Programming Assignment 1

TCSS 142 Autumn 2014

# Due: Friday, Nov. 14, 8 a.m.

Given the problem statement below, complete the following

* Program schedule chart (5 pts)
* Algorithm written in English / pseudocode (10 pts)
* Test plan (15 pts)
* Program that runs correctly including (70 pts)
* Coding style (meaningful identifiers, indentation, etc.)
* Comments (including the header with your name and course info at the top of the file)

In order to receive credit for this part, your code MUST run (comment out non-working parts to have your code run and for us to see that you attempted a solution)

* Extra credit :
* algorithm as a flowchart (extra credit 10 pts)
* modular approach - functions (extra credit 10 pts)

# You are NOT allowed to help one another with this program or use somebody else's code – check the rules listed on the syllabus. However, you may seek help from CSS mentors.

# Problem statement

For this program, you are to design and implement a Roman numeral calculator for positive integers. Your program is to input two Roman numbers and an arithmetic operator and to print the result of the operation to the screen - in both Roman and Arabic notations. The arithmetic operators that your program should recognize in the input are +, -, \*, and /. These operators are used to perform Python operations of integer addition, subtraction, multiplication, and division.The program is to be repeated as many times as the user wishes.

You may not use Python constructs that have not been discussed in class so far.

# Input

An input to your program will be interactive. When entering Roman numerals, we will assume

a purely additive notation, in which a number is simply the sum of its digits (4 equals IIII in this notation). Each number starts with the digit of the highest value and ends with the digit of the smallest value. The values of the Roman digits are as follows:

I 1

V 5

X 10

L 50

C 100

D 500

M 1000

After the number is entered, you are to make sure that no illegal digits or arithmetic operators were entered. When an invalid input occurs, the program is not to exit, but to ask the user for the given value again – until the valid string is entered. You may assume that Roman numerals are entered in the right order (i.e. MDC, not CDM or LIL) and as positive integers.

# Output

In addition to printing the Roman and Arabic equivalents of the calculation, each time a number is entered, you are to echo print it to the screen in its Arabic format. The following is a sample run of the program:

Enter the first number: LXXXXI

The first number is 91

Enter the second number: VIIII

The second number is 9

Enter the desired arithmetic operation: +

The result is C (100)

Do you want to repeat the program? Enter Y for yes, N for no: N

Make sure you guard against the division by zero. If it happens, display an error message. Also, subtraction could produce negative numbers, in which case print the number in its Roman format with the negative sign proceeding the actual number (i.e. –C for –100).

# Pseudocode and Flowchart

# A pseudocode is an informal description of an algorithm. For our purposes, it should be written in steps and include formulas, if any. There are a couple of examples provided in the textbook: p. 127, p. 176, p. 184 (should contain formulas as well), p. 200.

A flowchart is a graphical representation of an algorithm and two good examples are provided in the textbook on p. 126 and p. 136. Note that the flowchart elements include a rectangle to indicate calculations, a diamond to indicate a decision point, and a parallelogram to indicate input/output.

# Program Submission

If you want your assignment to be graded, it has to be compatible with our platform, namely Python3.4.1 The source code is to be called yourNetid\_project1.py

On or before the due date, use the link posted in Canvas next to Programming Assignment 1 to submit your code and all associated documentation. Make sure you know how to do that before the due date since late assignments will not be accepted. Valid documentation file formats are: doc, docx, rtf, pdf.

PROGRAM SCHEDULE

**Date Completed**

**Milestone Planned Actual**

Assignment received.

Requirements understood;

detailed specification recorded.

Top level of design complete.

All levels of top-down design complete

Coding complete (clean compile).

Testing planned.

Testing complete.

Program ready to turn in; all

external and internal documentation

complete.

Assignment turned in.

**TEST PLAN**

Reason for test case Input values Expected output\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_