// OOP345 Workshop 3: Templates

// File: reflect.txt

// Name: Jingmin (Jessica) Zhou

// Email: jzhou175@myseneca.ca

// ID: 119766194

// Date: 06/12/2020

In Workshop 3 (Templates), I have learned how to define and code a class template and derivate another class template from it (e.g. class Set and SetSummable), specialize a member function of a class template to process a particular type (e.g. member function sum() in class PairSummable), declare a static data member in a class template and initialize it (e.g. data member width in class SetSummable).

Firstly, explicit specialization of the member function in a class template is due to function logic based on the template type. For example, the return statement of sum() is based on the type V. If V takes the std::string type, the sum value should be the concatenated result of the strings, as required in this workshop concatenating the values stored using , as separator. If V takes other types, like int or float, the sum can be generated by using “+” for addition.

Secondly, the initial value for a summation in class PairSummable is based on the type of the value in the key-value pair, so the initial summation needs to be specialized. For V as std::string, the initValue should be set to an empty string “”, while is should be set to be 0 for V as int.

Thirdly, because class variable lasts the lifetime of the class definition and holds a value that all instances of the class can share, class variable is declared inside of class definition, but defined and initialized outside of class. If the class variable is initialized in class definition, it will only belong to one instance of the class, not the whole class. For example, static data member width can be accessed by any object of the PairSummable and updated dependent on the length of key and its lifetime will last within the class definition.

===========================================================================================================================================

Quiz 3 Reflection (55/58)

Q21 Code5: It can be complied successfully, but I though it failed with compiling error.

Q Code5: It cannot be linked successfully, but I thought the errors were from compiling stage.

Q Code14: when index = 3, I ignored the condition (i == n / 2) and no throw and catch happened. It should be throw I and execute the catch (int& value) block.