

## Membership Manager Report

The member manager program allows the user to add members to the society list, view current long-standing members in order of the length of their membership, and revoke the membership of those who are leaving or have left the society. It does this using a simple Member class to store all details of a member, two instances of a MemberList class to act as lists of current members and past members, and a MemberManager class to act as the controller calling all of the appropriate functions according to the user's input.

### Member Class

The member class stores the member's names, their membership number, whether they are a current member or not, and their join date as an int array. Other than the usual getters, setters and display functions the class has two overloaded functions, > and <=, which allow the comparison of two members when sorting. These functions first compare the members' join dates, followed by their last names if these are identical, followed by their first names if they're also identical, to see if one member is > or <= to the other. Dates are considered > each other if they are more recent in time, and names if they are later in the alphabet.

### MemberList Class

The MemberList class contains a dynamic array to contain up to 10'000 members, the number of members currently on the list, and a string to store the type of members the list is storing. Other than gets, sets and display functions, the class contains some list functions to add and remove members to and from the list (remove was included to allow current members to be removed and placed in the past members list when they leave the society), and check if it's empty. It also contains functions (some private to the class only) to allow the members to be sorted using a quicksort algorithm.

### MemberManager

The MemberManager class acts as a controller to hold the lists of members (current and past) and perform the basic functions offered to the user by calling the functions of the other classes. It contains a method corresponding to each user option, i.e. to add a member, to view long-standing members, and to revoke membership from a member that is leaving. It also has user options and therefore functions to display either of the current or past member lists to make testing and understanding of the lists' contents easier. These user option methods have helper methods to perform checks and get information, e.g. getting member information from the user and/or checking that their input is valid, one of which is a linear search function which allows the user to search for a member in the list to revoke membership, or check they're not already in the list. There are also some display functions and one to initialise some test data.

### main()

The main function simply creates a MemberManager for the session, calls its' menu display methods, and controls the flow through the menu and its' options. It does this by getting the user's validated menu choice and entering a switch statement to call the corresponding MemberManager's function, on a loop, till the user is done.

## Testing

I employed a testing strategy that involved going through and testing the different paths and cases for each main waiting list menu option, with particular focus on boundary cases, as they are the most likely to experience errors and failures of logic (e.g. attempting to work outside the boundaries of an array). I did this because the 5 main option methods call and make use of the other methods, and so testing these methods tests the other methods. For example, testing the option to add a member also involves testing the `binarySearch()` method for making sure the member is not already in the list in the process, since it is called and used as part of that add member process. The test results are detailed below; all tests yielded the expected result.

### **main() / Menu Navigation**

Input: [Run program]

Result: [Welcome displayed followed by the menu:]

1. *Add a member*
  2. *View long-standing members*
  3. *Revoke membership from a member*
  4. *View list of current members*
  5. *View list of past members*
- Please enter the number of your choice or 0 to exit:*

Input: 1

Result: [Enters and goes through the process for option 1]  
[Menu is displayed with a request for the user's choice again]

Input: 2

Result: [Enters and goes through the for option 2]  
[Menu is displayed with a request for the user's choice again]

Input: 3

Result: [Enters and goes through the process for option 3]  
[Menu is displayed with a request for the user's choice again]

Input: 4

Result: [Enters and goes through the process for option 4]  
[Menu is displayed with a request for the user's choice again]

Input 5 process

Result: [Enters and goes through the process for option 5]  
[Menu is displayed with a request for the user's choice again]

Input: 0

Result: [Program ends and exits.]

### **main() / Menu - Invalid Input**

Input: [Run program]

Result: [Welcome displayed followed by the menu:]

1. Add a member
  2. View long-standing members
  3. Revoke membership from a member
  4. View list of current members
  5. View list of past members
- Please enter the number of your choice or 0 to exit:

Input: 6

Result: That's not a valid choice, please enter 1-5, or 0 to exit:

Input: g

Result: That's not a valid choice, please enter 1-5 or 0 to exit:

Input: -1

Result: That's not a valid choice, please enter 1-5, or 0 to exit:

Input: 1.5

Result: That's not a valid choice, please enter 1-5, or 0 to exit:

#### **Option 1 – Add a member to the list**

##### **Valid Input – Current Member**

Input: [Select the add member option from the menu.]

