Oxy

Programmer Manual
Oxy

1. Problem Description

The Oxy class inherits from the tree class implemented in tree.t and defined in tree.h. It consists of functions which allow for employees, as implemented in employee.cpp and defined in employee.h, to be input into a binary search tree from a file and then displayed and manipulated by the user. The user can get employee data from a file, insert a new employee, update a current employee, display a current employee, list all of the records or a subset of the records by employee code, delete a current employee, and save any changes to a file which can then be read in by the program.

2. Class Oxy

Private data members:

bool populated	flag determining whether the tree has data or not

Private data functions:

printInOrder	prints tree using an inorder traversal
preOrderWrite	writes tree to a file using a preorder traversal
deleteTree	deletes the tree from the root using a postorder traversal
isValidSSN	ensure a valid social security number
validateName	ensure a valid filename input
validateInt	ensure a valid integer input
validateDouble	ensure a valid double input

Public data functions:

Oxy	constructor for an Oxy object
populate	get employee data from a file and put it into the tree
insertEmployee	insert a new employee into the tree
update	changes data for a current employee
retrieveEmployee	displays data for a current employee
writeToFile	writes tree data into a file
deleteEmployee	removes a current employee from the tree

3. High Level Program Solution

Oxy
    set populated to false

populate
    get input file from the user
    if the tree is populated, delete the tree
    read input file and insert employees into the tree
    set populated to true

insertEmployee
    get the information for each field for the employee data from the user
    insert the employee into the tree

update
    get last name of employee to update
    search tree for selected employee
    if the employee is not found, return
    display menu to allow user to select which field to update
    when the user finishes updating, delete the old employee
    insert the employee with the updated data

retrieveEmployee
    get last name of employee to update
    search tree for selected employee
    if the employee is not found, return
    calculate the pay of the employee
    display the selected employee as well as the calculated pay of the employee

writeToFile
    get the file name to store the data in from the user
    call helper function preOrderWrite to write to the file

deleteEmployee
    get last name of employee to update
    search tree for selected employee
    if the employee is not found, return
    delete selected employee from the tree

printInOrder
    if ALL is selected, display all employees
    if OFF is selected, display employees with a code of 'O'
    if FAC is selected, display employees with a code of 'F'
    if SAL is selected, display employees with a code of 'S'
    otherwise, the input is invalid

preOrderWrite
    write the left side of the tree data to an output file
    write the root of the tree data to an output file
    write the right side of the tree data to an output file

deleteTree
    delete left side of the tree
    delete right side of the tree
    delete the root

isValidSSN
    if the input length is not 11, return false
    if the first three characters, the middle two characters, or the last 4 characters are not
        integers, set hasDigit flag to false
    if the fourth or seventh characters are not hyphens, set hasHyphen flag to false
    if hasDigit flag and hasHyphen flag are both true, return true
    otherwise return false

validateName

if the input is larger than the buffer, reprompt until correct input

validateInt

get input from user

if the input is not an integer, or is less than a lower bound, reprompt until correct input

validateDouble

get input from user

if the input is not a double, or is less than a lower bound, reprompt until correct input