**Assignment 4 – Inheritance and Polymorphism**

Submit to MUOnline as a compressed (.zip) file containing your code project.

|  |  |  |
| --- | --- | --- |
| 1. | Ensure that your classes are each in their own files, the member variables are inaccessible except by public functions, and you apply the keywords **const** and **override** to functions and function parameters where appropriate. | 20% |
|  |  |  |
| 2. | Create the following class hierarchy:  A class Item with member variable **string name**, and a virtual member function **void Use()**. When Use() is called, it should display something like “Using Item X.” This could represent any item in a video game.    A class Equipment which inherits Item and has a member **bool equipped**. The class should override Item::Use() to either equip the item if it is unequipped, or unequip it if it is equipped, indicated by some output. This would be like a weapon you could equip in a game.  A class Consumable which inherits Item and has a member **int uses** and a member function **bool isEmpty()**. The class should also override Item::Use() and it should call isEmpty() to check if there are uses left, then either indicate that it is empty or reduce ‘uses’ by 1 and call the base class’s Use() if it is not empty. This would be like a health potion you can use x times in a game. | 40% |
| 3. | In main(), create an array of **Item** **Pointers** called ‘inventory’. Add instances of all 3 classes into the array, and then loop over them to display the results of each object’s Use() at least enough times to show running out of a Consumable item and unequipping an equipped Equipment item. The same line of code like  inventory[i]->Use(); should be enough to call the 3 different Use() functions (1 in each class) if you setup Polymorphism correctly. | 40% |