

Due Mar 27 at 11:59pm**Points** 100**Questions** 30**Available** Mar 22 at 12:01am - Mar 27 at 11:59pm 6 days**Time Limit** 240 Minutes

Instructions

Midterm Exam

Instructions

You have four hours to complete the exam. **You must complete the test in one sitting.**

This test is open-note/open-book/open-web, but **must be completed individually**. Communicating (in person or online) with anyone other than me is considered cheating and will result in a failing grade and disciplinary procedures.

Note: I recommend completing the exam questions in your own IDE and then pasting the answers into Canvas when you are ready to submit. You can review the [Posting Code to Canvas](#) if needed.

* Canvas Resources

 [How do I take a quiz? \(https://community.canvaslms.com/docs/DOC-10645-421241977\)](https://community.canvaslms.com/docs/DOC-10645-421241977)

 [How do I submit a quiz? \(https://community.canvaslms.com/docs/DOC-10583-421250759\)](https://community.canvaslms.com/docs/DOC-10583-421250759)

This quiz was locked Mar 27 at 11:59pm.

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	240 minutes	84.5 out of 100

Score for this quiz: **84.5** out of 100

Submitted Mar 24 at 12:05am

This attempt took 240 minutes.

Question 1**0 / 0 pts**

I understand that I must complete this test **on my own**.

The following activities are considered plagiarism/cheating and will result in a grade of 0 for the exam and possible disciplinary action:

- Discussing the test with anyone else, including students and tutors (in person or online)
- Communicating with others in person or online
- Posting code from the exam to study/question/discussion sites
- Submitting code that you did not write
- Copying code from a website or from another person

Note: You are allowed to discuss the test with the instructor, although there is no guarantee that she will be online while you are taking the exam.

If you have questions about this, stop immediately and contact Prof. Masters.

Correct!

-
- ☒ I understand and agree to all of the above.

Question 2**0 / 0 pts**

I understand that I must take the midterm exam in a single sitting- **all at once**.

If you have questions about this, stop immediately and contact Prof. Masters.


Correct!

-
- ☒ I understand

Section 1: Coding Questions

Provided Files

I've provided files for the coding questions. Use the tester program to see examples of how the code should work and to test your code.

[MidtermExamFiles.zip](https://ccsf.instructure.com/courses/47904/files/7254660/download?download_frd=1)  (https://ccsf.instructure.com/courses/47904/files/7254660/download?download_frd=1)

Pay close attention to:

- whether you are writing from the implementation perspective or client perspective
- what data structure you are asked to use
- whether efficiency (big-o) is part of the score

For all coding questions, for full credit:

- account for all possible times when the method could be invoked- your code should not crash or throw a runtime exception for things like empty and singleton datasets
- do **not** invoke the toArray() method
- do **not** use a data structure different from the one asked about in the question
- follow general best practice of coding (e.g., reduce duplicated code, follow naming conventions, etc.)
- ensure your answers are correct both for syntax (i.e., the code compiles) and semantics (i.e., the code accomplishes the task)

Write a complete method from the implementation perspective that is $O(n)$. This method will go inside the `LinkedBag` class. The method header is:

```
public void duplicateTheBag()
```

The method duplicates the contents of the bag. For example, if the bag contained (1, 2, 3), then after invoking the method, the bag would contain (1, 2, 3, 1, 2, 3) (in any order). See the tester program for more examples.

Notes:

- You **can** call other methods in `LinkedBag`.
- Having a linear $O(n)$ solution is worth 5 out of the 15 points.
 - Note: the driver program does **not** test for efficiency.

Question 3

14 / 15 pts

Paste your complete `duplicateTheBag()` method here.

Your Answer:

```
public void duplicateTheBag() {
    Node currentNode = firstNode;
    LinkedBag<T> tempBag = new LinkedBag<T>();

    // get the lastNode
    while (currentNode != null) {
        tempBag.add(currentNode.data);
        tempBag.add(currentNode.data);
        currentNode = currentNode.next;
    }

    this.firstNode = tempBag.firstNode;
}
```

-1 the code does not update numberOfEntries

Write a complete method from the client perspective that duplicates each element in place in a ListInterface object. For example, if the list contained (1, 2, 3), after invoking the method, the list would contain (1, 1, 2, 2, 3, 3). See the tester program for more examples.

