

EN3085 – Computing Lab 2

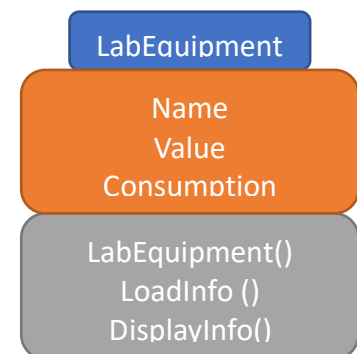
In this lab, the aim is to develop a code that would answer the questions from lab 1 but using object oriented concepts.

Question 1 - Using a vector with a user defined class

Use the C++ code provided (see finale page). This already answer the question 1 from lab 1 but this time using a class to collect equipment information.

In the main.cpp file, it also stores the equipment information in an array. An issue with the use of array is that it blocks 100 pieces of memory. Which is a waste of memory.

Modify the programme so instead of an array it uses a vector structure.



Question 2 – Creating a class LabEquipmentList

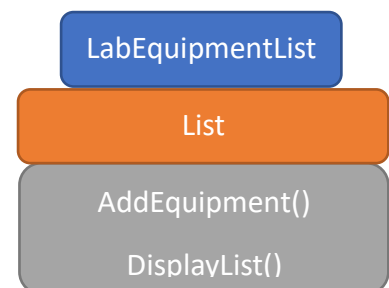
Add to the provided code a class called **LabEquipmentList** which has:

As attributes:

- **List**: a vector of objects **LabEquipment**

As methods:

- **AddEquipment()** - A member function that keeps asking on the screen if the user wants to add an equipment and if so that:
 - Collects the equipment information using an object of class **LabEquipment**
 - Stores the object in the class attribute **List**
- **DisplayList()** - A member function that:
 - Displays the number of equipment stored in the class attribute **List**.
 - Displays line by line for each piece of equipment stored in in the class attribute **List**, its name, value and daily power consumption.



Demonstrate it's use by implementing the following main file (or similar). This should perform the same task as question 2 in lab 1, so you can use the solution provided for lab 1 or your own code.

```
int main() {
    LabEquipmentList Mylab;

    Mylab.AddEquipment();
    Mylab.DisplayList();

    system("pause");
    return 0;
}
```

Question 3 – Member functions that saves in a file:

- Add a method (member function) to the class **LabEquipment** that saves the equipment information into a text file.

```
void LabEquipment::SaveToFile(ofstream& OutputFile);
```

- Demonstrate its use with the following code used in the main.cpp file

```
LabEquipment Equipment1;  
ofstream OutputFile("EquipmentList.txt");  
  
Equipment1.LoadInfo();  
Equipment1.SaveToFile(OutputFile);  
OutputFile.close();
```

- Add a method (member function) to the class **LabEquipmentList** that saves the information of all the recorded equipment into a text file:

```
void LabEquipmentList::SaveToFile(string FileName);
```

Demonstrate its use with the following code in the main file (making sure all required library are declared). This should perform the same task as question 3 in lab 1,

```
LabEquipmentList Mylab;  
  
Mylab.AddEquipment();  
Mylab.DisplayList();  
Mylab.SaveToFile("EquipmentList.txt");
```

Question 4 – Member Functions that reads from a file

Add a method to the class **LabEquipment** that modify the attributes values directly, without asking the user but directly through variables as follows:

```
void LabEquipment::SetInfo(string NewName, double NewValue, double NewConsumption);
```

Demonstrate it use with the following code in the main file

```
LabEquipment Equipment1;  
Equipment1.SetInfo("DeskLight", 12, 6);
```

Add a method to the class **LabEquipmentList** that loads the information stored on an existing file and return the number of loaded equipment:

```
unsigned int LabEquipmentList::GetListFromFile(string FileName)
```

Demonstrate its use with the following code in the main file. This should perform the same task as question 4 in lab 1,

```
LabEquipmentList Mylab;  
unsigned int NbEquipment;
```

```
NbEquipment = Mylab.GetListFromFile("EquipmentList.txt");  
cout << NbEquipment << " existing equipment loaded." << endl;  
if (NbEquipment !=0) Mylab.DisplayList();  
Mylab.AddEquipment();  
Mylab.DisplayList();  
Mylab.SaveToFile("EquipmentList.txt");  
Run the program multiple time to check the proper reading and writing in the file.
```

LabEquipment

Name
Value
Consumption

LabEquipment()
LoadInfo ()
DisplayInfo()
SaveToFile(ofstream)

LabEquipmentList

List

AddEquipment()
DisplayList()
SaveToFile(FileName)

LabEquipment

Name
Value
Consumption

LabEquipment()
LoadInfo ()
DisplayInfo()
SaveToFile(ofstream)
SetInfo(Values)

LabEquipmentList

List

AddEquipment()
DisplayList()
SaveToFile(FileName)

Files Provided

LabEquipment.h

```
#ifndef LabEquipment_H
#define LabEquipment_H

#include <string>
using namespace std;

class LabEquipment {
    string Name;
    double Value;
    double Consumption;
public:
    LabEquipment();
    void LoadInfo();
    void DisplayInfo();
};
#endif
```

LabEquipment.cpp

```
#include "LabEquipment.h"
#include <iostream>
using namespace std;

void LabEquipment::LoadInfo() {
    double wattage, hours;
    cout << "What is the name of the new equipment? ";
    cin >> Name;
    cout << "What is its current value? ";
    cin >> Value;
    cout << "What is its wattage? ";
    cin >> wattage;
    cout << "What is its daily usage in hours? ";
    cin >> hours;
    Consumption = wattage * hours / 1000;
}

void LabEquipment::DisplayInfo() {
    cout << Name << ", " << Value << " Pounds, " << Consumption << "KWh" << endl;
}

LabEquipment::LabEquipment() {
    Name = "";
    Value = 0;
    Consumption = 0;
}
```

main.cpp

```
#include <iostream>
#include "LabEquipment.h"
using namespace std;

int main() {
    LabEquipment EquipmentList[100];
    LabEquipment Equipment;
    Equipment.LoadInfo();
    EquipmentList[0] = Equipment;
    cout << "The following equipment has been added to the list:" << endl;
    EquipmentList[0].DisplayInfo();
    system("pause");
    return 0;
}
```