LAB 3

CSE225L

Classes & Objects

In this lab, we will:

- Implement and test the **dynArr** class using the header (**.h**) and source (**.cpp**) files.
- Demonstrate the use of constructors and destructors for creating and deleting arrays.
- Utilize the **setValue** and **getValue** methods to modify and access array elements.
- Modify the class to include functionality for dynamically resizing the array using **allocate(int s)**.
- Extend the class to support a 2D array where all rows have the same column size.

DYNARR

dynarr.h

```
#ifndef DYNARR_H
#define DYNARR_H

class dynArr
{
private:
    int *data;
    int size;

public:
    dynArr();
    dynArr(int);
    ~dynArr();
    void setValue(int, int);
    int getValue(int);
};

#endif // DYNARR_H
```

dynarr.cpp

```
#include "dynarr.h"
#include <iostream>
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;
}
```

DYNARR

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.

Task 2:

- Modify the header and source files to add a member function void **allocate(int s)** that allows you to change the size of the array. Ensure that memory is not leaked.

Task 3:

- Modify the header and source files again to support a two-dimensional array where all rows are the same size. The user will specify both the number of rows and columns, as well as the content of the array. You will take the input from the user in the main function.