Notes:

- For full credit, do **not** create another list (or any data structure) inside the method.
 - Only modify the parameter list.
 - Not creating another data structure is worth 10 out of the 20 points.
- This question will **not** be scored based on efficiency (big-O).

The method header is:

```
public static void addDuplicateNeighbors(ListInterface<String> list)
```

Question 4

13 / 20 pts

Paste your complete addDuplicateNeighbors(...) method here.

Your Answer:

```
public static void addDuplicateNeighbors(ListInterface<String> list) {  
    int size = list.getLength();  
  
    for (int i = 1; i <= size; i++) {  
        list.add(i, list.getEntry(i));  
    }  
}
```

-7 the code does not add the correct duplicates; iterate from 1 to size*2 and then increment i=i+2

Write a complete method from the implementation perspective. This method will go **inside** the AList class. The method header is:

```
public void addToFront(ANode<T> first)
```

The method takes in a linked chain of values and adds them **in order** to the **front** of the AList. See the tester program for examples. Pay close attention to the order of the chain elements!

For full credit, write a method that is $O(n)$.

Notes:

- The ANode class is provided. This class is exactly the **same** as a regular Node. I created a class with a different name so that it doesn't interfere with LinkedList or LList.
- To create a local variable inside the method, declare it as type ANode<T>.
- Having a linear solution is worth 3 of the 20 points.
 - Note: the driver program does **not** test for efficiency.

Question 5

17 / 20 pts

Paste your complete addToFront(...) method here.

Your Answer:

```
public void addToFront(ANode<T> first) {
    ANode<T> currentNode = first;

    int counter = 1;
    while (currentNode != null) {
        this.add(counter, currentNode.data);
        counter++;
        currentNode = currentNode.next;
    }
}
```

-3 repeatedly invoking `add(int,T)` makes this method $O(n^2)$ instead of $O(n)$; figure out the size of the chain that needs to be added and do a single shift of elements and then insert into the array

Section 2: Lists and Bags (From the Client Perspective)

Question 6

2 / 2 pts

`myBag` is an object of type **BagInterface**<String>. `myBag` currently contains the Strings:

snoopy

woodstock

charlie

lucy

What will be printed after the following is executed?

```
System.out.println(myBag.remove());
```

☐ charlie

☐ lucy

☐ snoopy

Correct!

- ☐ woodstock
- ☐ null
- ☒ you cannot tell from the provided code

Question 7**2 / 2 pts**

list is an object of type **ListInterface**<String> that contains the Strings in the order listed:

cat

dog

zebra

mouse

Which of the following statements will result in list containing:

cat

zebra

mouse

☐ none of these is correct

☐ list.remove(0);

☐ list.remove(4);

☒ list.remove(2);

☐ list.remove(3);

☐ list.remove(1);

Correct!

Question 8

2 / 2 pts

myList is an object of type **ListInterface**<String>. myList currently contains the Strings in the order listed:

boat

car

bike

plane

What will be printed after the following is executed?

```
System.out.println(myList.getEntry(1));
```

☐ none of these is correct

☐ null

☐ bike

☒ boat

☐ plane

☐ you cannot tell from the provided code

☐ car

Correct!

Question 9

2 / 2 pts

myList is an object of type **ListInterface**<String>. myList currently contains the Strings in the order listed:

delaware

california

iowa

montana

What will the list contain after the following is executed?

```
myList.add(2, "maine");
```

Correct!

- ☐ none of these is correct
- ☒ delaware maine california iowa montana
- ☐ maine delaware california iowa montana
- ☐ delaware california maine montana
- ☐ delaware maine iowa montana
- ☐ delaware california maine iowa montana

Question 10**2 / 2 pts**

myJavaList is an object of type **List**<String>. Note: this is the Java interface from the standard library. myJavaList currently contains the Strings in the order listed:

delaware

california

iowa

montana

What will the list contain after the following is executed?

```
myJavaList.add(2, "maine");
```

- ☐ delaware california iowa montana
- ☐ none of these is correct

Correct!

