

CSCE 2014 – Programming Project Report

Student Name: Centarrius Brooks

Submission Date: 9/12/24

Academic Integrity Statement: I pledge that I have neither given nor received unauthorized help on this programming assignment.

Problem Statement:

The program reads from a file named top100.txt, which contains information about the top 100 college basketball recruits. The inputs of the program were the specific position, state, & grade of said recruit. The outputs were the data of the recruits that matched those specifics in the order of FirstName, LastName, Position, City, State, HighSchool, Height, Weight, Stars, Grade, and School. This output data was well formatted, with each recruit separated by a blank line.

Design:

When it comes to design, I decided to call the RecruitDB methods in the main program with “hard coded” parameters instead of a main menu. Even though a menu would be antically more pleasing, it would be more unnecessary work. I used the vector data structure to store a dynamic list of Recruit objects in the RecruitDB class. I use linear search algorithms for search methods such as “searchByState” etc. These methods filter recruits based on criteria like state, position, and grade. File input/output algorithms were used through the read_txt and print_txt methods. These methods employed simple file reading and writing algorithms using ifstream and ofstream objects. The pros were that exception handling in read_txt ensures robust error management, improving the program's reliability. The cons are that vectors can only handle a small amount of data, same for Linear Search which may not be ideal as the dataset grows.

Implementation:

Instead of extending my original code from project 1. I decided to extend Dr Gauch's code as his solution was perfect and there would be no worries about issues later. My code for project 1 did work, however it just wasn't nearly as good as it should have been. The error checking and whitespace issues were rampant. I first defined the Recruit class with basic member variables and methods for handling recruit data. Implemented and tested the read_txt and print_txt methods in the Recruit class for file input/output operations. Developed the RecruitDB class to manage a vector of recruits and added methods to read from and write to files. Implemented search methods in RecruitDB to filter recruits based on different criteria. Debugged the code to handle invalid inputs and ensure robust error handling. The last step was testing each search method one by one and debugging as needed. My developmental timeline was a little over a week with about two days for each step.

Testing:

I tested my program a variety of ways. I used normal cases such as strings for search methods such as `searchByState` or `searchByPosition` but also with abnormal cases such as numbers in these search methods with string parameters and even exclamation points. I even tried “zebra” in the `searchByGrade` method but it didn’t compile since that method was defined with a `int` parameters. Everything worked as I expected. Pictures of this is below.

```
41
42
43 // Search for recruits by Grade
44 int grade = zebra;
45 db.searchByGrade(grade);
46
47 return 0;
48 }
49
50
```

input

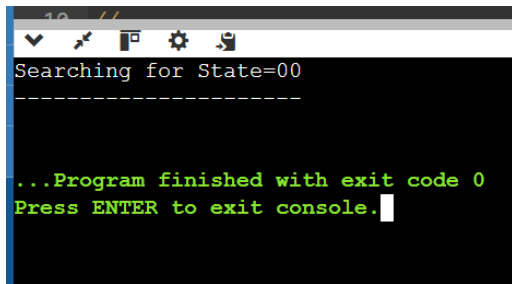
Compilation failed due to following error(s).

```
main.cpp: In function 'int main()':
main.cpp:44:17: error: 'zebra' was not declared in this scope
 44 |     int grade = zebra;
    |
```

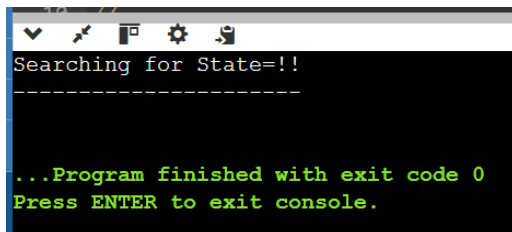
```
Searching for State=AR
-----
FirstName: (Isaiah)
LastName: (Sealy)
Position: (SF)
City: (Springdale)
State: (AR)
HighSchool: (Springdale High School)
Height: (6'6'')
Weight: (195)
Stars: (****)
Grade: (84)
School: (Uncommitted)

FirstName: (Terrion)
LastName: (Burgess)
Position: (PF)
City: (Marion)
State: (AR)
HighSchool: (Benton High School)
Height: (6'8'')
Weight: (195)
Stars: (****)
Grade: (82)
School: (Uncommitted)

...Program finished with exit code 0
Press ENTER to exit console.[]
```



```
10 //  
v ↗ □ ⚙ 📄  
Searching for State=00  
-----  
  
...Program finished with exit code 0  
Press ENTER to exit console.
```



```
11 //  
v ↗ □ ⚙ 📄  
Searching for State=!!  
-----  
  
...Program finished with exit code 0  
Press ENTER to exit console.
```

Conclusions:

This project was a success and was little more difficult than I expected but I made it. I would go to office hours next time to get help from Dr. Gauch directly as he could have steered me in the right direction sooner. This project took about 8 days to complete. I am excited for the next project - linked lists.