National University of San Agustin
System Engineer School

TO - Lab 01 - Interfaces, Abstraction and Encapsulation

Christian E. Portugal-Zambrano

September 11, 2018

1 Interfaces vs Abstraction vs Encapsulation

The concept of interfaces vary from text to text but all of them try to explain that this is a way to describe the behavior of class without a particular implementation.
As we are working with C++ this kind of interfaces are represented with abstract classes, but we must keep in mind to always separate the interface from the implementation (data abstraction), so in this way a change in the implementation must keep intact the interface, all of this is achieved through encapsulation (exposing just the necessary interfaces).

2 Pre-requisites

For this practice you must have the implementation of a Simple Linked List and make some review of abstract class and virtual functions.

3 List Interface

Here I present to you a preliminar List Interface, we need to keep in mind that this interface would represent more than one kind of List e.g Double Linked List, Circular List, and may be the implementation of each one is different e.g array based and pointer based.

```
#ifndef ILIST_H_
#define ILIST_H_

#include "CNode.h"

template <class T>
class IList{
    public:
        virtual  void insert(CNode<T>* _node) = 0;
        virtual ~CLinkedStructures() = 0;
};
```

First, we need to integrate our previously List with this interface and respond this questions:

- What virtual keyword do?
- What virtual <> = 0 means?
- A destructor must be virtual?
- What is the behavior of the constructor when the List is integrated into the IList interface?
- All the method we have in List must be integrated into IList? Why? Explain.

4 Warm up

Good, you have integrated your List into the interface IList, now make the design of all the List structures you know and integrate with IList. Have you seen some similar between all of them? I recommend to you implement Array and Pointer based Simple, Circular and Double List.

5 To do

Implement other structures like Stacks, Queues and Trees, try to integrate into List interface, after that respond these questions:

- What is the order of calls to constructors?
- Could you correctly integrated Tree Structure into IList interface? Explain.
- Make an UML design to explain the integration.
- Make some examples about how to use Polymorphism with these classes.

6 Deep inside

This section is just for anyone who wants to make a deep inside into the theory and like challenges[1], you will have to prepare a presentation of just 5 minutes to explain and run the code, then you will have to defend yourself another five minutes of questions. I want that all students benefit from your presentation, remember that we are here to learn. If you want to do this please email to cportugalz@unsa.edu.pe

7 Deadline

This report will be qualified in the next class through the presentation of all the code required by this practice. Remember that plagiarism will be punished. All question and doubts must be done to the email.

[1] This must not be included into the report