5 Assignment P3 [Assignment ID: templates]

5.1 Preamble (Please Read Carefully)

Before starting work on this assignment, it is **critically important** that you **carefully** read Section 1 (titled "General Information") which starts on page 1 of this document.

5.2 Topics Covered

This assignment covers material primarily related to the following: templates, template class, template function.

5.3 Problems

1. Vector3 TEMPLATE CLASS. In this problem, we revisit the DoubleVector3 class developed in an earlier assignment problem. You should use the code that you wrote for the DoubleVector3 class as a starting point for this problem.
   (a) Implement a template class called Vector3 that represents a three-dimensional vector. The Vector3 class should take a single parameter corresponding to the type of the elements in the vector (i.e., the coordinate type). The Vector3 class should have the same interface as the DoubleVector3 class. That is, the Vector3 class should provide all of the same public member functions and supporting non-member functions as the DoubleVector3 class, as identified in the file DoubleVector3.hpp from the earlier assignment problem. Also, the parameter and return types of corresponding functions in the code for Vector3<T> and DoubleVector3 should match exactly in the case that T is chosen as **double**. The Vector3 class should work for coordinate types of **double** and **int** (i.e., Vector3<**double**> and Vector3<**int**>). All of the code for the Vector3 template class should be placed in the file Vector3.hpp.
   (b) To confirm that the Vector3 template class works correctly, write a test program called testVector3 to test your code. For convenience, you may want to make use of the assert macro (defined in the header file cassert) in much of your test code. A simple example is provided in the file testVector3.cpp, which is available from the course web site. Write the code that does the actual testing of your Vector3 template class as a template function called testVector3. The testVector3 template should take a single type as a parameter, which corresponds to the parameter for the Vector3 class to be tested. Thus, calling testVector3<T> will test Vector3<T>. In the function main, call testVector3<**int**> and testVector3<**double**> to test the Vector3 class. **Be thorough in your testing**, as your Vector3 class must also work for the instructor's test code, which is not the same as yours.