



ENSC 251

Final Project

Sahaj Singh, Nicholas Mah, Zach Hsu, Daniel Yu



Design Process

Agile Methodology

01

Fundamentals

Information from lectures
and other resources

02

Implementation

Creating programs in
code

03

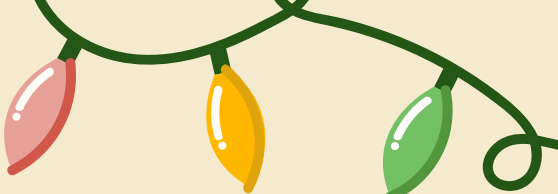

Testing

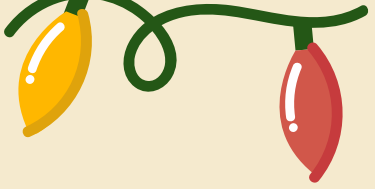
Rigorous testing and
debugging

04

Improvement

Editing functions to
improve efficiency and
reduce bugs





Design Process

- Implemented various helper functions



```
1 int Get_Number() // checks validity of input
2 {
3     string line;
4     int user_input;
5     while (getline(cin, line))
6     {
7         stringstream ss(line);
8         if (ss >> user_input)
9         {
10             if (ss.eof())
11             {
12                 break;
13             }
14         }
15         cout << "Please enter a number as input only for the previous input: " << '\n'
16              << ">> ";
17     }
18     return user_input;
19 };
```

User Interface



Terminal

Console based User Interface with robust error checking



Characters

Used spacers such as ">>" and "-" to create space between menu options and outputs

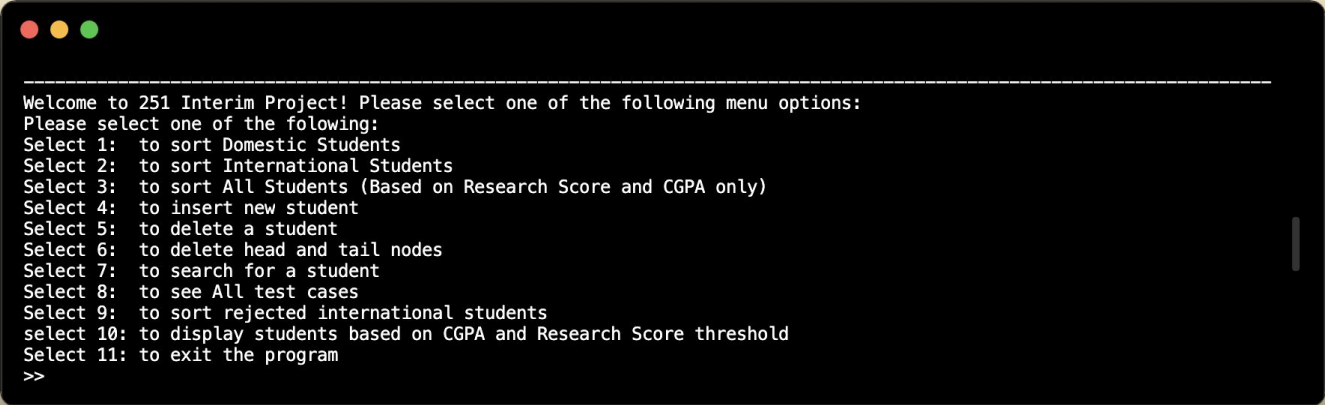


Submenus


A main menu simple and easy to understand




User Interface



```
-----  
Welcome to 251 Interim Project! Please select one of the following menu options:  
Please select one of the following:  
Select 1:  to sort Domestic Students  
Select 2:  to sort International Students  
Select 3:  to sort All Students (Based on Research Score and CGPA only)  
Select 4:  to insert new student  
Select 5:  to delete a student  
Select 6:  to delete head and tail nodes  
Select 7:  to search for a student  
Select 8:  to see All test cases  
Select 9:  to sort rejected international students  
select 10: to display students based on CGPA and Research Score threshold  
Select 11: to exit the program  
>>
```



```
-----  
Please enter what you would like to sort by:  
Please select one of the following:  
First Name: F, Last Name: L, CGPA: G, Research Score: R, Province: P  
>>
```





Core Classes

Domestic/International/Student

- students' information based on classification wanted

DomesticStudentList/InternationalStudentList/StudentList

- create and append lists based on classification wanted





Core Functions

Stu_Sort

- Merge sort functions responsible for creating all lists based on respective hierarchies

