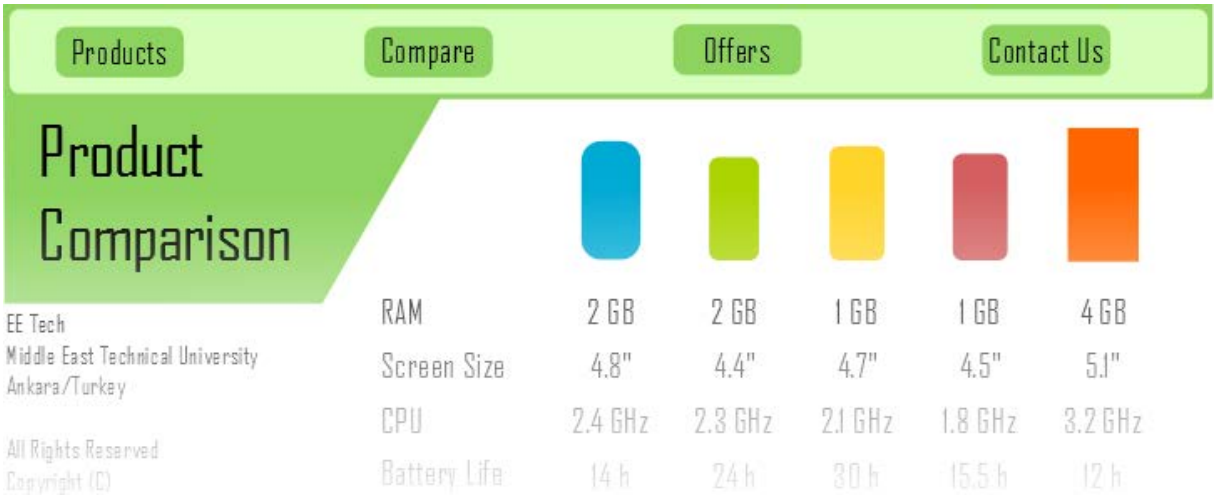
EE 441 HW#1 **CompareTech!**

Due Date: November 10, 2014 (23:59)

TAs:	Yeti Ziya Gürbüz (C)	ARC-202	yeti@eee.metu.edu.tr
	Tayfun Eylen	EA-404	eylen@metu.edu.tr
	Emin Zerman *	ARC-202 / DB-14	zerman@metu.edu.tr

*** You may ask HW#1 related questions to zerman@metu.edu.tr.**



1. Introduction

EE Tech is a company selling electronic devices over the Internet. Assume that you have just graduated from university and found a job on EE Tech. The company aims to make their website more user-friendly. You may also think that the graphical design of the website (seen above) is not the best but let's leave it to graphical designers. Your boss asks you to create a comparison algorithm in C++. The challenge about this task is that all the properties (attributes) of the products have been written in simple HTML language by hand (i.e. there isn't any class structure that you can use). So, you should first create a simple class structure for the products of the company. You should then construct a comparison algorithm based on your class structure. You are also required to create an interface that can communicate with your software via your company's website.

2. Requirements

a. Create a class structure to be used to hold all the properties of the electronic devices that your company sells. Implement some methods that will help you in sorting and comparison tasks. Required properties and associated methods will be explained in detail in the following section.

b. Create a simple User Interface (UI) so that your website can communicate with your software and get the results of comparisons. The UI will also be helpful in testing and debugging your software.

c. Write a program that will process a txt file that includes information about the electronic devices your company sells.

d. Integrate all the parts above and test the functionality of the software.

3. Homework

Three different parts in this homework are explained in detail below:

a. Implementation of the Class Structure

You need to implement a class structure that holds information about a product. EE Tech is selling only electronic devices. Name your class as *ElectronicProduct*. Assume that you only need the following properties in our class (classify private and public ones yourself):

int productID
string productName
string *productType* – (can have values smartPhone, laptopComp, gameConsole, and tv)
float *deviceCpu* (in GHz)
int *deviceRam* (in GB)
int *deviceMemory* (in GB)
float *screenSize* (in inches)
float *batteryLife* (in hours)
bool *has4g*
bool *hasBluetooth*
bool *hasWifi*
bool *hasHdmi*
bool *hasUsb*
bool *hasFrontCamera*
string *opticalDevice* – (can have values bluray, cd, dvd)
float *price*
float *deviceBenefit*
float priceOverPerformance

