

Assignment on Dynamic Array Lists

Building a DAL from Scratch in C++ and making a Library Mgmt Program with it

Due on: **02:00** PM, Saturday, 10 April 2021

Total Marks: 30

CSE 204 (Data Structure and Algorithms Sessional - I)

Lec Raiyan Rahman

Dept of CSE, MIST

raiyan@cse.mist.ac.bd

# **Introduction**

Dynamic Array Lists can be a very useful way of working with arrays especially when we’re looking for efficiency in terms of space. In the last lab, we got introduced to the concept of Dynamic Arrays and how they can be implemented. Then we saw two ways we can implement in using C++:

1. Using the built-in STL called Vector. (std::vector).
2. Making a DAL from scratch.   
   (This gave us much more flexibility on how we can use and implement such dynamic arrays).

Now, it’s time for you to put the concept into use and make the library mgmt program we made in CSE 106 more efficient with DALs. You’ll also need to add some additional features on top of the code we did in class.

How to Prepare:

1. In case you need a refresher, check your class notes in case you have one. This is usually the best place to start.
2. Go through the slides and the sample codes. Check code 1 and code 3 carefully, I’ve added some more to them.
3. This should give you a refresher. In case you need more, check out the class recording.
4. Then read the assignment statement carefully and get to coding!

**Class Resources:** [click here](https://drive.google.com/drive/folders/1WuvnrgI-TLy4C1d9rpxAKMNn80E_G-6z?usp=sharing)

**Class Recording:** [click here](https://youtu.be/mVYyO2-j9_A)

## Assignment 1 (20 marks)

Adding some additional features to our DAL program. Here’s what you need to do:

1. Download the “Code 1 - DAL from Scratch.cpp” from [this](https://drive.google.com/drive/folders/1WuvnrgI-TLy4C1d9rpxAKMNn80E_G-6z?usp=sharing) folder. **Write your code on top of it in the appropriate portions.**
2. Implement the menu options 3-7 on your own.
3. The description for each of the option is given in the following table:

| **Initially, consider the current Dynamic Array is:**  **5 6 5 4 5 7 2 4 9 8** | | | |
| --- | --- | --- | --- |
| **Menu Option** | **Description** | **Sample Input** | **Sample Output** |
| 3 | Let the user delete the value from a particular position in the DAL. | Enter Position to Delete from: 6 | Deleted the value in position 6 which was ‘7’.  The current Array is:  **5 6 5 4 5 2 4 9 8** |
| 4 | Let the user insert a value into a particular position in the DAL. | Enter Position to Insert Into: 6  Enter the Value to insert: 7 | Inserted ‘7’ at position 6.  The current Array is:  **5 6 5 4 5 2 4 9 8** |
| 5 | Delete **ALL** matching values. Resize the Dynamic array as per the algorithm shown in class  (think when it’ll be resized carefully). | Enter the value to delete: 5 | Deleted all occurrences of ‘5’.  Total 3 items deleted. Array Resized.  The current Array is:  **6 4 2 4 9 8** |
| 6 | Replace **ALL** matching values.  There may be two cases as shown in the sample input-output. | Enter the value to replace: 5  Enter the value to be replaced by: 6 | No occurrence of ‘5’ was found. No Items were replaced. |
| Enter the value to replace: 8  Enter the value to be replaced by: 6 | ‘8’ replaced with ‘6’. Total 1 replacement.  The current Array is:  **6 4 2 4 9 6** |
| 7 | A unique function that you think can be beneficial for DALs.  Write appropriate **comments** on top of your function in the C++ code. | Your function. Your Design. | |

## Assignment 2 (10 marks)

Rewriting the library management program from CSE 106 in an efficient way using DALs. For this-

1. Once you’re done with Assignment 1, copy and paste the appropriate portion of the custom Dynamic array code (the class and the class functions) on top of the library code.   
   You may also include it as a user-defined header file. For example, #include “myDAL.h”.
2. Now rewrite the library code so that it uses the “DynamicArray” class that you created to declare the array of objects. (struct library l[num]). Make appropriate changes throughout the code.
3. You may use the std::vector instead of the “DynamicArray” class. No marks will be reduced.   
   Check the code “code 3 DAL with vector - with objects.cpp” for an example of how to use vectors with structures.

Note: All the codes (incl. The library mgmt code) are given in the resources folder [linked](https://drive.google.com/drive/folders/1WuvnrgI-TLy4C1d9rpxAKMNn80E_G-6z?usp=sharing) above.

Submission Guidelines:

1. Name the Assignment 1 as **yourStudentID\_Assignment1**.cpp. For example, 202014005\_Assignment1.cpp.
2. Name the Assignment 2 as **yourStudentID\_Assignment2**.cpp. For example, 202014005\_Assignment2.cpp.   
   If you use a custom header file for it, name it **yourStudentID\_header**.h.
3. **Submit all files individually using the “turn in” option in the classroom. No need to ZIP them.**