Result: Please enter the member's details below.  
First Name:

Input: a

Result: Last Name:

Input: a

Result: Please enter the date the member joined as individual integers:  
Day (1 - 31):

Input: 1

Result: Month (1 - 12):

Input: 1

Result: Year in full (e.g. 1990):

Input: 1200

Result: Membership Number:

Input: 1

Result: Are they a current member?  
1. Yes.  
2. No.  
Enter number of choice:

Input: 1

Result: The member was successfully added to the list of current members.  
[Inspection of list reveals they were successfully added to the correct list.]

### Valid Input – Past Member

Input: [Select the add member option from the menu.]  
Result: *Please enter the member's details below.*  
*First Name:*  
Input: *b*  
Result: *Last Name:*  
Input: *b*  
Result: *Please enter the date the member joined as individual integers:*  
*Day (1 - 31):*  
Input: *2*  
Result: *Month (1 - 12):*  
Input: *2*  
Result: *Year in full (e.g. 1990):*  
Input: *1200*  
Result: *Membership Number:*  
Input: *2*  
Result: *Are they a current member?*  
*1. Yes.*  
*2. No.*  
*Enter number of choice:*  
Input: *1*  
Result: *The member was successfully added to the list of past members.*  
*[Inspection of list reveals they were successfully added to the correct list.]*

### Leap Year Date

Input: [Select the add member option from the menu and enter a and a for the names and 29/2/2016 (2016 was a leap year)]  
Result: *Membership Number:*  
*[Aka was accepted]*

### Invalid Input – Join Day

Input: [Select the add member option from the menu and enter a and a for the names.]  
Result: *Please enter the date the member joined as individual integers:*  
*Day (1 - 31):*  
Input: *0*  
Result: *That's not a valid day, please enter a number within the range 1 - 31:*  
Input: *32*  
Result: *That's not a valid day, please enter a number within the range 1 - 31:*  
Input: *abc*  
Result: *That's not a valid day, please enter a number within the range 1 - 31:*  
Input: *1.5*  
Result: *That's not a valid day, please enter a number within the range 1 - 31:*  
Input: *-1*

Result: *That's not a valid day, please enter a number within the range 1 - 31:*

Input: 20/03/2017 [or] 20-03-2017

Result: *That's not a valid day, please enter a number within the range 1 - 31:*

#### **Invalid Input – Join Month**

Input: [Select the add member option from the menu and enter a and a for the names and 1 for the day.]

Result: *Month (1 - 12):*

Input: 0

Result: *That's not a valid month, please enter a number within the range 1 - 12:*

Input: 13

Result: *That's not a valid month, please enter a number within the range 1 - 12:*

Input: abc

Result: *That's not a valid month, please enter a number within the range 1 - 12:*

Input: 1.5

Result: *That's not a valid month, please enter a number within the range 1 - 12:*

Input: -1

Result: *That's not a valid month, please enter a number within the range 1 - 12:*

Input: 20/03/2017 [or] 20-03-2017

Result: *That's not a valid month, please enter a number within the range 1 - 12:*

#### **Invalid Input – Join Year**

Input: [Select the add member option from the menu and enter a and a for the names and 1 for the day and month.]

