CS3100/5100: Data Structures and Algorithms

Programming Assignment #3

1 Project Description

For this assignment, you will write a simple database for storing and retrieving employee records using a Binary Search Tree. You should begin by implementing class Employee in Employee.h, and Employee.cpp, which will contain information about a single employee (see below). Then you will implement a Binary Search Tree in which each Binary Tree Node stores an employee record with the EmployeeID as the key. The Binary Search Tree is implemented in BinaryTreeNode.h, BinarySearchTree.h, and BinarySearchTree.cpp.

An Employee record inside an Employee database file should support the following fields: Last Name (string), First Name (string) and Employee ID (integer). I will provide an input database file that contains lines with three items: Last Name, First Name and Employee ID. Last Name, First Name and Employee ID are separated by spaces. The valid Employee ID is a number from 0 - 9999999. Duplicate Employee IDs are not allowed in the database file.

In the beginning of your main() function, you should open the input database file, create a Binary Search Tree. Then you should read a specified number of Employee records that contain last name, first name, and EmployeeID from the input database file, and insert these Employee records into the Binary Search Tree one by one using the Binary Search Tree's insertion method (The Employee ID will be used as the key for Binary Search Tree).

Then you should provide a USER INTERFACE (MENU) that supports the following operations in the main() function:

- Insert new record: prompt the user for all fields, create an employee record and insert it into the Binary Search Tree.
- Delete Record: Ask the user for an Employee ID and delete it from the Binary Search Tree.
- Search on Employee ID: Print all data to the screen for an Employee whose Employee ID is given via the keyboard.
- Save the employee records in the binary search tree to disk using inorder tree traversal. If you also save the employee records in the binary search tree to disk using preorder tree traversal or postorder tree traversal, you will get 20 points of bonus points (10 points per tree travesal method)
- Quit.

Running your program should produce a menu similar to the one shown in the example below. When loading a database from the disk, all current records should be deleted, and the database should be loaded from a file, and the Binary Search Tree should be rebuilt.

The search operation should print the Employee objects found. For example, an Employee ID query for 662312 would return results similar to the following:
MENU
(I)nsert new record

(D)elete record
(E)mployee ID search
(S)ave database to a file
(Q)uit

Enter choice: E

Enter Employee ID: 662312
Searching...

20 Employees searched. Found 1 record:

Last: Powers
First: Susan
EID: 662312

2 Bonus Task

If your binary search tree class is implemented as a template class, that can not only support employee record (i.e., the object stored in the Binary tree node is an employee record), but also can support integers, doubles, strings, or other object types (i.e., the Binary tree node stores an integer, a double, a string, or an object of other object type), then you can get bonus points (20 points). You should test the template classes in your main.cpp file.

3 Requirements

1. In order to use the c++ compiler environment installed under the school's unix server, unixapps1.wright.edu, you need to connect to this unix server remotely using a secure shell client, putty. You can remotely connect to this unix server, unixapps1.wright.edu, on campus from a Wright State computer or use your own laptop connecting to the WSU wifi network named WSU-Secure. Note that you cannot remotely connect to this computer using a secure shell client using computers outside Wright State University without installing VPN or use the campus WSU_EZ_CONNECT wifi network.

2. You must submit an ELECTRONIC COPY of your source program through Pilot before the due date. If for some reason Pilot is unavailable, submit your source code to the instructor Meilin Liu.

3. Your main program should create a user interface similar to the example above. The file name for the main program should be lab3.cpp.

4. Submit all your source codes (Employee.h, Employee.cpp, BinaryTreeNode.h, BinarySearchTree.h, BinarySearchTree.cpp, and lab3.cpp), makefile, possibly a README file, and any other required files. You are recommended to explain your programs clearly in the README file.

5. All the submitted project files should have: Course Number / Course Title, Your Name, Prof.s Name, Date, and the Project Name. If you did not include these required contents in your submitted files, then 5 points will be deducted. You also need to submit a makefile or a compiling command to compile your source codes. If not, another 5 points will be deducted.

6. The instructor will test your programs under WSU's UNIX environment, e.g.,unixapps1.wright.edu. It is YOUR responsibility to make your programs workable and runnable by others under school's UNIX environment.

7. The programming assignment is individual. You must do the project by yourself. If you allow others to copy your programs or answers, you will get the same punishment as those who copy yours.