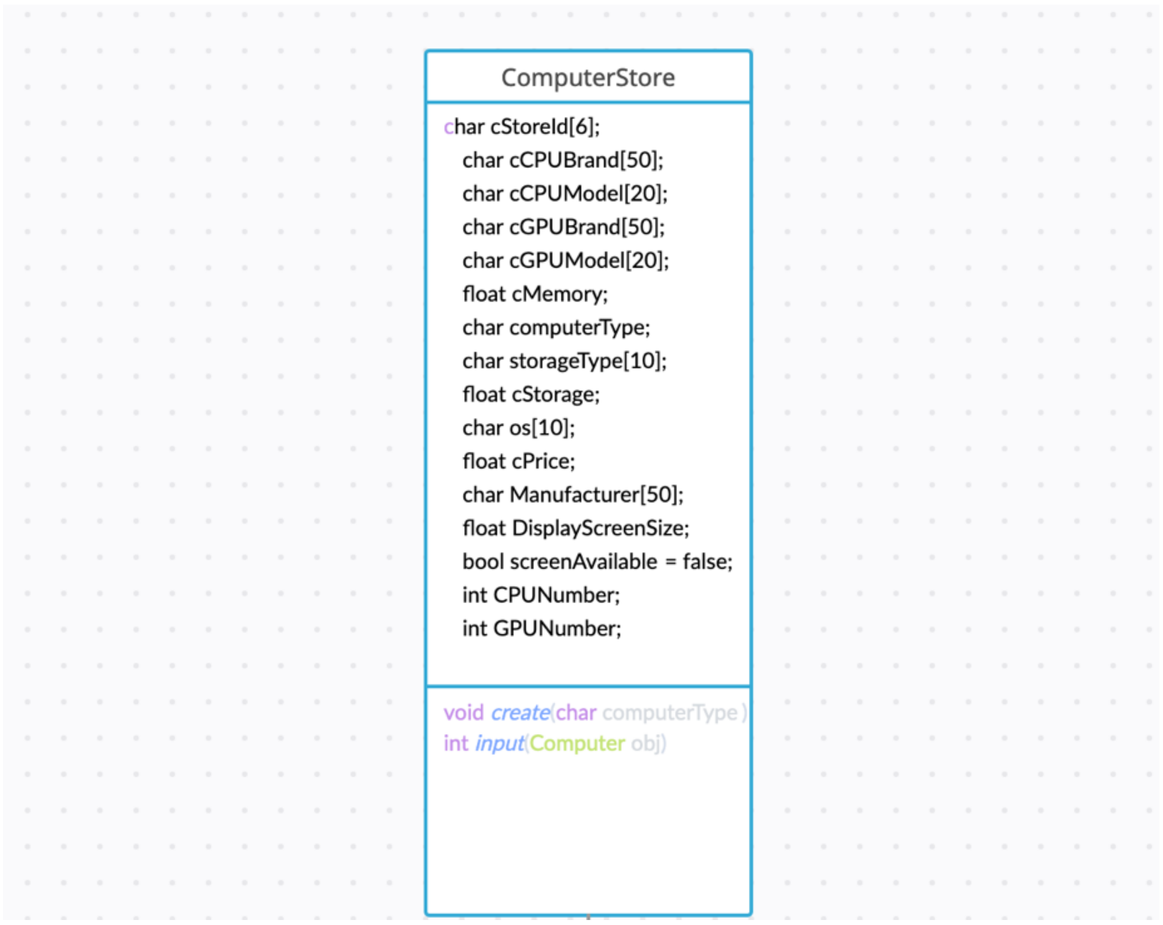
ECS793P Coursework Report

Name: **Franklin Antony**

Student ID: **200626510**

Hierarchy design

We use a class **ComputerStore** and it contains all the details of the store. It basically acts like a centralised inventory system.



Approach to programming

- The computerStore acts the main class. Whole flow is through that class.
- When we run the main() function we fetch from files and store it as an array of objects.
- We have Add, Remove, Search and Display functions in the main menu
- ***Add Function***:
  - The user can add the record of Laptop, Server or Desktop entries based on their choices.
  - The data is written directly into the file. This also comes in handy to save into file on exit.
- ***Remove Function:***
  - This function mainly searches for ID and removes the record

- ***Search Function****:*
  We can search based on
  - ID
  - Type
  - Price
  - Storage
  - Screen size
- ***Display Function:***
  - This function mainly displays all the records

Issues faced and limitations

- The inputs sometimes have issues with newline and cin.ignore() was used to avoid it in many places.
- In some cases, the choices where causing the infinite loop and method of checking the value if it is empty or not, was used to avoid it in many places.

Code Explanation

Functions used in this project

1. main() – main function
2. create() - for creating computer object
3. inputIntoFile () – for writing into the file
4. clearFile() – to clear file before writing into the file
5. returnComputerType() – to return the computer type
6. printrecord() – prints the object
7. printAllComputers() – displays all the computers in the file
8. fileToMemory() – this function reads the array of objects from file into the memory
9. SetDisplayScreenSize()- to set the screen size
10. addComputer() - adding computer by calling create. Called from main function.
11. removeComputer() – to remove the computer from the list
12. searchComputer() – to search something in the computer
13. mainMenu() – menu to decide the flow control of the program

Testing

Here are some random tests.

Function	Inputs	Expected behaviour	Observed behaviour
Main Menu	1	Add Item	Goes to Add item
Main Menu	a	Error input	Sometimes causes error/infinite loop
Add Menu Memory(integer)	12	Saves and moved to next input	Saves and moved to next input
Add Menu Memory(integer)	a	Shows error and saves 0	Infinite loop
Add Menu Storage(integer)	asd	Shows error and saves 0	Infinite loop
Remove Menu ID (string)	asds	Checks and removes if present	Checks and removes if present
Search Menu ID (string)	test	searches if ID present	searches if present
Search Menu Price (double)	1200	searches if price range is present	searches if price range is present
Search Menu Price (double)	a	searches if price range is present	Crashes/infinite loop
Search Menu Storage (int)	12	searches if storage range is present	searches if storage range is present
Search Menu Storage (int)	😊(emoji)	searches if storage range is present	Crashes/infinite loop

Main Menu Display menu	(No choice)	Displays all records	Displays all records
Main Menu (Want to continue?)	Y	Shows the Menu again	Shows the Menu again
Main Menu (Want to continue?)	12	Shows the error	Shows the error
Main Menu (Want to continue?)	😊(emoji)	Shows the error	App crashes