CS 2412 assignment 1

Due on Wednesday, Oct 1, 2014

Each student is required to do this assignment **individually** and to hand in the following:

- All the documents of the solved problems, including your statements, source codes, input and output data etc. You may include any materials shown your work on this assignment.
- Send your computer programs to `cs2412@peace.lakeheadu.ca`.

Stable your answer sheets together and include the following information clearly Name, Student Number, Assignment Number, Course Number (CS 2412)

The score of the assignment will depend on:

- Format, readability and documentation: 25 %
- Correctness: 75 %

Deposit (submit) your answer set in Classroom on due date.

Note: you need to develop good quality programs, not just solve the problem correctly. You need to write and compile different functions separately, make your own header file and maybe makefile if you use gcc, give enough comments in your source codes, etc.

This assignment includes 4 problems. However, Problem 1 is from lab work.

Problem 1.

Using dynamic memory allocation to implement a stack of integers with the following functions:

`CreateStack, ClearStack, StackEmpty, StackFull, Pop, Push`

You can use your work from lab for this problem.

Problem 2.

Using the functions of Problem 1 to write a function `reverse` which reverses a list of integers. The program can read a list of integers from a file and output a list in a reverse order.

Problem 3.

Make a C program that changes a decimal integer to a hexadecimal number. If the remainder is 10, 11, 12, 13 14 or 15, print A, B, C, D, E, or F, respectively. (Hint: Review Programs 3-15, 3-16 of the text book for Problems 2 and 3).

Problem 4.

Write a program that implements the infix-to-postfix notation. The program should read an infix string consisting of multi-characters variable identities, parentheses, and the `+, -, *` and `/` operations; call the conversion algorithm; and then print the resulting postfix expression. After transforming an algorithm, it should loop and convert another infix string. Test your program. (Hint: You need stack for characters for Problems 3 and 4).