OOD Spring 2024
Assignment 1 : ArrayOfBits **Due: Tuesday March 5th, 2024**

1.1 You are asked to implement a C++ class to model an array of bits of arbitrary size. The idea is that we want to store **n** bits and want to make sure that we ***do not waste storage and each bit occupies exactly one bit of memory.*** This is because the class is to be used in an embedded application that cannot assume the presence of the STL and must also use the smallest amount of storage possible. Your class have to offer an API that enables looking up and setting the value of a bit (for example, x.**getValue**(25) would return the value of the 25th bit, and x.**setValue**(25, 1) would set its value to 1). You should also provide the following:

- appropriate constructors, destructor, and member function **printArray** which prints only bit indices which have the value of 1.
- member function **merge**: z = x.**merge**(y) returns a new object z to be the *union* of the two arrays of x and y. The z object should accommodate the bits in both x and y with their values, such that the bit at index i of z will be 1 only if the corresponding bit in x is 1 or the corresponding bit in y is 1.
- member function **differ**: z = x. **differ**(y) returns a new object z to be the *difference* between the two arrays (x - y). The z object should accommodate the bits in x such that the bit at index i of z will be 1 only if the corresponding bit in x is 1 and the corresponding bit in y is 0.

You are also asked to implement a main program that uses your class to demonstrate the above interfaces and functionality.

1.2 You are asked to build on your work in (1.1) to provide the index operator (**[ ]**) to enable accessing specific bits in your array. For example, to enable code like:

```
{ // ...
      BitArray b(40);
      cout << b.getValue(25); // prints the value of the bit at index 25
      cout << b[25];     // also prints the value of the bit at index 25
  // ...
}
```

Please provide a test program to demonstrate that your class and operators behave as expected and handles errors in an appropriate manner.

Assignment Submission: assignment should be submitted on Blackboard 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-A1.zip.

More tips and remarks on the next page...

◆ simple tips for this assignment:

1) You need to divide your code into 3 files, one header file (.h), and 2 (.cpp) files (one for constructors and functions implementations and the other for testing the code functionality).
2) You're **not allowed to use any built-in class/library** as it's mentioned "cannot assume the presence of the STL".
3) You need to model an array of bits of arbitrary size, so need to provide an **int constructor** that allows the user of your class to specify the number of bits that the user likes to store, also recommended to provide a **default constructor** that specify a default value of the number of bits (e.g. 32).
4) You need to use the dynamic memory allocation to handle the arbitrary size, so you have to provide a valid implementation for each of **copy constructor, move constructor, and destructor**.
5) You have to take care of the **input validation** for your constructors and functions, the appropriate type of the parameters, and when you need to define a function to be const or not.
6) Provide **error reporting**, at least printing a message in case of an invalid input.
7) Your test file has to be complete which means you have to test the functionality of each part of your code and make sure it works fine.
8) With respect to 1.2: (**No need to create a new class**) you're just required to update your class in part 1.1 and add an implementation for [ ] operator (**only the const version of it which allows only for reading the bit value at a specific index**) as stated in the code segment provided in the assignment description, also need to update your test program to demonstrate that your class and operators behaves as expected.
9) 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.
10) Part of the points are awarded for a professional evaluation of your submission.

If you have any questions, don't hesitate to send us an email.