

CCT College Dublin Continuous Assessment

Programme Title:	<i>H.Dip. in Computing</i>		
Delivery Mode:	FT		
Cohort Details:	Feb 2024 cohort		
Module Title(s):	Algorithms & Constructs Project Skills & Professionalism Software Development Fundamentals		
Assignment Type:	<i>Integrated / Individual</i>	Weighting(s):	Algorithms & Constructs – 60% Project Skills & Professionalism – 60 % Software Development Fundamentals – 55%
Assignment Title:	<i>System Modelling & Build</i>		
Lecturer(s):	<i>Ken Healy / Kayoum Khbuli / James Garza</i>		
Issue Date:	<i>15th Oct 2024</i>		
Submission Deadline Date:	<i>Saturday 16th Nov at 11.59pm</i>		

Student name: Jhanett Nicolas Apaza

Student code: 2024132

Contend

Contend.....	2
Introduction.....	3
Challenges and Operations.....	3
Using Enums for the Menu.....	3
Sorting a List of Names.....	3
Searching for a Name.....	4
Adding New User Input.....	4
Generating Random People.....	4
Conclusion.....	4

Introduction

For my project, I developed a terminal-based program that manages a list of employees or applicants for a company. This program helps with sorting, searching, adding new names, and generating random data. In this essay, I will explain the challenges, core functions, and how the program is organized.

Challenges and Operations

The goal of my program is to make it easier to find information about employees, like their Manager type and Department. Some key challenges include:

- ☐ **Efficient Searching:** Quickly finding a person's details in a large list.
- ☐ **Sorting Data:** Keeping names in order to make searching faster.
- ☐ **User Input Validation:** Making sure users enter correct data.
- ☐ **Scalability:** Handling more data as the company grows.

Using Enums for the Menu

To make the program easy to use, I implemented a menu system using Enums. Enums are helpful because they create a list of fixed options that the user can choose from. In my program, the menu has four options:

- ★ Sort the List
- ★ Search for a Name
- ★ Add a New Entry
- ★ Generate Random People

Using Enums makes the code more organized and less error-prone. It helps ensure that only valid options are available, improving the user experience.

Sorting a List of Names

My program reads a list of names from a file called Applicants_Form.txt and sorts them in alphabetical order. Quick Sort works by selecting a “pivot” and arranging the list so that all names before the pivot are alphabetically smaller and those after are larger.

I chose Quick Sort because it has an average time complexity of $O(n \log n)$ which is faster than other algorithms like Bubble Sort or Insertion Sort. These other algorithms are slower, especially when the list is long, making them less suitable for this task.

The program shows only the first 20 sorted names to keep the output easy to read.

Searching for a Name

The program can also search for a person's name and return details like their Manager type and Department. Binary Search algorithm, which is very efficient for sorted lists. It works by dividing the list into halves and checking if the name is in the left or right half. This method is faster than Linear Search, especially for large lists, because it only takes $O(\log n)$ time.

I chose Binary Search over Linear Search because it is much quicker when the list is sorted. Linear Search would have to check each name one by one, making it too slow for larger datasets.

Adding New User Input

The program allows users to add new names, select a Manager type, and choose a Department. It is interactive and asks users for their input, ensuring that only valid Manager types and Departments are chosen. For example, if the user enters a name, they must also select a valid Manager and Department from a predefined list. This helps keep the data consistent and error-free.

Generating Random People

To test the program, I added a feature to generate random names with assigned Manager types and Departments. This is useful for filling the list with sample data. It also helps test the program's ability to handle larger datasets, and ensures that random data fits within the existing categories for Managers and Departments, avoiding conflicts.

Conclusion

Overall, this project helped me understand how to use algorithms and data structures to solve real-world problems in managing information. It also showed the importance of making software user-friendly and efficient.

The code can be found here -> https://github.com/JhanettSis/CA_2/tree/main/src/ca_2