// OOP345 Workshop 1: Across Translation Units

// File: reflect.txt

// Name: Jingmin (Jessica) Zhou

// Email: jzhou175@myseneca.ca

// ID: 119766194

// Date: 05/29/2020

In Workshop 1 (Across Translation Units), I have learned how to interface the operating system through main() function with command-line arguments, how to make the memory utilization more efficient through dynamic memory, and know more about the concept of linkage, like external and internal linkages.

Firstly, internal linkage refers to an entity only in a scope of its translation unit, while external linkage refers to an entity accessible through the whole program, which is the combination of all translation units. For example, the variable g\_sysClock (with extern keyword) in the workshop is defined in the Event module and also accessible through the whole application, while the variable counter (with static) is defined in function display() and only visible to its scope and other scopes within its translation unit.

Secondly, static variables defined in a scope can be only initialized once and their values are shared through function calls or by all objects of class. In this workshop, static variable counter is defined in function display() and used to record the times of display() calls, which is very convenient and helpful counter for the program.

Thirdly, dynamic memory allocation is used instead of static memory for Event description property, which upgrades the Event class and prevents the waste of memory because memory is allocated only based on the length of user-defined string. Copy constroctor and assignment operator need to be implemented for deep copy and assignment. For example, a copy Event is created from currentEvent and then copy Event is assigned to archive[] Event in w1\_p2.cpp file.

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

Quiz 0&1 Reflection

Quiz0: (34/39) with 5 questions answered incorrectly

Q1: The access modifier protected: inaccessible outside the class but accessible by any derived class of that class.

Q4: a helper function: global functions that overload operators

Q18: object-oriented classes may contain: hiding all implementation details within itself, inheriting the structure of another class and a single interface providing multiple implementations.

Q35: The keyword virtual on a member function's declaration specifies: the compiler inserts code that binds the call to most derived version of the member function based on the dynamic type.

Q39: The stages of creating an executable are: linkage

Quiz1: (42/43) with 1 question answered incorrectly

I chose the correct answer, but probably saved it failed or forgot to save it.