Code Design and Data Structures

|  |  |
| --- | --- |
| **Assessment Task Number:** Part 2 – Implement a Double Linked List | |
| **Unit Code(s):** | **Unit Title(s):** |
| ICTPRG547 | Apply advanced programming skills in another language |
| CUADIG512 | Design digital applications |
| **Instructions to Learners:** | |

For this task, you must create a double-linked list class and write a program that demonstrates its use. You must also write a sorting algorithm that will sort your list.

**Requirements:**

You are tasked with creating a Double-Linked List class. The class must support the following operations:

* Inserting and deleting a node at the front of the list
* Inserting and deleting a node at the end of the list
* Inserting and deleting a node at an arbitrary location in the list
* Returning a count of how many nodes are in the list
* Checking if the list is empty
* Returning the first or last node in the list
* Sorting the list (as appropriate)

**Implementation:**

There are two ways you can demonstrate the implementation of your double-linked list.

1. Implement an object pool within the CDDS\_Optimise program found within the AIE Student Samples solution available on GitHub <https://github.com/AcademyOfInteractiveEntertainment/AIEYear1Samples>   
     
   For this application, sorting the list may not be necessary, depending on your object pool implementation.

**OR**

1. Create a small, stand-alone application to demonstrate and test your linked list class. Your test application need not be a game, but it must allow the user to verify that the double-linked list class works without inspecting the code (an application containing a graphic user interface created using a third-party library like RayLib is recommended).  
     
   For this implementation, you will need to demonstrate sorting on the list.

|  |  |  |
| --- | --- | --- |
| **Task** | | **Evidence Criteria** |
| 1. | Double-Linked List | Write a custom implementation of a double-linked list and demonstrate its use in a test application. |
| **Submission Requirements:** | | |
| You will need to submit the following:   * A Release build of each application that can execute as a stand-alone program * Your complete Visual Studio project   Be sure to remove any temporary build folders (i.e., the Debug and Release folders). Only project files, source code files, and any resource files used should be included in your submission.  Package all files in a single compressed archive file (.zip, .7z, or .rar) | | |