Name: Jingmin Zhou (Jessica)

Student ID: 119766194

Email: [jzhou175@myseneca.ca](mailto:jzhou175@myseneca.ca)

In the Workshop 2 (Dynamic Memory & Function Overloading), I have learned how to create and use references to modify content of variables, how to overload functions, and how to allocate dynamic memory at runtime and deallocate that memory when it is no longer required.

Firstly, I allocated and deallocated dynamic memory for an array, e.g. allocate memory (gift.wrap = Wrapping[gift.wrap\_layers];) in function wrap(Gift&) and deallocate that memory (delete[] gift.wrap; gift.wrap = nullptr;) in function unwrap(Gift&).

Secondly, I incorporated dynamic memory along with user input, like the allocation memory to store many Wrappings in the Gift based on the wrap\_layers entered from user.

Thirdly, the concept of overloaded functions is enhanced through gifting functions in Gift module. Four gifting functions (gifting(char\* g\_description), gifting(double& g\_price), gifting(int& g\_units) and gifting(Gift& gift)) distinguish different meanings through different types of parameters.

Finally, functions wrap(Gift&), unwrap(Gift&) and gifting(Gift&) take Gift reference as parameter, which means that these functions take an object rather than a copy. Pass-by-reference is likely to pass-by-address mechanism in C, but it makes code more readable than pass-by-address code.