**Data Structures**

**Assignment #2 (To be done in groups of 3)**

**Due Date: 5th April 2019 midnight**

The attached solution “**WeaponShopAssign2.zip**” contains basic working code for adding weapons to an in game collection. The weapons are presently stored using a simple hash table.

You are required to modify the C# “**WeaponShopAssign2** “solution to implement the storage of the weapons using a Binary Search Tree. You must also extend the project to allow the player to have a backpack (where all the items bought will be placed).

Part 1) (worth 65% of the marks)

You must implement the ability to perform the following operations

* Insertion
* Search
* In-order Traversal (for printing)

The shop view must be updated to show the number of items in stock next to each item.

**NOTE: The Binary tree is to be ordered by Weapon Name.**

Part 2) (worth 10% of the marks)

You must allow the ability to Delete Weapons from the shop which requires the following operation

* Deletion

Part 3) (worth 15% of the marks)

You must modify the Player class to have a Backpack object (instead of an array of weapons).

You must implement the Backpack class as a linked list of weapons. The linked list is to be implemented in a way such that new weapons are added at the end of the list.

When viewing the Player, the items in the backpack must also be shown.

Part 4) (worth 5 % of the marks)

After creating the player, you must present a menu with functionality similar to what is show below:

1. Add Items to the shop
2. Delete Items from the shop
3. Buy from the Shop
4. View backpack
5. View Player
6. Exit

Submission Requirements:

**The Student ID number for each member of the group must be commented at the top of the file.**

**Choose one member of the group to make submissions on behalf of the group.**

**The chosen member can make as many submissions (versions) as they like. Only the last submission will be marked.**

**You must upload 2 items:**

**The completed submission document (NOT ZIPPED)**

**FULLY modified C# project as a zip file.**