**Inventory/Player class (WIP)**

**Develop the following:**

**1.** Create a template class called *Inventory* that stores an array of *T* items and set the maximum size as you please. Make use of *std::array* instead of the old C-style array.

**2.** Add a function called *Add()* that takes in the item to add as well as which index number to use when indexing the array.

**3.** Overload the **[]** operator such that it allows the user to index an element of the array either to get or set that particular value.

**4.** Add a function called *DisplayItems()* that iterates through the array and outputs the values to the console using *std::cout*. Note that this will only work with simple data types such as *int*, *float*, or even *std::string*.

**5.** Use *assert()* where you feel it necessary to avoid invalid values being used.

**1.** Create a class called *Inventory* that will encapsulate the properties and behaviour of an inventory item in a game. Add member variables such as *name*, *value*, etc.

**2.** Overload the **<** and **>** operators such that two *Inventory* items can be compared. Within these functions compare the *values* of the inventory items and return *true* or *false* accordingly.

**3.** Create a class called *Player* that will store a *std::list* of *Inventory* items.

**4.** Add a member function that will allow an inventory item to be added to the player.

**5.** Add another member function that will display all inventory items on screen. Before display, make sure to *sort* the items in the player’s inventory list. Use a *lambda* to perform the comparison and try to sort in *ascending* as well as *descending* order.

**6.** Add a *IsDead()* predicate function to the *Inventory* class that will return an *isDead* flag, which will state if that particular inventory item is to be removed from the player or not.

**7.** Set some of the inventory items’ *isDead* flag to *true* and then find and remove all of the dead items from the player’s list. Use a lambda here as well.

Create a *std::list* of *std::string* objects that will store a variety of inventory items.   
**2.** Create a loop that runs and asks the user to add items to the player's inventory. Store the user's desired item in the list, anywhere you like. The user should type *exit* to break out of the loop.   
**3.** For each iteration of the loop, clear the console screen, display the player's current items and prompt the user to enter more items. Make sure that items with a vowel are listed with *An* and the consonant items with *A*. For example, *An apple* or *A sword*.   
**4.** If the player is carrying nothing, the console should display something along the lines of *Our hero is carrying nothing at all*.   
**5.** Ask the user which item they want to remove and take that item out from the list.   
**6.** Use the *find()* algorithm to search for a particular item and remove that.