

**<22/SP-COP-2800-72035> Java Advanced**

**<Assignment 13-08>**

Document Version: 0.1

Version Date: July 17, 2022

Created By: David Duron

# Document Version Control

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Rationale |
| 0.1 | 2022 JUL 17 | David Duron | Submit Assignment |

# Document Purpose

The purpose of this document is to define the MyStack class and discuss how to implement and use it.

# Technical Specifications

## Purpose of Technical Implementation

The purpose of the MyStack class is create an object that copies information that was used as a reference to create an array, and copies the array. The use case for this class is that each object will have their own reference in the event the source material changes for the original reference; this means that the MyStack instance will not have its reference material changed by mistake. The class is constructed with no arguments and has one property known as list. The class has a variety of methods that are used to perform the “Deep Copy” procedure that we are trying to achieve.

## Technical Implementation Components

I supplemented the MyStack class. The MyStack class begins by creating an object with zero arguments. From there, we can use the push method to add elements to our list property; after that, all methods are available to be used to ensure the object meets the expectations of the developer such as: toString, copyList, clone, peek, getSize, isEmpty.

**Properties**

1. list: this stores ArrayList object

**Methods**

1. isEmpty(): a Boolean to see if any elements exist within the list property
2. getSize(): displays the amount of elements plus 1 (because zero is counted).
3. Peek(): displays the last element of the list property.
4. Pop(): removes the last element of the list property.
5. Push(object a): adds the object passed as an argument to the last element of the list property.
6. Clone(): here, we throw “CloneNotSupportedException” which allows use to use methods from the Java Cloneable super class. We create a new object instance of MyStack class and call its .clone() method from the super class. From there, we use the object and call its .copyList() method.
7. copyList(): we create a new ArrayList object, use a for loop to iterate through all elements of the this.list and add the elements to the newly created ArrayList object.
8. toString(): display the amount of elements are inside of the object.

**Constructors**

The developer can create an instance of the MyStack class one way

1. MyStack MyStack\_variable = new MyStack()

# Technical Implementation Pseudocode

1. Create MyStack MyStack\_variable = new MyStack()
2. Add elements to the object’s ArrayList property using the push() method
3. Clone the first stack into a second object instance using the clone() method
4. Use the pop() method to remove the most recently added element.
5. end

End