- ☐ delaware california iowa maine montana
- ☐ delaware california iowa montana maine
- ☐ maine delaware california iowa montana
- ☒ delaware california maine iowa montana
- ☐ delaware maine california iowa montana

Section 3: Array-Based Bags and Lists

Question 11

2 / 2 pts

The private data for an **ArrayBag** object called aBag is below:

numberOfEntries					
6					
bag[0]	bag[1]	bag[2]	bag[3]	bag[4]	bag[5]
17	31	29	42	16	58

What will be the contents of the ArrayBag object's array (called bag) after executing `aBag.remove(42);`

- ☐ 17 31 29 16 58 null
- ☐ none of these is correct
- ☐ null 31 29 17 16 58
- ☐ 17 31 29 null 16 58
- ☐ 31 29 17 16 58 null
- ☒ 17 31 29 58 16 null

Correct!**Question 12****2 / 2 pts**

The private data for an **AList** object called aList is below:

numberOfEntries						
5						
list[0]	list[1]	list[2]	list[3]	list[4]	list[5]	list[6]
	42	32	36	61	24	

What will be the contents of the AList object's array (called list) after executing `aList.add(2,92);`

- ☐ none of these is correct
- ☐ 92 42 32 36 61 24
- ☐ 42 32 36 92 61 24

Correct!☐ 42 32 92 36 61 24☒ 42 92 32 36 61 24**Question 13****0 / 1 pts**

In the AList implementation, when the list is empty, the list array is null (meaning `this.list==null`).

Un Answered☒ True**Correct Answer**☐ False**Question 14****1 / 1 pts**

In the AList implementation, when the list is empty, `list.length` is equal to 0.

☐ True**Correct!**☒ False**Question 15****1 / 1 pts**

In the AList implementation, when the list is empty, `numberOfEntries` is equal to 0.

Correct!☒ True☐ False

Section 4: Nodes and Linked-Based Implementations

Question 16

1 / 1 pts

In the LinkedBag implementation, when the bag is empty, firstNode is null (meaning firstNode==null).

Correct!☒ True☐ False

Question 17

1 / 1 pts

In the LinkedBag implementation, when the bag is empty, firstNode.data is null (meaning firstNode.data==null).

Correct!☐ True☒ False

Question 18**0 / 2 pts**

firstNode is the first node of the chain below.

14 -> 26 -> 37 -> 65

What will be output by passing firstNode to the following method?
(Ignore spacing in the output.)

```
public void mystery(Node first) {  
    Node current = first;  
    while(current.next!=null) {  
        System.out.print(current.data);  
        current = current.next;  
    }  
}
```

Correct Answer

- ☐ 14 26 37
- ☐ 14 26 37 65
- ☐ 14 26
- ☐ none of these is correct

You Answered

- ☒ the code will crash (an exception or error will be thrown)

Question 19**2 / 2 pts**

Refer to the following lists and code for the question below:

list1: 20 -> 37 -> 45
list2: 13 -> 29


```
public void mystery(Node nodeA, Node nodeB) {  
    nodeB.next = nodeA.next;  
    nodeA.next = nodeB;  
}
```

What will **list1** contain after calling `mystery(list1.firstNode, list2.firstNode);`?

Correct!

☐ 29 -> 20 -> 37 -> 45

☒ 20 -> 13 -> 37 -> 45

☐ 29 -> 37 -> 45

☐ 29 -> 20 -> 13 -> 37 -> 45

☐ 20 -> 13 -> 29

☐ 20 -> 37 -> 45

☐ 20 -> 29 -> 13 -> 37 -> 45

☐ 20 -> 29 -> 37 -> 45

☐ none of these is correct

☐ 13 -> 20 -> 29

Question 20

2 / 2 pts

firstNode is the first node of the chain below.

14 -> 42 -> 19

What is the value of `currentNode.data` after invoking the following code?

```
currentNode = firstNode;  
currentNode = currentNode.next;
```

- ☐ null
- ☐ 14
- ☐ none of these is correct
- ☐ 19
- ☒ 42

Correct!**Question 21****2 / 2 pts**

After the above code, what is the contents of the chain headed by firstNode?

- ☐ 14 -> 19
- ☐ 42
- ☒ 14 -> 42 -> 19
- ☐ none of these is correct
- ☐ 14
- ☐ 42 -> 19
- ☐ 14 -> 42
- ☐ 19

Correct!

Question 22**2 / 2 pts**

firstNode is the first node of the chain below.