Result: *Year in full (e.g. 1990):*

Input: 0

Result: *Sorry, that is not a valid year, please enter a full year from 1000 AD onwards:*

Input: 90

Result: *Sorry, that is not a valid year, please enter a full year from 1000 AD onwards:*

Input: abc

Result: *Sorry, that is not a valid year, please enter a full year from 1000 AD onwards:*

Input: 1.5

Result: *Sorry, that is not a valid year, please enter a full year from 1000 AD onwards:*

Input: -1

Result: *Sorry, that is not a valid year, please enter a full year from 1000 AD onwards:*

Input: 20/03/2017 [or] 20-03-2017

Result: *Sorry, that is not a valid year, please enter a full year from 1000 AD onwards:*

#### **Invalid Input – Whole Date**

Input: [Select the add member option from the menu and enter a and a for the names and 29/02/2017 (not a leap year)]

Result: *Sorry, it seems that's not a valid date according to the Gregorian calendar. Please try again.*

Input: [Select the add member option from the menu and enter a and a for the names and 31/06/1990 (June has 30 days)]

Result: *Sorry, it seems that's not a valid date according to the Gregorian calendar. Please try again.*

Input: [Select the add member option from the menu and enter a and a for the names and 30/2/2016 (2016 was a leap year)]

Result: *Sorry, it seems that's not a valid date according to the Gregorian calendar. Please try again.*

#### **Invalid Input – Membership Number**

Input: [Add a member with a membership number of 1, then select the add member option from the menu and enter a and a for the names and 1/1/1200 for the date]

Result: *Membership Number:*

Input: 1 [other member already has this number]

Result: *That's not valid membership number, please enter a whole number that is different than other current members:*

Input: 1.5

Result: *That's not valid membership number, please enter a whole number that is different than other current members:*

Input: abc

Result: *That's not valid membership number, please enter a whole number that is different than other current members:*

#### **Invalid Input – Membership Status**

Input: [Select the add member option from the menu and enter a and a for the names and 1/1/1200 for the date and 3 for membership number]

Result: *Are they a current member?*

1. Yes.

2. No.

*Enter number of choice:*

Input: 0

Result: *That's not a valid choice, please enter 1 for yes or 2 for no:*

Input: 1.5

Result: *That's not a valid choice, please enter 1 for yes or 2 for no:*

Input: abc

Result: *That's not a valid choice, please enter 1 for yes or 2 for no:*

Input: 3

Result: *That's not a valid choice, please enter 1 for yes or 2 for no:*

## Option 2 – View long-standing members

### Invalid Input

Input: [Test data list of current members initialised at the start:

```
// Order of params: firstName, lastName, memNum, day, month, year, currentMember?
Member mem1 = Member("Carrie", "Morris", 1000, 30, 06, 1990, true);
Member mem2 = Member("Kayleigh", "Stewart", 2000, 28, 02, 1992, true);
Member mem3 = Member("Garreth", "Tigwell", 3000, 03, 12, 1976, true);
Member mem4 = Member("Harry", "Potter", 4000, 03, 12, 1976, true);
Member mem5 = Member("John", "Doe", 5000, 03, 12, 1976, true);
Member mem6 = Member("Jane", "Deer", 6000, 15, 06, 1990, true);
Member mem7 = Member("Helena", "Bell", 7000, 01, 06, 1990, true);
Member mem8 = Member("Dawn", "Morris", 8000, 30, 06, 1990, true);
Member mem9 = Member("Gordon", "Winks", 9000, 28, 07, 1990, true);
Member mem10 = Member("Ryeowook", "Kim", 10000, 20, 01, 1990, true);
Member mem11 = Member("Heechul", "Kim", 1001, 13, 8, 2017, true);
Member mem12 = Member("Caitlin", "Morris", 2001, 30, 06, 2000, true);
Member mem13 = Member("John", "Watson", 3001, 22, 10, 2009, true);
Member mem14 = Member("Antonio", "Black", 4001, 01, 05, 1745, true);
Member mem15 = Member("Joseph", "White", 5001, 19, 03, 2001, true);
Member mem16 = Member("Adam", "White", 6001, 19, 03, 1980, true);
Member mem17 = Member("Laura", "Smith", 7001, 13, 03, 1981, true);
Member mem18 = Member("Josh", "Collie", 8001, 13, 03, 1981, true);
Member mem19 = Member("Ann", "Brown", 9001, 29, 11, 1979, true);
Member mem20 = Member("Jennifer", "Fowler", 1002, 19, 03, 1950, true);
Member mem21 = Member("Blake", "McDonald", 1003, 19, 03, 1964, true);
```

Includes some with the same join years, join dates, and some with the same dates and last names, and some in every membership length category, and some with join dates on the category boundaries.]

