Name: Tai Ngoc Bui
CSCI 3731
Hw10

1. What is the fundamental difference between how Java and C++ implement collections?

In Java, as Object is the super class of all classes, thus, we can put any kind of object in a collection of Objects. Unlike Java, C++ does not have a universal base class like Java's Object class. This distinction has profound implications when it comes to collections. C++ uses templates to overcome this lack of a universal base class, allowing users to create standard collections for different types of objects.

2. What is the difference between a template and a class?

Template is not a class, but a specification for how to make a class. Moreover, template's arguments are listed inside angle brackets, while class' arguments are listed inside parentheses. On the other hand, while class's definition is defined in .cc files, template's methods are listed in the header file.

3. What are some of the drawbacks of templates?

Each time a template is instantiated, the compiler creates and compiles a separate version of the class. Moreover, the compiler has to recompile all of the class's methods for each instantiation. Thus, templates can make executable files larger and slow down compilation.

Another problem with template is that templates with arguments as different primitive types such as double and float are treated as completely unrelated classes and not allowed to cast one to another. Moreover, template does not allow writing a method that would take either as an argument, unless the method is itself a template.

4. What is an iterator?

Unlike Java's iterator which is treated as an interface, an iterator in C++ is a class that represents the current element of a collection, and lets user steps from one element to the next.