Software Development with C++ Templates (541)

Lab Submission 1 Checklist

[ ] 1.1. Fraction Class

[ ] Fraction class with arithmetic operators +, -, *, /
[ ] Fractions always presented in normalized form
[ ] Error handling
[ ] Test driver
[ ] Comments (shortcomings, etc.):

[ ] 2.1. Declaration vs Definition

A text document describing with a section for the elements to be put into the header (.h) and a section for what should go into the source (.cc) file is sufficient. Other formats (eg, a .h and a .cc file) are also acceptable. Important is a quick comment for each element why it should go into the respective file.

[ ] 2.2. Fraction class with separate compilation

[ ] fraction.h and fraction.cc files containing the fraction
[ ] util.h and util.cc files (or other) for other functions
[ ] .cc file with main which makes use of the above
[ ] Test driver
[ ] Comments (shortcomings, etc.):

[ ] 2.3. RPN Calculator

The user-interface does not have to follow precisely the sample given in the exercise.
[ ] Arithmetic operations, +, -, *, /
[ ] Error handling
[ ] Comments (shortcomings, etc.):

[ ] 2.4. Spell Checker

[ ] Spell checking words, identify punctuation characters, etc. correctly
[ ] Comments (shortcomings, etc.):

[ ] 3.1. Function that the compiler cannot inline

[ ] Briefly explain here:

[ ] 3.2. Persistent Vector
[ ] Vector can write data to a file, can read it in
[ ] Test driver with different pvectors with different data types
[ ] Comments (shortcomings, etc.):

[ ] 3.3. RPN Calculator with pvector
[ ] Comments (shortcomings, etc.):

[ ] 3.4. RPN Calculator with pvector and Templates
[ ] Upon restarting the calculator, we get back the stack
[ ] Tested with int
[ ] Tested with fraction
[ ] Comments (shortcomings, etc.):

[ ] 4.1. Persistent Vector with Traits
[ ] pvector with traits
[ ] Test driver for pvector<int>
[ ] Test driver for pvector<string>
[ ] Test driver for pvector<fraction>
[ ] pset with traits
[ ] Test driver for different data types
[ ] Comments (shortcomings, etc.):

[ ] 4.2. Interactive dictionary with pset
[ ] Upon restarting the dictionary, we get back the newly inserted words?
[ ] Uses pset from 4.1
[ ] Comments (shortcomings, etc.):

[ ] 4.3. RPN Calculator with standard library algorithm
[ ] Comments (shortcomings, etc.):

[ ] 4.4. Combineops
[ ] Give a short sample here:

[ ] 4.5. Simple Connect 4
[ ] Comments (shortcomings, etc.):

[ ] 4.6. Connect 4 with template-based player

[ ] Human player uses the player interface
[ ] Computer player uses the player interface
[ ] Computer player identifies whether he can win
[ ] How does the computer player check whether he can win:

[ ] Comments (shortcomings, etc.):

[ ] 5.1. Rectangles and Squares

[ ] Your advice to the computer scientists: