help of the C++ library?
     - Why do we need to use the `putback()` function in the token stream to evaluate an expression correctly?

5. **Submit to eCampus for grading a zip/tar file that consists of the files:** README, sources for the lab part, and the first and the second of the calculator program implementations.

   Important: The zip/tar file name should be in this format: **LastName_UIN_calculator**.