Get_Number

- Ensuring validity of inputs whenever needed



Find Functions

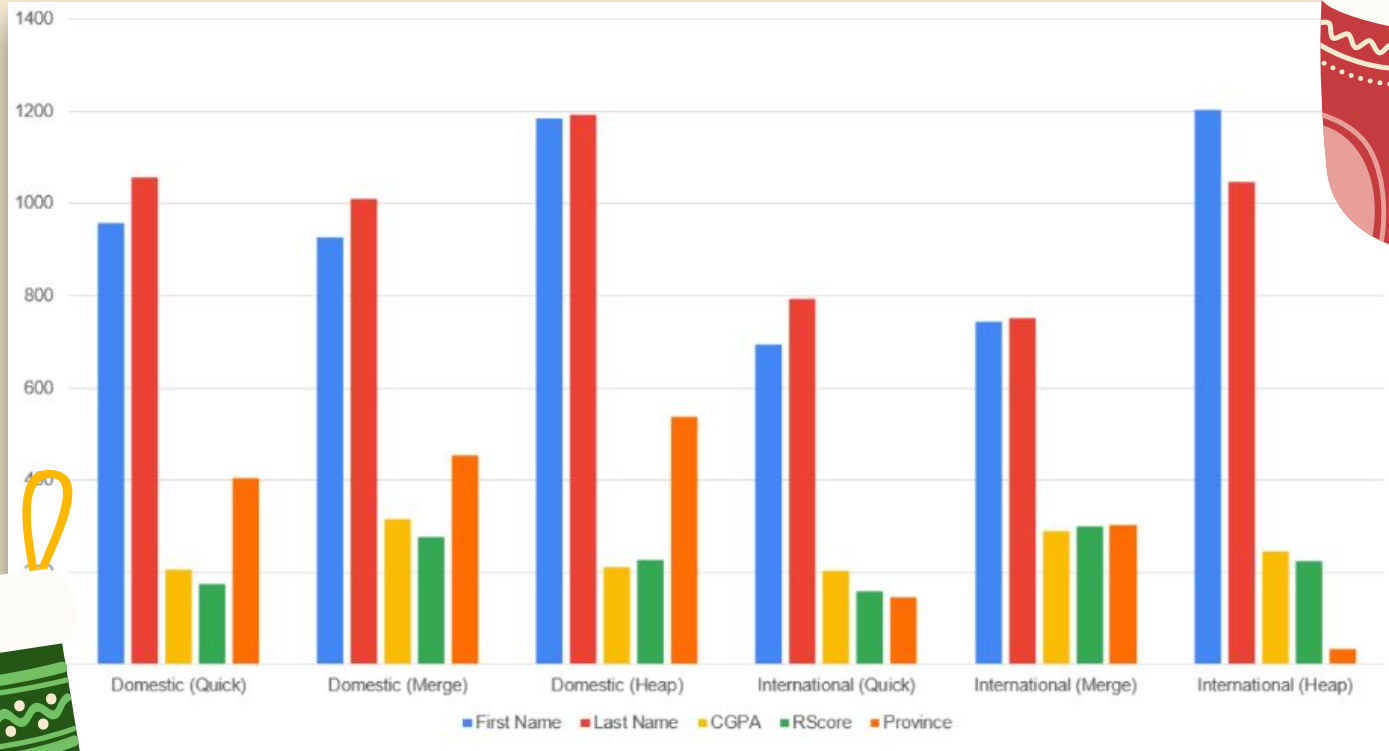
- Traverses linked lists based on specific classification
- 

Our Innovations

- Re-structured code to use interim material .
Read data into array
→ Convert Array to Linked List
→ Delete Array and Maintain linked lists
- Additional option to display and sort:
 - merged student list
 - rejected international students
- Display student type in merged student list rather than displaying excess information
- All lists are maintained despite merging
 - Can even look up students using threshold in all lists
- Showcase head and tail nodes to user



Average Sorting Times



Project Management

Meetings

Met frequently both in person and online

VSCode

IDE of choice, used the Github extension with Live Share

Github

Utilised a private Github repository to keep files organized

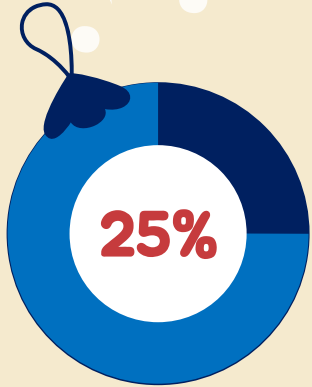
Discord

Primary form of communication

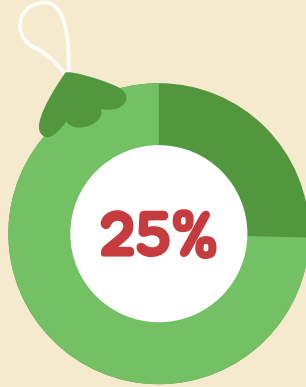


Work Distribution

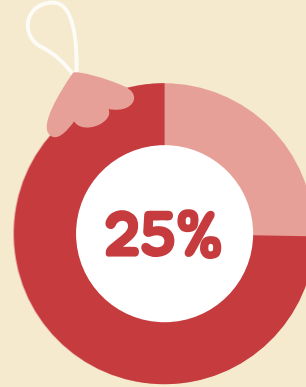
Equal Work Distribution amongst all members



Zach



Daniel



Sahaj



Nick

Key Takeaways



Github

Used Github in a professional manner as a team



Linked Lists

Learned about data structures such as linked lists



Time Management

Planning and exercising conscious control of time



Polymorphism

Exercised modularization throughout the program



Communication

Teamwork between members



OOP

Gained in depth understanding in regards with object-oriented programming



Thanks!



Does anyone have any questions?

