

**CSE225L – Data Structures and Algorithms Lab**  
**Lab 02**  
**Classes & Objects**

Recall the class for dynamic array we developed in our 2<sup>nd</sup> lecture. In this lab session, you will implement and test that class.

**Before you begin:** Create a C++ project. Then create a header file and a source file for your project. Follow the steps below to create files.

1. Create a header file.
  - a. File ⇒ New ⇒ File. Select C/C++ header. Select Go ⇒ Next.
  - b. Click the browse button beside the textbox labelled “Filename with full path”. A window appears. In the textbox labelled “File name”, type in the name you want to assign to the header file (in this case dynarr.h). Select Save ⇒ select All ⇒ select Finish.
  - c. Under the "Management" pane ⇒ Choose "Projects" tab ⇒ Expand the project node ⇒ Expand "Headers" node. You will see the file you just created. It is mostly blank except for a few lines of code. It is time to add your own code to it. For your convenience, the content of this file is already provided with this manual (refer to the table in the opposite page). Add the content of dynarr.h just before the `#endif` statement.
2. Next, create the source file
  - a. File ⇒ New ⇒ File. Select C/C++ source. Select Go ⇒ Next.
  - b. Select "C++" ⇒ Next.
  - c. Click the browse button beside the textbox labelled “Filename with full path”. A window appears. In the textbox labelled “File name”, type in the name you want to assign to the source file (in this case dynarr.cpp). Select Save ⇒ select All ⇒ select Finish.
  - d. Under the "Management" pane ⇒ Choose "Projects" tab ⇒ Expand the project node ⇒ Expand "Sources" node. You will see the file you just created. It is completely blank. It is time to add your own code to it. For your convenience, the content of this file is already provided with this manual (refer to the table in the opposite page). Add the content of dynarr.cpp to the newly created file.
3. In the opposite page, will find some tasks that you have to perform using the code provided. You have to modify the file main.cpp and perform these task in that file.
4. To build the program, select "Build" menu ⇒ Build.
5. To run the program, select "Build" menu ⇒ Run.

<pre> <b>dynarr.h</b>  #ifndef DYNARR_H_INCLUDED #define DYNARR_H_INCLUDED class dynArr {     private:         int *data;         int size;     public:         dynArr();         dynArr(int);         ~dynArr();         void setValue(int, int);         int getValue(int); }; #endif // DYNARR_H_INCLUDED </pre>	<pre> <b>dynarr.cpp</b>  #include "dynarr.h" #include &lt;iostream&gt; using namespace std;  dynArr::dynArr() {     data = NULL;     size = 0; } dynArr::dynArr(int s) {     data = new int[s];     size = s; } dynArr::~dynArr() {     delete [] data; } int dynArr::getValue(int index) {     return data[index]; } void dynArr::setValue(int index, int value) {     data[index] = value; } </pre>
---	---

### Tasks:

**Task 1:** In the driver file (main.cpp), perform the following sub-tasks.

1. Create two objects of this class, one with no constructor argument and one with the argument 5.
2. Take five input values from the user and store them in the array inside the second object using the set method.
3. For the second object, print all the values you just stored.

Note that, you cannot assign anything in the first object since the array inside it has size 0. Neither can you change the size of this array to some other size.

**Task 2:** Modify the header and the source files. Add a member function **void allocate(int s)** which allows you to change the size of the array. Make sure that memory is not leaked.

**Task 3:** Modify the header file and the source files again, so that it works for two dimensional array where all the rows are the same size. The user will specify the number of rows and columns as well as the content of the array, which you will take as input from user in the main function.