

Problem Set 08 - Vector

A vector is a resizable array. The typical vector data structure methods are discussed in Lecture 04. Given this information, complete each of the tasks below.

Tasks:

1. Create a header file named "Vector.h" that defines a generic class *Vector* that publicly inherits the interface *ArrayList* and *Object* from the accompanying header files "ArrayList.h" and "Util.h", respectively.
2. Within the body of *Vector*, include
 - ☐ A private generic *Array* field named *data*.
 - ☐ A private long field named *size*.
3. Within the body of *Vector*, define
 - ☐ A public default constructor that assigns 0 to *size* and a generic *Array* object of size 30 to *data*.
 - ☐ A public overloaded constructor that takes a long parameter. It assigns a generic *Array* object of a size equal to the parameter to *data* and 0 to *size* if the parameter is positive; otherwise, it assigns a generic *Array* object of size 30 to *data* and 0 to *size*.
 - ☐ A public copy constructor.
 - ☐ A public assignment overloaded.
 - ☐ A public empty destructor.
4. Within the body of *Vector*, define a public overridden `ToString()` method from *Object* that returns a string which is a list of the elements of the dataset all enclosed in square braces such that a comma separates each element.
5. Within the body of *Vector*, define a public overridden `Resize()` method from *ArrayList* that resizes *data* to the size of the parameter if the parameter is positive and retain all values of *data* that are within the new size range.
6. Within the body of *Vector*, define
 - ☐ A public overridden `Size()` method from *ArrayList* that returns *size*.
 - ☐ A public overridden `IsEmpty()` method from *ArrayList* that returns true if *size* is zero; otherwise, it returns false.
7. Within the body of *Vector*, define
 - ☐ A public overridden `Search()` method from *ArrayList* that returns the index of the first occurrence of the parameter in the dataset if the parameter is a dataset member; otherwise, it returns the size of the dataset.
 - ☐ A public overridden `Contains()` method from *ArrayList* that returns true if the parameter is a dataset member; otherwise, it returns false.
 - ☐ Public overridden `At()` methods from *ArrayList* that returns the element of the dataset whose index is equal to the parameter if the parameter is a valid dataset index; otherwise, it throws an "out of bound" error message.
 - ☐ Public overridden `operator[]()` methods from *ArrayList* that returns the element of the dataset whose index is equal to the parameter if the parameter is a valid dataset index; otherwise, it throws an "out of bound" error message.

8. Within the body of *Vector*, define

- ☐ A public overridden **Append()** method from *ArrayList* that adds the parameter to the end of the dataset. It resizes the dataset before the insertion if the dataset is at capacity.
- ☐ A public overridden **InsertAt()** method from *ArrayList* that adds the first parameter to the dataset at the index equal to the second parameter if the second parameter is a valid dataset index. It resizes the dataset before the insertion if the dataset is at capacity.

9. Within the body of *Vector*, define

- ☐ A public overridden **Erase()** method from *ArrayList* that removes the first occurrence of the parameter from the dataset if the parameter is a dataset member; otherwise, it does nothing.
- ☐ A public overridden **Detach()** method from *ArrayList* that removes the last element from the dataset.
- ☐ A public overridden **Remove()** method from *ArrayList* that removes the element of the dataset whose index is equal to the parameter if the parameter is a valid dataset index; otherwise, it does nothing.

10. Create a cpp file named "main.cpp" that

- ☐ Includes the '**Vector.h**' header file.
- ☐ Declares a vector object.
- ☐ Inserts the first 30 consecutive positive multiples of 4 into the object.
- ☐ Displays the content of the object.
- ☐ Resizes the object to 15.
- ☐ Displays the content of the object.