A device may not have all have properties at all. In order to find the properties of a device, *findRelevantProperties()* method should be used. This function may return a string or an integer array according to your choice and should work as illustrated in the following table :

productType	**smartPhone**	**laptopComp**	**gameConsole**	**tv**
productID, productName	✓	✓	✓	✓
deviceCpu	✓	✓	✓	
deviceRam	✓	✓	✓	
deviceMemory	✓	✓	✓	
screenSize	✓	✓		✓
batteryLife	✓	✓		
has4g	✓			
hasBluetooth	✓	✓	✓	
hasWifi			✓	✓
hasHdmi		✓		✓
hasUsb			✓	✓
hasFrontCamera	✓			
opticalDevice		✓	✓	
Price	✓	✓	✓	✓
deviceBenefit	✓	✓	✓	✓
priceOverPerformance	✓	✓	✓	✓

Do not forget to initialize properties that are not relevant to a device in order to create an object without an error.

You also need three other functions *findDeviceBenefit()* to find the performance of the device, *findBenefitOfOpticalDevice()* to find what optical properties the device has (which simply returns 1 for CD, 2 for DVD, and 3 for Bluray), and *calcPriceOverPerformance()* to calculate and update the price-performance ratio. The performance of the device is to be computed as:

$$deviceBenefit = (variables\ with\ \text{has}\ prefixes\ and\ optical\ device) * other\ variables$$

For example, *deviceBenefit* for *gameConsole* type can be computed as:

$$deviceBenefit = (hasBluetooth + hasWifi + hasUsb + findBenefitOfOpticalDevice()) * deviceCpu * deviceRam * deviceMemory$$

Please notice that different types of devices require different computations for performance. Therefore, you should implement appropriate *get* functions (i.e. *getDeviceCpu()*, *getHas4g()*, etc.) in order to get the appropriate variable values. You also will need to set these values while creating a product database, so it would be helpful to write both *get* and *set* methods for the variables.

b. Creation of User Interface

In order to test and debug your program you need to create a simple user interface. The required UI may look like the following:

```
--------------------------------
EE Tech Product Comparison
-- Your Name -- Student ID --
Enter your choice:
   1- Enter new product
   2- Import all product data from TXT
   3- Display all products
   4- Choose 3 products to compare
   0- Exit program
>
```

You may simply use the following pseudo-code:

```
void main()
        create an ElectronicDevice object or pointer to ElectronicDevice object array
        while true
                welcome user and request input
                if userInput is 1
                        enterNewProduct()       //Which simply creates a new object
                        displayObject()
                if userInput is 2
                        objectArray = parseTxtFile (filename, array)
                        notify user by printing mesage "done"
                if userInput is 3
                        displayAllObjects()
                if userInput is 4
                        ask three different product IDs
                        check if products are of same kind
                        calculate priceOverPerformance and
                if userInput is 0
                        print farewell and exit
        end while
end main
```

Test your own software using different test case scenarios in order to achieve a fully functional program before submitting your homework.

c. Text Processing

Assume that a company intern has already created a txt file that includes all the properties of the products. You need to get this information from the txt file to your program by utilizing appropriate C++ functions. In other words, you should **"parse"** this text file before proceeding to the next step. You

are expected to learn file parsing using C++ by yourselves. You may find file parsing examples on the Internet. You may use the following pseudo code as a start:

```
void parseTxtFile (filename, array)
        open the file by filename
        while until the end of file
                read lines
                extract useful information
        end
        return the useful information in array
```

Once you complete the task of extracting the required information, you should write it in an array or multiple array structures. Input txt file should have "allProductData.txt" as its filename and this txt file should be placed in the same folder with your "main.cpp" file. An example of such a txt file is provided on the ODTUClass course web page along with this homework. Please, copy this file into your own project folder.

Please be aware that <u>your homework may be tested using different content files during grading</u> (content may change but filename will always be "allProductData.txt").

4. Examples

The content of "allProductData.txt" looks like the following:

```
EE Tech All Product Data
============================================================
Name:   Xperience One
Type:   Smartphone
CPU:    3.2 GHz
RAM:    4 GB
Screen: 4.5 "
Mem:    32 GB
hasBT:  True
has4g:  False
battLife: 24 h
                ...
                ...
===
Name:   Atlas T2000
Type:   LaptopComputer
CPU:    3.4 GHz
RAM:    8 GB
Screen: 15.4 "
                ...
===
Name:   FunStation 3.2
Type:   GameConsole
CPU:    2.8 GHz
RAM:    8 GB
hasBT:  True

                ...
============================================================
```

<u>IMPORTANT:</u> Please notice that not all products have the same properties, so pay attention to the relevant properties for each device type while parsing the txt file.

