CST 238 Fall 2020, Dr. Gross, CSUMB

Homework XX. ArrayList

Due: XXX

Objectives:

- Use dynamic arrays to implement a list
- Resizing the dynamic array

How to turn in:

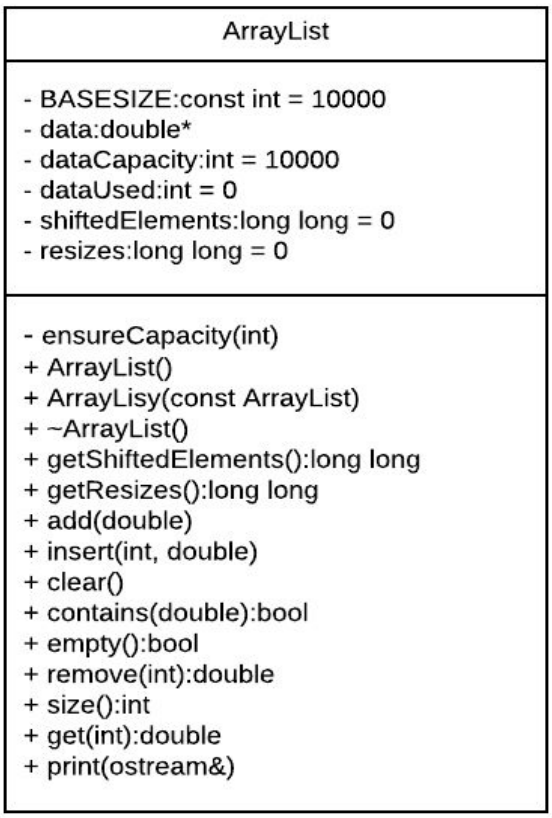
- Submit the `ArrayList.cpp` to ILearn
- *Follow instructions carefully* or be prepared to lose points

Step 1. Getting Started

1. Create a new repo (in an IDE of your choosing)
2. Add the files `List.h`, `ArrayList.h`, and `List.cpp` to the repo
3. Create a new file called `ArrayList.cpp`, and include `ArrayList.h` in that file
    a. Don't forget to write your name, date and synopsis
4. Create a `main.cpp`, include `ArrayList.h` and `List.h`

Step 2. Implement and Test

Class diagram: (need to update: add the `getCapacity()` method to diagram)



Brief descriptions of attributes and methods:

`BASESIZE`: capacity of arraylist at start
`data`: The pointer to the first element in the arraylist
`dataCapacity`: current capacity of arraylist
`dataUsed`: current number of elements in the arraylist
`shiftedElements`: number of elements that have been shifted during inserts & removes
`resizes`: number of times the arraylist has had to increase capacity by BASESIZE
`ensureCapacity`(int): a method to check if an insert/add is valid, if not, it will resize
`ArrayList()`: constructor
`ArrayList(const Arraylist&)`: Copy Constructor
`~ArrayList()`: Deconstructor
`getShiftedElements()`: returns the value of shiftedElements
`getResizes()`: returns the value of resizes
`add(double)`: add a new element to the end of the list (append)
`insert(int, double)`: insert a value into the arraylist at any index
`clear()`: deletes all the elements in the arraylist
`contains(double)`: checks if a value is present in the list, returning true if so, false otherwise
`empty()`: checks if the arraylist is empty, returns true/false
`remove(int)`: removes an element from the arraylist at a given index
`size()`: returns the value of dataUsed
`get(int)`: returns the value of the given index, if it is valid, -1 otherwise
`print(ostream&)`: prints the elements in the list

Attribute/Method Name	Description
`BASESIZE`	capacity of arraylist at start

NOTE: `print(ostream&)` should output the list in the same format as shown below (`[0, 1, 2, 3]`)

Example of tests (including but not limited to) you should run:

(try two column src/output)

```
cout << "===== empty =====" << endl;
if(a.empty()){
    cout << "size = " << a.size() << endl;
}

cout << "\n===== add 3 elements =====" << endl;
for(int i = 0; i < 3; i++){
    a.add(i);
}
if(a.size() == 3){
    cout << "size = " << a.size() << endl;
}

cout << "\n===== not empty =====" << endl;
if(!a.empty()){
    cout << "list is not empty\n";
}

cout << "\n===== clearing list =====" << endl;
a.clear();
if(a.empty()){
    cout << "size = " << a.size() << endl;
}

cout << "\n===== inserting 5 elements =====" << endl;
for(int i = 0; i < 5; i++){
    a.insert(i, i);
}
cout << "list elements: " << a << endl;
cout << "size = " << a.size() << endl;


cout << "\n===== contains 3 =====" << endl;
if(a.contains(3)){
    cout << "list contains 3\n";
}

cout << "\n===== does not contain 5 =====" << endl;
if(!a.contains(5)){
    cout << "list does not contain 5\n";
}

cout << "\n===== removing 2 =====" << endl;
a.remove(2);
if(!a.contains(2)){
    cout << "size = " << a.size() << endl;
    cout << "list elements: " << a << endl;
}

cout << "\n===== removing last 2 elements =====" << endl;
for(int i = 2; i > 0; i--){
```

```
    a.remove(a.size() - 1);
}
cout << "size = " << a.size() << endl;
cout << "list elements: " << a << endl;


cout << "\n===== inserting 3 elements into the middle =====" << endl;
for(int i = 4; i > 1; i--){
    a.insert(a.size()/2, i);
}
cout << "size = " << a.size() << endl;
cout << "list elements: " << a << endl;
```

Output:

```
===== empty =====
size = 0

===== add 3 elements =====
size = 3

===== not empty =====
list is not empty

===== clearing list =====
size = 0

===== inserting 5 elements =====
list elements: [0, 1, 2, 3, 4]
size = 5

===== contains 3 =====
list contains 3

===== does not contain 5 =====
list does not contain 5

===== removing 2 =====
size = 4
list elements: [0, 1, 3, 4]

===== removing last 2 elements =====
size = 2
list elements: [0, 1]

===== inserting 3 elements into the middle =====
size = 5
list elements: [0, 3, 2, 4, 1]
```

Rubric:

Method	**Points**
empty	10
insert	26
remove	20
add	16
clear	8
contains	5
get	7
getCapacity	4
getResizes	2
getShiftedElements	2
Total points	**100**

(add notes that explain insert to empty, non-empty, and cause resize will have effects on their grade)
Note: `getCapacity, getResizes, & getShiftedElements` are all implemented in `ArrayList.h`, so the objective is to return the correct number.

Final Notes:

- Develop tests in `main.cpp` to check the functionality of each method you implement
- Upload both `ArrayList.cpp` and `main.cpp`
- `No cout statements`