Lab 3

Templates

In this workshop, you design and code a couple of class templates and test it on two different instantiations.

**LEARNING OUTCOMES**

Upon successful completion of this workshop, you will have demonstrated the abilities to

* design and code a class template
* demonstrate the use of class templates using different data types

**SPECIFICATIONS**

The source files for this workshop include:

* **List.h** - defines a class template for a list of elements of any data type
* **Pair.h** – defines a class template to hold a key-value pair of data

List Template

Design and code a class template named **List** for managing an array of any datatype.  The class generated by your template contains an array of dimension **N** of type **T** where **T** and **N** are template parameters, which the programmer who uses your template can specify.  Save your template in a header file named **List.h**.

Your design includes the following member functions:

* **List()** - default constructor - adopts a safe empty state
* **size\_t size() const** - returns the *number of entires* in the array
* **const T& operator[](int index) const** – overloaded operator that returns the requested array index
* **void operator+=(const T&)** – overloaded operator that adds an element to the array if space is available.

Pair Template

Design and code a class template named **Pair** for managing a key-value pair off template data types **A** and **B**.

Your design includes the following member functions:

* **Pair()** – default constructor – adopts a safe empty state
* **Pair(const A&, const B&)** – overloaded constructor that sets the key-value pair elements of the class
* **const A& getKey() const** – returns a read-only version of the stored key information
* **const B& getValue() const** – returns a read-only version of the stored value information

Results

The main program will use your List and Pair template objects to create two types of lists using different datatypes. At the end of the program, the two lists will be written to an output file and should look something like the following:

Price List

-----------

Apple : 1.23

Black Tea : 5.29

Juice : 3.45

Entries

-------

CPU : Central Processing Unit

NVIDIA : Gaming Video Card