An example of outputs may be as follows. These outputs are given only to indicate the required formatting of the output and do not reflect actual results:

```
--------------------------------
EE Tech Product Comparison
-- Ali Yazar -- 1551234 --
Enter your choice:
    1- Enter new product
    2- Import all product data from TXT
    3- Display all products
    4- Choose 3 products to compare
    0- Exit program
>1
Enter Product Name:
>GamerEngine 5
Enter Product Type:
>GameConsole
Enter CPU (in GHz):
>3.5
Enter RAM (in GB):
>4
                    ...
                    ...
Enter Price (in TL):
>459
Product is saved. Details are below:
--------
PID:    1
Name:   GamerEngine 5
Type:   GameConsole
CPU:    3.5 GHz
RAM:    4 GB
                    ...
                    ...
Price:  459 TL
--------
--------------------------------
EE Tech Product Comparison
-- Ali Yazar -- 1551234 --
Enter your choice:
    1- Enter new product
    2- Import all product data from TXT
    3- Display all products
    4- Choose 3 products to compare
    0- Exit program
>2
All products are successfully imported!
--------------------------------
EE Tech Product Comparison
-- Ali Yazar -- 1551234 --
Enter your choice:
    1- Enter new product
    2- Import all product data from TXT
    3- Display all products
    4- Choose 3 products to compare
    0- Exit program
>3
-----------------------------------------------
PID     Type            Name
-----------------------------------------------
1       Smartphone      Xperience One
2       Smartphone      myPhone 8
3       Smartphone      A3
4       Smartphone      G-5000
5       LaptopComp      Atlas T2000
                ...
                ...
13      GameConsole     FunStation 3.2
-----------------------------------------------
--------------------------------
EE Tech Product Comparison
-- Ali Yazar -- 1551234 --
Enter your choice:
    1- Enter new product
    2- Import all product data from TXT
    3- Display all products
    4- Choose 3 products to compare
    0- Exit program
>4
Choose three PIDs:
>1
>2
>4
You chose 1, 2, and 4. The comparison is below:
-------------------------------------------------------
        Xperience One   myPhone 8       A3
-------------------------------------------------------
CPU:    2.8 GHz         3.2 GHz         3.2 GHz
RAM:    3 GB            4 GB            4 GB
Screen: 4.5"            4.7"            5.2"
                ...
                ...
Price:  650 TL          800 TL          700 TL
-------------------------------------------------------
Benf:   54              62              75
Prc/Pf: 12.03           12.90           9.33
-------------------------------------------------------
Best choice is "A3".
--------------------------------
EE Tech Product Comparison
-- Ali Yazar -- 1551234 --
Enter your choice:
    1- Enter new product
    2- Import all product data from TXT
```

3- Display all products
4- Choose 3 products to compare
0- Exit program
>4
Choose three PIDs:
>1
>2
>5
You chose 1, 2, and 5. Invalid operation. Please choose all products of same kind.

EE Tech Product Comparison
-- Ali Yazar -- 1551234 --
Enter your choice:
1- Enter new product
2- Import all product data from TXT
3- Display all products
4- Choose 3 products to compare
0- Exit program
>0
Thank you for using. Program will exit now!

---

5. Notice

As an engineering candidate, each student is expected to submit a fully functional homework. In business life, no one would pay for an ill-working program. So, please check your work and debug your program using different inputs and examine corresponding outputs carefully in order to expose any unpredicted or hidden errors in your code. In addition, delivering a product/program with an undesired format harm your reputation in both academic and business life.

You are required to add comments to your code for your colleagues to understand. You may not see this as an important issue; however, it becomes very important if your program exceeds a certain size. You may be working as a programmer next year this time and may be continuing some other programmers' work in your new position. So, please keep your programs comprehensible.

6. Submission

- Homework creation and submission procedures will be covered during the recitation hours. Use Code::Blocks IDE and choose GNU GCC Compiler while creating your project. Name your project as "e1XXXXXX_HW1" where Xs are the digits of your student ID number. Send the whole project folder compressed in a rar or zip file. Name your submission as e1XXXXXX_ee441_hw1.rar. You will not get full credit if you fail to submit your project folder as required.
- Your C++ program should follow object oriented principles, including proper class and method usage and should be correctly structured including private and public components. Your work will be graded on its correctness, efficiency and clarity as a whole.
- You should insert comments to your source code at appropriate places without including any unnecessary detail.
- Late submissions are welcome, but are penalized according to the following policy:
    - 1 day late submission: HW will be evaluated out of 70.
    - 2 days late submission: HW will be evaluated out of 50.
    - 3 days late submission: HW will be evaluated out of 30.
    - 4 or more days late submission: HW will NOT be evaluated.

We wish you success!