ECE 2420 Programming Exercise #8

(Associative Arrays: Binary Search Trees)

# Overview

In the last PEX7, you implemented a parts lookup database. Your database has been wildly successful. Your company has added thousands of parts each with hundreds of parameters. You have now become a victim of your own success. The database has grown so much and used so frequently that the performance is lacking. You need to make the lookup times significantly faster in order for the system to be usable. In this programming exercise the coder will use the same associative array interface from PEX7 and re-implement the underlying implementation using binary search trees.

# Parts Catalog Architecture

The parts catalog is an associative array of associative arrays. The outermost array is keyed by part name, (a string in this case). The content of the outer associative array is the inner associative array. The inner associative array is keyed by a parameter name and the content is the parameter value, (both strings).

When iterating through the structure it should be possible to print all part names in order and then print each parameter and associated value for that part in order of parameter name. It should also be possible to extract any given parameter value by using the get methods.

# Programming Concepts

This exercise covers many programming concepts including inheritance, deep copy, constructor types including default and delete, operator overloading, pass by ref and by val, return by ref, stack vs heap, binary search trees, templates, reference counting pointers, lambdas, closures, functors, and abstract base classes.

# System Requirements

The design must use the provided interface header verbatim. This will allow automated testing of the design you produce. See grading rubric for specific system requirements and associated grade values.

# Turn-in Procedures

Turn in all source code via a git push to your 2420 repository by 11:59p.m. on <DUE DATE>

Grading Rubric

(ECE 2420 PEX8)

|  |  |  |
| --- | --- | --- |
| Requirement / Criteria | Available Points | Student’s Score |
| Uses base class interface verbatim | 10 |  |
| Can be created on the stack | 10 |  |
| Is copy constructible and assignable | 10 |  |
| Correctly deep copies in insert | 10 |  |
| Modification after “get” modifies content in structure | 10 |  |
| Modification in “forEach” modifies content in structure | 10 |  |
| Method “forEach” iterates in order | 10 |  |
| Inserting functions correctly | 10 |  |
| Deleting functions correctly | 20 |  |
| “getting” and “deleting” non-existent key functions correctly | 10 |  |
| System is free of memory leaks | 10 |  |
| Total | **120** |  |