/\*

Name: Tian Zheng

Student Number: 155 394 174

Email: tzheng26@myseneca.ca

\*/

1. The advantage that smart pointer syntax provides.

With smart pointer, I do not have to explicitly destroy the object when it is not useful any more. However, with raw pointer, I have to code to delete the pointer:

priceList += p;

delete p;

p = nullptr;

2. List the difference between raw and smart pointer syntax in your solution.

(1). To declare a raw pointer, the syntax is: T\* src

While to declare a smart pointer, the syntax is: std::unique\_ptr<T> src

(2). To define a raw pointer, the syntax is: Product\* p = new Product(desc[i].desc, price[j].price);

While to define a smart pointer, the syntax is: std::unique\_ptr<Product> p(new Product(desc[i].desc, price[j].price));

(3). To use a raw pointer, the syntax is: priceList += p;

While to use a smart pointer, the syntax is: priceList += std::move(p);

(4). When a raw pointer is no longer useful, I have to delete it, and the syntax is:

delete p;

p = nullptr;

While for a smart pointer, there is no need performing the deleting step.