

You are asked to implement a C++ class to model a **sorted array of unsigned integers**. The class is to be used in an embedded application that cannot assume the presence of the STL. The array has to be dynamically allocated in such a way that allows programmers using it to specify the required size.

Your class should:

- (1) provide the appropriate constructors and destructor;
- (2) provide methods for updating, and showing/setting numbers in/to the array (e.g., to be used like: `somearray.setAt(index, value)` and `somearray.getAt(index)`);
- (3) provide overloaded operators: assignment (`=`), index (`[]`), equality (`==`), and stream output (`<<`);
- (4) make use of move semantics when appropriate; and
- (5) provide error reporting to client programmers using the class.

Finally, you should also provide a test program to demonstrate that your class behaves as expected (and that the array remains sorted).

Assignment Submission: assignment should be submitted on black board as a .zip file that contains all your source code and how to compile and build your assignment. Please name your zip file as follows: **your-full-name-AUCID-A2.zip**.

More Tips and Remarks on the next page ...

◆ **More tips and remarks for this assignment:**

1 - You need to divide your code into 3 files, header file (.h), and 2 (.cpp) files (one for constructors and functions implementations and the other for testing the code functionality).

2 - As stated in the assignment description "you need to provide the appropriate constructors and destructor", so you have to provide an implementation for each of the following:- **default constructor, int constructor, copy constructor, move constructor, and destructor.**

3 - You are required to provide an implementation for:

- **setAt method - getAt method - copy assignment operator - move assignment operator - equality operator - stream output operator (<<).**

4 - Also you are required to provide an implementation for **index operator [] constant and non-constant versions (which allow for both reading and modifying of an array element).**

5 - The most important part of this assignment is to keep your array sorted at any point of your program, you may think of sorting with any algorithm when inserting a new value to your array.

6 - **There is a nice trick when trying to use the non-const version of [] operator and trying to modify an element of your array.** You have to keep your array sorted at any time, you can consider the following code segment.

```
{ // Assume A is SortedArray object holds { 5, 7, 8, 11}
    A[2] = 3; // after this statement A should be { 3, 5, 7, 11}
    cout << A[2]; // should print 7
    // ...
}
```

7 - You have to **provide a clear error reporting**, it's recommended to use exceptions and try to use classes like 'invalid_argument' and 'out_of_range' in <stdexcept> and catch error using try-catch block in main. (you can check examples in Ch. 9 of Deitel book (9th edition) to see how to use exceptions).

8 - **You have to care about input validation and boundary checking, defining a function as const or not, and the return type and parameter type for each function.**

9 - You may need to initialize array elements at beginning to a default value, assuming that you will sort your array in ascending order, so recommended to initialize elements by 'UINT_MAX' value.

10 - **Recommended to review Array class in Ch. 10 of Deitel book (9th edition), it's a very helpful reference for all parts of that assignment.**

11 - You don't **need to submit a whole project**, you just need to submit your source code files (.h file and two .cpp files), you can include them in a compressed file (e.g. yourName_ID.rar) and upload it to the blackboard.