

**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 `EmployeeID` 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 `EmployeeID` and delete it from the Binary Search Tree.
- Search on `EmployeeID`: Print all data to the screen for an `Employee` whose `EmployeeID` is given via the keyboard.
- Save the employee records in the binary search tree to disk using inorder tree traversal.
- 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 `EmployeeID` 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



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.