[Select the option to view the long-standing members.]

Result: *Please enter the current date as individual integers:*

Input: [enter a date of 19/3/2017 – see option 1 for date validation testing]

Result: [Displays all current members and only current members, in order of date joined, or last name if the dates are the same, or first name if the dates and last names are the same, split up according to how long they have been members. Members with < 10 years membership are not included, and members on the category boundary (join dates of 19/3/?? here) are correctly sorted into the correct category.]:

```

Long-standing members on 19/3/2017:

50 years+

Mem No.      Date Joined      Name
4001         1/5/1745        Black, Antonio
1002         19/3/1950       Fowler, Jenifer
1003         19/3/1964       McDonald, Blake

40 years+

Mem No.      Date Joined      Name
5000         3/12/1976       Doe, John
4000         3/12/1976       Potter, Harry
3000         3/12/1976       Tigwell, Garreth

30 years+

Mem No.      Date Joined      Name
9001         29/11/1979      Brown, Ann
6001         19/3/1980       White, Adam
8001         13/3/1981       Collie, Josh
7001         13/3/1981       Smith, Laura

20 years+

Mem No.      Date Joined      Name
10000        20/1/1990       Kim, Ryeowook
7000         1/6/1990       Bell, Helena
6000         15/6/1990      Deer, Jane
1000         30/6/1990      Morris, Carrie
8000         30/6/1990      Morris, Dawn
9000         28/7/1990      Winks, Gordon
2000         28/2/1992      Stewart, Kayleigh

10 years+

Mem No.      Date Joined      Name
2001         30/6/2000      Morris, Caitlin
5001         19/3/2001      White, Joseph

1. Add a member
2. View long-standing members
3. Revoke membership from a member
4. View list of current members
5. View list of past members
Please enter the number of your choice or 0 to exit:

```

### Option 3 – Revoke membership from a leaving member

#### Invalid Input – Not In Current Members List

Input: [Select the revoke membership option from the menu with only the test data members in the list]

Result: *Please enter the membership number of the member leaving the society:*

Input: 1

Result: *Sorry, that member doesn't appear to be on our list of current members.*

#### Valid Input – In Current Members List

Input: [Select the revoke membership option from the menu with only the test data members in the list]

Result: *Please enter the membership number of the member leaving the society:*



Input: 1001

Result: *The membership has successfully been revoked.*

[Inspection shows the member is no longer in the current members list and is now in the past members list]

## Displaying the list

### Current Members

Input: [Select the display current members option from the menu]

Result: [Prints a list of all current members in whatever order the list currently is in which is unsorted if it's the first thing done upon opening the program, sorted if it's done after selecting an option that involves sorting the list.]

### Past Members

Input: [Select the display past members option from the menu]

Result: [Prints a list of all past members that is unsorted, as past members are added without sorting and never require sorting when using the program.]

## Continuing Use

Input: [Add 3 members to the lists – 2 in current, 1 – past, then revoke the membership for all current members (i.e. current list is empty but past members will have many more) and try to view a list of long-standing members]

Result: *Sorry, there are no current members on the list at all!*

Input: [Select the option to display the list of current members]

Result: *Sorry, it looks like there's no current members on the list to display.*

Input: [Select the option to display the list of past members]

Result: [Displays the list of past members which now includes all of the members that were previously in the current members list]

Input: [Add 3 more members via option 1]

Result: [All members have been successfully added to the current members list]