Assignment 2-part 2

Deadline:	**Anytime before** Sunday, 29th May 2016, 23:59 (midnight)
Evaluation:	10 marks – which is 5% of your final grade
Late Submission:	5% per hour (or fraction of hour) it is late
Purpose:	Practice with generic programming by using C++ templates.

Problem to solve:
You will design and use a class template. The name of the class template must be Bin. The type "Bin" represents an unordered collection of n items (at locations 0 to n-1), where each item is of a generic type called "itemType". Duplicate items are allowed. You may assume that the type "itemType" supports “operator=()" and "==".The minimum required interface for the class Bin is presented in the table below. Your solution should overload the output operator such that code as presented in Figure 1 will work to provide the output provided in Figure 2. You should write all your code in a file named **a2p2.h**

The a2p2.h file should be organized as follows:
1. Comments about the authors of the solution, about reasonable assumptions you made and any other facts we should be aware when testing your solution
2. Including all necessary files
3. The **Bin class template** listing-no member function implementation inside the class is allowed.
4. The **Bin class template** member functions implementations,
5. Implementation of all global functions, including the function `printInfo()`, that should display on screen all authors of the assignment solution submitted for marking.

The Bin class should have (at least) all of the following (both A and B parts):
A) **private** members:

Name	Type	Purpose
`capacity`	`unsigned int`	Maximum number of items that can be stored in the Bin before reallocation occurs.
`size`	`unsigned int`	The actual number of items in the Bin.
`array`	`itemType*`	The address of the array of items.
`reallocate()`		It increases the Bin’s object capacity.

B) **public** function members:

Name	Purpose
`Bin()`	Default constructor. Initializes the Bin to empty (array points to NULL, size and capacity are 0).
`~Bin()`	Destructor. De-allocates memory & updates size and capacity to 0.
`Bin(..), ..`	The copy and the move constructors
`..operator=(..),..`	The assignment operators (copy and move)
`length() const`	Returns the number of actual items in the Bin.
`isEmpty()const`	Returns true if the Bin has no item and false otherwise.
`print()const`	Displays all "size" elements in a table of 5 elements per row (10 spaces per element).
`find( const itemType& x ) const`	Locates the first instance of a specified item. Returns the item's position if it was located; size otherwise.

`retrieve(itemType& x,unsigned i)const`	Retrieves the item at position i (unless i is not valid). Returns true if the item was retrieved; false otherwise.
`remove( itemType& x, unsigned i )`	Removes the item at position i (unless i is not valid). Rearranges the elements in the Bin such that no empty place occurs inside the Bin object. Returns true if the item was removed; false otherwise.
`insert(const itemType &x, unsigned i)`	Inserts x into the Bin at position i. If the Bin is full, allocates additional space before inserting the item. Returns true if the item was inserted; false otherwise.

Make sure that **all features** of your **Bin** class implementation work as specified in the requirements and as discussed in the lectures/labs.

Hand-in:
Submit **a2p2.h** electronically using STREAM.

Miscellaneous:

1. Using exception is not required for this assignment, but if you want you can use them.
2. Write YOUR ID NUMBER(S), and YOUR **FAMILY** NAME(S) first, assignment number, what the program does at the beginning of the file you send electronically and *at least* comment each function.
3. When working in pairs, send one solution file per pair.
4. Marks will be allocated for: correctness, completeness, use of C++ constructs, **good OOP style**-as presented in lectures & labs, good structure for the solution, documentation, and clear on screen output display.
5. Using goto, **global variables** or C-like I/O constructs (i.e *printf*, *fprintf*, *scanf*, *FILE**, etc) is not allowed and it will be penalised. Only **const** global variables are allowed.
6. Programs that do not compile in the lab, using gcc, **get 0 marks**.
7. Do not include a main function in your a2p2.h file. An a2p2.h file containing a main function **will get 0 marks.**
8. STL containers should not be used for this assignment and any solution using an STL container will **get 0 marks.**

1 main.cpp

```cpp
#include "a2p2.h"
using namespace std;

//Demonstrate selected operations for the Bin type
int main(){
  printInfo();      //display authors details
   //------------Flaot elements used---------------
  Bin<float> flObj; //using Bin of float values

  flObj.insert( 500.6, 0 ); flObj.insert( 300.78, 1 );
  flObj.insert( 400.8, 0 ); flObj.insert( 700.1, 2 );

  cout<<flObj<<endl;

  cout << "\nValue returned by 'isEmpty': " << (flObj.isEmpty()?"true":"false") << endl;

  float item=0.0;     //to store retrived values
  bool flag;          //was retriving successful?
  unsigned last=flObj.length();
  for (unsigned i=0; i <=last; ++i){
    flag = flObj.retrieve( item, i );
    cout << "Retrive from position " <<i<<" was"<<
            (flag == true ? " succesful ":" unsuccesful ");
    cout<< " item: " << (flag == true ? item: 0)  << endl;
  }
  item = 800;
  unsigned position = flObj.find( item );
  cout<<"Element " <<item <<( position < flObj.length() ? " ":" not ")<<"found.";
  cout << "\nValue returned by 'isEmpty': " << (flObj.isEmpty()?"true":"false");
  cout << "\nValue returned by 'length': " << flObj.length();
  //------------String elements used---------------
  cout<<"\n\n A bin of strings:\n";
  Bin<string> strObj;  //using Bin of string values
  strObj.insert( "be happy!", 0 );  strObj.insert( "worry", 0 );
  strObj.insert( "Don't",  0 );

  cout<<strObj;

  cout<<"\n\nAll done! Bye. Bye...";
}
```

Figure 1. A possible driver for testing the Bin type

```
main

 159.234 A 2 part 2
 Calude E.  00010001
 Duck D.     0000002


     400.8     500.6     700.1    300.78

Value returned by 'isEmpty': false
Retrive from position 0 was succesful  item: 400.8
Retrive from position 1 was succesful  item: 500.6
Retrive from position 2 was succesful  item: 700.1
Retrive from position 3 was succesful  item: 300.78
Retrive from position 4 was unsuccesful  item: 0
Element 800 not found.
Value returned by 'isEmpty': false
Value returned by 'length': 4

 A bin of strings:

     Don't     worry be happy!

All done! Bye. Bye...
```

Figure 2. Output produced by the program in Figure 1.

If you have any questions about this assignment, please ask the lecturer before its due time!