48 -> 35 -> 16 -> 23

What is the value of currentNode.data after invoking the following code?

```
currentNode = firstNode.next;  
currentNode.next = currentNode.next.next;  
currentNode = currentNode.next;
```

Correct!

- ☒ 23
- ☐ 35
- ☐ 16
- ☐ 48
- ☐ none of these is correct
- ☐ null

Question 23**2 / 2 pts**

After the above code, what is the contents of the chain headed by firstNode?

- ☐ 16 -> 23
- ☐ 48 -> 35 -> 16 -> 23

Correct!

- ☒ 48 -> 35 -> 23
- ☐ 35 -> 16 -> 23
- ☐ none of these is correct
- ☐ 48 -> 35 -> 16
- ☐ 35 -> 16
- ☐ 48 -> 35

Section 5: Timing and Efficiency

Question 24

0 / 1 pts

An algorithm that is $O(2^n)$ is considered more efficient (better) than an algorithm that is $O(n^2)$.

You Answered☒ True**Correct Answer**☐ False

Question 25

3 / 3 pts

What is the order of growth of the following operations for **lists** implemented with **arrays**?

Correct!

add to the beginning of a list

O(n)



Correct!

add to the end of a list

O(1)



Correct!

delete from the beginning or middle of a list

O(n)



Correct!

delete from the end of a list

O(1)



Correct!

determine if the list contains an element

O(n)



Correct!

retrieve an element at a specific position

O(1)



Other Incorrect Match Options:

- none of these is correct
- $O(n^2)$

Question 26**3 / 3 pts**

What is the order of growth of the following operations for **lists** implemented with **linked nodes with a head (firstNode) pointer only**?

Note: there is no tail pointer and these are singly-linked nodes (not doubly linked nodes).

Correct!

add to the beginning of a list

O(1)



Correct!

add to the end of a list

O(n)



Correct!**delete from the beginning
of a list** $O(1)$ **Correct!****delete from the end of a
list** $O(n)$ **Correct!****determine if the list
contains an element** $O(n)$ **Correct!****retrieve an element at a
specific position** $O(n)$ 

Other Incorrect Match Options:

- none of these is correct
- $O(n^2)$

Question 27**0.5 / 1 pts**

What is the order of growth of the following operations for **lists** implemented with **linked nodes with a head (firstNode) and a tail (lastNode) pointer**?

Note: these are singly-linked nodes (not doubly linked nodes).

Correct!

add to the end of a list

$O(1)$



Not Answered

delete from the end of a list

$O(1)$



Correct Answer

$O(n)$

Other Incorrect Match Options:

- $O(n^2)$
- none of these is correct

Question 28

2 / 2 pts

Select the order of growth for the following algorithm. myArrayList is of type **ArrayList<Integer>**. Note: this is the Java class from the standard library.

```
int stop = myArrayList.size() / 2;
for(int i=0; i < stop; i++) {
    if(myArrayList.contains(i)) {
        System.out.println("Match found! " + i);
    }
}
```

☐ $O(n^3)$

Correct!

- ☐ none of the above
- ☐ $O(n \log n)$
- ☐ $O(\log n)$
- ☐ $O(n)$
- ☒ $O(n^2)$

Question 29**2 / 2 pts**

Select the order of growth for the following algorithm. `myArrayList` is of type **`ArrayList<Integer>`**. Note: this is the Java class from the standard library.

```
int stop = myArrayList.size();
for(int i=0; i < stop; i++) {
    System.out.println("Item: " + myArrayList.get(i));
}
```

Correct!

- ☐ $O(n^2)$
- ☐ $O(\log n)$
- ☐ none of the above
- ☐ $O(n \log n)$
- ☒ $O(n)$
- ☐ $O(n^3)$

Question 30**2 / 2 pts**

Select the order of growth for the following algorithm. myLList is of type **LList<Integer>**.

```
int stop = myLList.getLength();
for(int i=0; i < stop; i++) {
    System.out.println("Item: " + myLList.getEntry
(i));
}
```

☐ none of these is correct☐ $O(n^3)$ ☐ $O(n)$ ☒ $O(n^2)$ ☐ $O(\log n)$ ☐ $O(n \log n)$ **Correct!**Quiz Score: **84.5** out of 100