CS1020E: Data Structures and Algorithms I

Tutorial 4 – Template, ADT

(Week starting 9 February 2015)

1. Calculator

Calculator is a very simple computation machine to add, subtract, divide, and multiply numbers. A simple computer program could easily emulate the behavior given the proper input. However, numbers are not always integer, it can be real-numbers (approximated by double in C++). As such, every computation would require as much duplication as the number of types of numbers in our programming language.

Template, on the other hand, simplifies the coding by automatically generating the code for each data type. In other words, template is automated code duplication mechanism. Your job here is to design and implement a calculator using template *without using Math library in C++*.

a) Implement a template calculator with the following functions: `add`, `sub`, `mult`, and `div` to represent addition, subtraction, multiplication and division operation, respectively. Each of these functions takes in two arguments of the same type and returns that type in the return value.

b) Add a function for exponentiation using repeated multiplication for your function that takes in two values, the base, and the exponent. What should be the data type of the arguments?

c) Using your code, add a code to perform addition of two `strings`. Would the program still compile? Explain.

d) Now, add a code to perform division of two `strings`. Would the program still compile? Explain.

2. Template and Operator Overloading

Given the calculator above, we propose a constraint that every computation would require as much duplication as the number of types of numbers in our programming language. Our solution to the calculator problem uses template. Now, we will test the usefulness of using template in building our calculator by actually creating a new type of number: `Ratio`.

Our `Ratio` class will be defined as a mathematical ratio with more relaxation. It is mathematical ratio in the sense that it has numerator and denominator in the form of $\frac{numerator}{denominator}$. However, unlike mathematical ratio, the denominator need not be reduced to its lowest term. Similarly, we would allow using double as both the numerator and denominator.

a) Code a class representing `Ratio`. It should have numerator and denominator as its data member, and a single function `value()` that returns the `Ratio` class as a `double`.

b) Override the `+`, `-`, `*`, and `/` operators to take in two `Ratio` class as its arguments and return a new `Ratio` representing the addition, subtraction, multiplication, and division operation on `Ratio`.

Please try to make sure that your operator overloading would make it work in conjunction with the **calculator** above.

CS1020E: Data Structures and Algorithms I

3. Vector, Pair and Algorithm

You are given a list *L* of *n* students. Each student *k-th* in the list *L* is recorded by name (no space), matric number, and CAP. Your task is to sort the list by decreasing CAP. If two students have the same CAP, sort them by lexicographic order of their names. Print the sorted list of student with following information:

`Name, Matric number, CAP, and the position in original list L.`

Sample input:

```
4
Anna       A006 3.0
Bertha     A007 2.5
Clare      A008 4.0
Diane      A009 3.0
```

Sample out:

```
Clare      A008  4.0   3
Anna       A006  3.0   1
Diane      A009  3.0   4
Bertha     A007  2.5   2
```

Hints: You can use vector of pairs of pairs and sort algorithm.

4. Iterator

In this question, we will play a word game using `string`. In C++, `string` is treated logically to be equivalent to list of characters. So, as you may expect, `string` has its own `iterator`. Hence, you are required to use an `iterator` (or a few `iterators`) to solve this problem.

Palindrome:

Palindrome is a word which can be written the same forward or backward. In our definition, we extend the definition of palindrome to sentences. It is a sentence that the alphabets (i.e. exclude white spaces and punctuation marks) are the same when run forward or backward. For example, "Madam, I'm Adam" is a palindrome. Note that characters are not case-sensitive. Write a function to recognize a palindrome when given a single string using only a single loop-statement, without duplicating the string, and a maximum of two iterators.