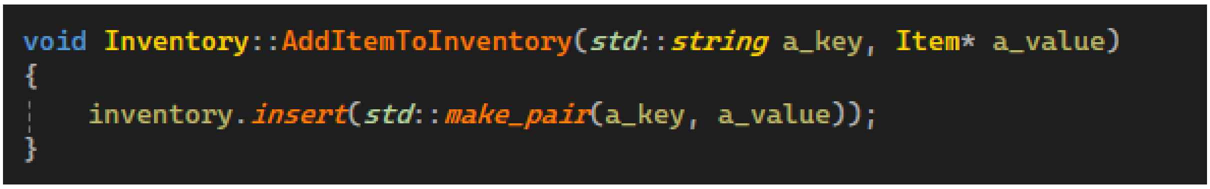
Inventory Report
Liam Whitehouse - 103862481

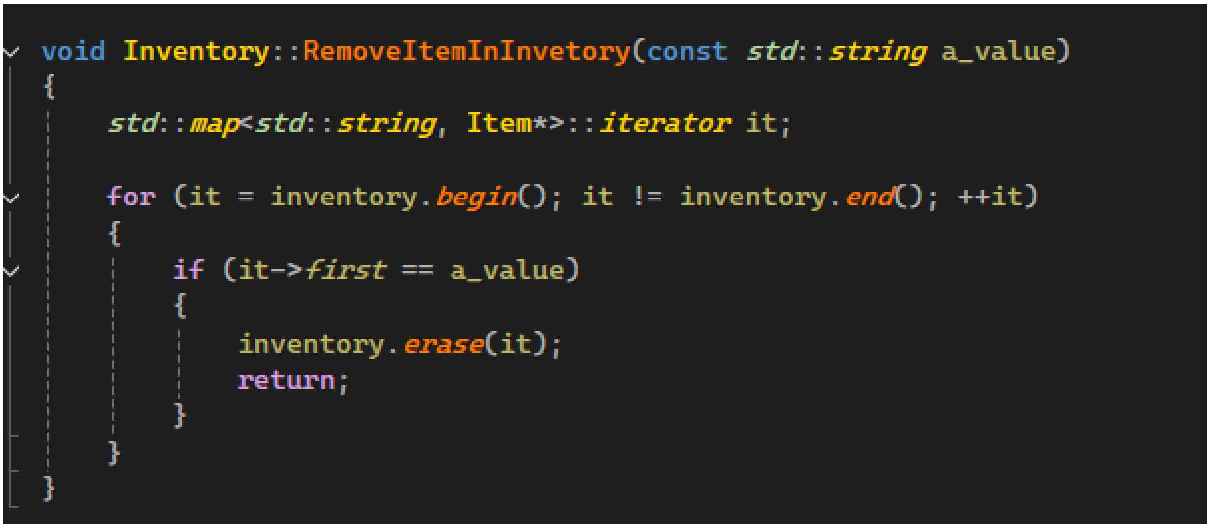
For the games inventory I have decide to use a Hashmap. This is due to the Key and Value look-up, due to the game being a text-based adventure, it made sense to easily search for their values rather than looping through an entire array from start to finish.

Firstly, we have the AddItemToInventoryFunction which as the name implies, adds a new Item to the Inventory.



```
void Inventory::AddItemToInventory(std::string a_key, Item* a_value)
{
    inventory.insert(std::make_pair(a_key, a_value));
}
```

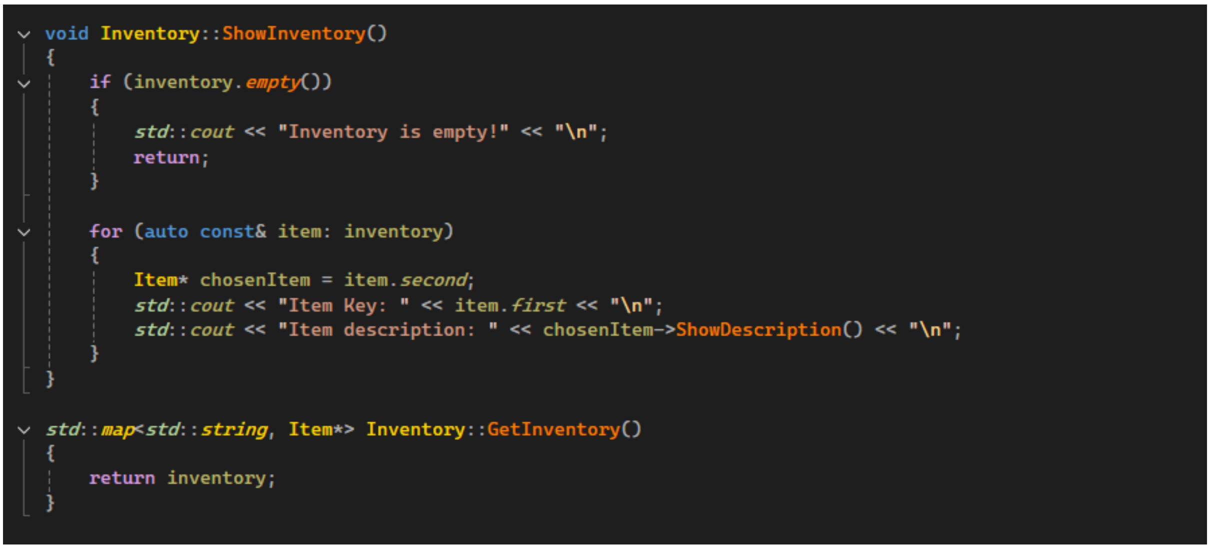
Next, we have removing an item from the Inventory.



```
void Inventory::RemoveItemInInvetory(const std::string a_value)
{
    std::map<std::string, Item*>::iterator it;

    for (it = inventory.begin(); it != inventory.end(); ++it)
    {
        if (it->first == a_value)
        {
            inventory.erase(it);
            return;
        }
    }
}
```

Then our final function is to show the current inventory as well as returning the entire inventory.



```cpp
void Inventory::ShowInventory()
{
    if (inventory.empty())
    {
        std::cout << "Inventory is empty!" << "\n";
        return;
    }

    for (auto const& item: inventory)
    {
        Item* chosenItem = item.second;
        std::cout << "Item Key: " << item.first << "\n";
        std::cout << "Item description: " << chosenItem->ShowDescription() << "\n";
    }
}

std::map<std::string, Item*> Inventory::GetInventory()
{
    return inventory;
}
```