Programmer Manual

Programmer Manual
Football Frenzy

1. Problem Description

This program keeps track of NFL standings and allows editing and retrieval of previous standings by the user. A doubly linked list is used to store all of the data for the teams. The user can add, delete, and edit teams, as well as simulate games. The standings are kept sorted after any user operations first by division, then by win percentage, and finally alphabetically by team name. If the user chooses, the updated standings can be saved into a file which can be read back into this program.

1.1 On Using a Doubly Linked List

The backing storage for all of the teams is the stl::list. This is a doubly linked, dynamically allocated list which allows for efficient traversal and sorting. The doubly linked feature occurs and is used primarily in the sorting functionality, where the list is traversed both forwards and backwards.

Backwards traversal in a singly linked list is much more inefficient, due to being forced to go all the way to the end of the list to get back to the beginning and start over. The doubly linked list allows for the standings to be traversed both forwards and backwards efficiently, which allows for more efficient searches and sorts.

Using an array or a vector for this project has a few pros and cons. The pros are that arrays and vectors are random element access, allowing for constant time retrieval of data. They are much worse than linked lists though at inserting in the front or the middle of the data, due to each element needing to be shifted over. The linked list allows insertion at any point in the list efficiently.

2. Data Types and Classes

The data types used in this program fall into two categories: predefined data types and programmer-defined data types. The following subsections address the data types used.

2.1 int (predefined type)

Variables:

input	user input for the main menu
buffer	the maximum size of a name
currentDiv	the current division to be printed
wins	number of a teams wins
losses	number of a teams losses
ties	number of a teams ties
divNum	division a team is in

2.2 char (predefined type)

Variables:

name[]	team name
confName[]	conference name
divName[]	division name

2.3 string (predefined type)

Variables:

inFileName	name of the input file
outFileName	name of the output file

dateStr — date to be written to the output file

2.4 ifstream (predefined type)

Variables:
inFile — input file

2.5 ofstream (predefined type)

Variables:
outFile — output file

2.6 double (predefined type)

Variables:
winPercentage — the win percentage of a team

2.7 Standings (programmer-defined type)

This class has:

Data members:
list<team> li
bool isEmpty

Member functions:
Standings
print
init
addTeam
deleteTeam
playGame
output
editTeam
initDivision
isConfGame
isDivGame
updateWinPercent
standingsSort
deleteList
validatedInput

See the programmer manual for the Standings class for more details

2.8 Record (programmer-defined type)

This struct has:

Data members:
int wins
int losses
int ties

2.9 Team (programmer-defined type)

This struct has:

Data members: char [] name
Record total
Record home
Record away
Record division
Record conference
int divNum
double winPercent
char [] confName
char [] divName

Member functions: operator overload <

3. High Level Program Solution

Main Program
Print the title
Get the file with the initial data holding the NFL standings
Create the Standings object and populate the linked list with the data from the input file
Print the menu
Prompt the user for input

Call member classes from the Standings class for the input option:

1. Print the menu
2. Add a team
3. Delete a team
4. Play a game
5. Edit a team
6. Output results to a file
7. Read in a different file
8. Print the title again
9. Exit the program

function makeSelection()

The primary functionality of the program. Prompts the user for input and handles all appropriate function calls from the menu

function printMenu()

Print the menu options along with the corresponding integer selections, along with a prompt for the user to input their option

function printTitle()

Print the title screen art

4. Limitations and Suggestions

This program is limited by the user keeping track of the NFL standings in reality. If they input incorrect data, the standings will then be incorrect. The teams are also sorted a specific way. The program could be extended to allow the user to choose to sort the teams by a different parameter, such as number of total wins, or have the teams not sorted by conference and division in order to see the standings of the entire NFL in a sorted manner.