

# Programming Assignment 1

Please use this document as part of your submission for this assignment. Name the file following this convention: <Last Name><First Name>\_Assignment<num> as in DoeJohn\_Assignment1.

## Problem Statement:

This programming assignment is based on the Shipping Container problem described in the first lecture. You need to implement a data structure that supports storing shipping containers. Assume each shipping container has a unique id (e.g. “BOS0001”) that contains a three character code for the destination city followed by a numerical value.

The data structure should have a compact array where each position in the array represents a unique destination location. That array position will contain a reference to another array that contains references to each of the containers that are destined for that location.

Here is the class for the ShippingContainer:

```
public class ShippingContainer {  
    private String id;  
    public ShippingContainer(String newId){  
        this.id = newId;  
    }  
    public String getId() {  
        return id;  
    }  
}
```

Here is the interface for the interface to the container storage:

```
public interface IContainerStorage {  
    // got methods?  
}
```

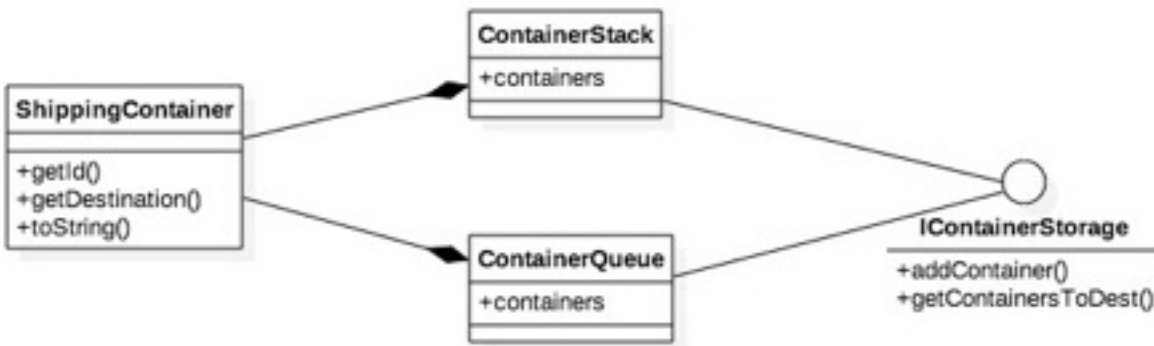
You will need to implement two variations of this solution. (Hint: two different classes implementing same IContainerStorage interface):

- A. Store in the order they are unloaded with the first one unloaded being the first one shipped out (Queue)

- B. Store in the order they are unloaded with the last one unloaded being the first one shipped out (Stack)

## 1. UML Diagram

Provide a class diagram of the classes you implement in this programming assignment. Please include your classes (including properties and methods) including class inheritance and class relationships.



## 2. Design the interface

The IContainerStorage and the classes that inherit from it will need a way to add containers to storage and get a list of all the containers going to a particular destination.

## 3. Source Code

Please paste your source code into this document and provide a zip file of your files as well.

See attached zip file. Too much code to place here.

## 4. Output

Store 50 containers in varying quantities going to at most 5 destinations. Then retrieve the containers for each destination and print out the container ids to show the order they are shipped.

**\*\*Testing stack implementation\*\***

Containers going to BOS

[BOS44, BOS42, BOS36, BOS32, BOS30, BOS26, BOS23, BOS22, BOS21, BOS19, BOS18, BOS11, BOS3]

Containers going to SF0

[SF049, SF045, SF041, SF029, SF028, SF024, SF016, SF012, SF05, SF00]

Containers going to LAX

[LAX47, LAX35, LAX33, LAX20, LAX17, LAX14, LAX9, LAX7, LAX2, LAX1]

Containers going to SJC

[SJC43, SJC40, SJC39, SJC31, SJC27, SJC25, SJC10, SJC8, SJC6, SJC4]

Containers going to IAD

[IAD48, IAD46, IAD38, IAD37, IAD34, IAD15, IAD13]

**\*\*Testing queue implementation\*\***

Containers going to BOS

[BOS9, BOS15, BOS16, BOS21, BOS23, BOS33, BOS34, BOS40, BOS43, BOS44, BOS45, BOS48]

Containers going to SFO

[SFO5, SFO10, SFO13, SFO14, SFO19, SFO20, SFO22, SFO24, SFO25, SFO29, SFO30, SFO32, SFO35, SFO39, SFO41]

Containers going to LAX

[LAX0, LAX1, LAX3, LAX4, LAX6, LAX7, LAX8, LAX11, LAX12, LAX28, LAX47, LAX49]

Containers going to SJC

[SJC2, SJC17, SJC31, SJC38]

Containers going to IAD

[IAD18, IAD26, IAD27, IAD36, IAD37, IAD42, IAD46]

## 5. Runtime Analysis

Identify the runtime analysis (Big O Notation) for adding and retrieving storage containers from your data structure for each of the variations.

Assuming constant lookup times for Java's implementation of a HashMap of LinkedLists and a HashMap of Stacks, ContainerQueue is  $O(1)$  because I can immediately insert all elements and return them at once. ContainerStack is  $O(n)$  because I must step through the existing stack and manually add each element to the new array.

## Evaluation

	D	C-	C+	B-	B+	A	Grade	
<b>1. Clarity</b>	Disorganized or hard-to-understand		Satisfactory but some parts of the submission are disorganized or hard to understand	Generally organized and clear	Very clear, organized and persuasive presentation of ideas and designs	Exceptionally clear, organized and persuasive presentation of ideas and designs		0.0
<b>2. Technical Soundness</b>	Little understanding of, or insight into material technically		Some understanding of material technically	Overall understanding of much material technically	Very good overall understanding of technical material, with some real depth	Excellent, deep understanding of technical material and its inter-relationships		0.0
<b>3. Thoroughness &amp; Coverage</b>	Hardly covers any of the major relevant issues		Covers some of the major relevant issues	Reasonable coverage of the major relevant areas	Thorough coverage of almost all of the major relevant issues	Exceptionally thorough coverage of all major relevant issues		0.0
<b>4. Relevance</b>	Mostly unfocused	Focus is off topic or on insubstantial or secondary issues	Only some of the content is meaningful and on topic	Most or all of the content is reasonably meaningful and on-topic	All of the content is reasonably meaningful and on-topic	All of the content is entirely relevant and meaningful		0.0
						Assignment Grade:		0.0