

## Programming Exercises 01

### List Implementation (Linked List)

#### OBJECTIVE

To be able to implement the list ADT using linked list.

#### INSTRUCTIONS

1. You are to work on this activity individually or by pair. There should only be one submission per person or per pair.
2. You are to write a code for a program that follows the details in the PROGRAM SPECIFICATION section. There should only be one program for this PE.
3. You are to develop your program following the structured programming approach at the very least. **No global variable declarations are allowed.**
4. All your program files (source code files, input files, etc.) should be together with your main program file (the one that contains the main function). There should be no need to put certain files in certain folders just so your program will compile and run.
5. You only need to submit the source code file you have created. If you created more than one source code file (i.e. source code is broken down into several files) or there are other files (e.g. input files) involved, archive them in a single zip file. Name your source code file or zip file, whichever is applicable, using your surname followed by the PE # similar to this example: Rizal\_02 (for individual) or Bonifacio\_Rizal\_02 (for pair/group).

#### PROGRAM SPECIFICATION

Your program will have the following main menu:

*List MENU*

*[1] Add Item*  
*[2] Delete Item*  
*[3] Display List*  
*[4] Delete List*  
*[0] Exit Program*

*Enter choice:*

Details of the menu items are as follows:

##### **Add Item**

Once selected, the program should ask the user for an item to be added. Then, it should ask where to insert the item. Similar to as follows:

*Please enter the item to be added:*

*Where to add the item?*

- [1] Start of the list*
- [2] End of the list*
- [3] Specific position*
- [0] Cancel*

*Enter choice:*

Each menu item under *Add Item* are detailed as follows:

- *Start of the list*: add the element to the start of the list.
- *End of the list*: add the element to the end of the list.
- *Specific position*: ask the user for the position where to add the element. If the position exists, insert the element to that position. Otherwise, add the element to the end of the list.

### ***Delete Item***

Once selected, the program should ask the user for an item to be deleted. If the list does not contain the item, the program should display a message similar to as follows:

"The list contains no such element"

Else, the program should delete the first occurrence of the item in the list and display a message similar to as follows:

"The element \_\_, which is at position \_\_, has been deleted"

### ***Display List***

Once selected, the program should display the content of the list. Your display should also indicate the positions of the elements.

### ***Delete List***

Once selected, the program should empty the list and display a message similar to as follows:

"The list has been emptied!"

### ***Exit Program***

Once selected, the program should terminate.

Note: After processing the selected main menu item, the program should loop back to the main menu. The program should end only when the user selects the main menu item ***Exit***.