

CS 401: Introduction to Advanced Studies
Department of Computer Science
Illinois Institute of Technology

Lab 5

1. (5 points) Complete the code in CS401ArrayImpl.java as discussed in class.

Your output, when you run CS401Array.java should be as follows:

```
Inserting element at index 0
Inserting element at index 1
Inserting element at index 2
Inserting element at index 3
Inserting element at index 4
Capacity reached. Increasing storage...
Inserting element at index 5
Inserting element at index 6
There are 7 chores to do today!
These are:
0 [Make bed:10]
1 [Make dinner:20]
2 [Exercise:30]
3 [Walk the dog:15]
4 [Watch movie:120]
5 [Sleep:480]
6 [Do CS401 homework:80]
There are 9 chores to do today!
These are:
0 [Listen to music:60]
1 [Make bed:10]
2 [Make dinner:20]
3 [Read a book:50]
4 [Exercise:30]
5 [Walk the dog:15]
6 [Watch movie:120]
7 [Sleep:480]
8 [Do CS401 homework:80]
[Read a book:50] is present
[Jog:20] is not present
[Read a book:50] removed
There are 8 chores to do today!
These are:
0 [Listen to music:60]
1 [Make bed:10]
2 [Make dinner:20]
3 [Exercise:30]
4 [Walk the dog:15]
5 [Watch movie:120]
6 [Sleep:480]
7 [Do CS401 homework:80]
```

2. (4 points) Implement a method to return a portion of a list in CS401ArrayImpl.java. The signature of the method is as follows:

```
/**
 * subList(int from, int to): Returns the view of a portion of the list between the specified index from,
 * inclusive, and index to, exclusive. That is returns a sublist [from, to).
 *
 * Ensure that from and to are valid indices.
 *
 * The method returns a new CS401ArrayImpl<E> object.
 */
public CS401ArrayImpl<E> subList(int from, int to);
```

Here is a test program to test your code.

```
public class CS401ArrayTest
{
    public static void main(String[] args) {
        CS401ArrayImpl<Integer> numbers = new CS401ArrayImpl<Integer>();

        numbers.add(100); numbers.add(200);
        numbers.add(300); numbers.add(400);
        numbers.add(500); numbers.add(600);
        numbers.add(700); numbers.add(700);

        print_list(numbers);

        CS401ArrayImpl<Integer> sub = numbers.subList(1,4);
        print_list(sub);
    }

    private static void print_list(CS401ArrayImpl<Integer> numbers) {

        for (int i = 0; i < numbers.size(); i++)
        {
            System.out.print(numbers.get(i) + " ");
        }
        System.out.println();
    }
}
```

The sublist should contain 200, 300, 400; that is the sublist is composed of the indices [1,4), or 3 elements.

3. (1 point) Assume you have a linked list whose current state is depicted as shown on the next page. Add a new element with the value 10 at the beginning of this list.

Draw each step as the new value is added to the beginning of the list.

See this week's lecture and note that we went through 4 steps to add an object to the linked list: (1) allocate memory for the specific object, (2) allocate memory for a linked list element to hold the specific object, (3) insert the specific object into the linked list element, and (4) insert the linked list

element into the linked list. In our class the linked list was holding objects of type Chores. In this problem the linked list is holding regular *int* objects, as such, you do not need step (1).

You may draw multiple panels, in each panel please show the state of the linked list and how the head reference changes, etc.

