C++ Fundamentals: Exam 2

The following tasks should be submitted to the SoftUni Judge system, which will be open starting **Sunday, 3 February 2019, 09:00** (in the morning) and will close on **Sunday, 3 February 2019, 15:00**.

For this exam, the code for each task should be a single C++ file, the contents of which you copy-paste into the Judge system.

Please be mindful of the strict input and output requirements for each task, as the tasks are evaluated automatically and not following the requirements strictly may result in your program's output being evaluated as incorrect, even if the program's logic is mostly correct.

You can use C++03 and C++11 features in your code.

Unless explicitly stated, any integer input fits into **`int`** and any floating-point input can be stored in **`double`**. On the Judge system, a C++ **`int`** is a **`32-bit`** signed integer and a C++ **`double`** is a **`64-bit`** IEEE754 floating point number.

NOTE: the tasks here are NOT ordered by difficulty level.

Task 2 - Car Dealership

A business man has an AutoHouse, he needs a program that calculates all the costs that he has.

The program should have the following informations (datas):

AutoHouse - Capital **(** the capital can not be zero or negative, if the condition is true, the Capital default value should be 10 000 **)**

AutoHouse - Overcharge tax : (the overcharge can not be zero or negative, if the condition is true the Overcharge default value should be 500). This is the tax of single sold car.

- EXAMPLE : Total Incoming = price of the car + overcharge

AutoHouse - Storage of all cars.

- HINT : You can use array or vector

AutoHouse Methods:

- Setters, Getters (if you need)
- Show All Cars (Print informations about the cars)
- Buy A Car (Add cars in storage only if the Capital is sufficient, decrease the Capital in function of the car price)
- Sell All Cars (Empty the storage and prints Capital Before sells and After Sells, increase the Capital in function of Total Incoming)

Car - Make (Brand of car)

Car - Year (Year of manufacture, can not be under 1950 and after 2019, if the condition is true, the Year default value should be 2000)

Car - Price (Price of car, floating point number, can not be zero or negative, if the condition is true the Price default value should be 1000)

Car Methods :

- Setters, Getters (if you need)
- Print Info (Prints MAKE, YEAR, PRICE of the car)

Write a MENU that has the current select options:

1 - Show All Cars

2 - Sell All Cars

3 - Buy a Car

0 - Exit menu

Input Example

10000 (Initial Capital)
500 (Tax Overcharge)
3 (Choice from the menu BUY a car)
1 (Choice from the menu SHOW ALL CARS)
0 (Choice from the menu EXIT MENU)
Audi (Car Make)
2009 (Car Year of manufacture)
5700 (Car Price)

Output Example

```
Make : Audi
Year : 2009
Price : 5700
```

Restrictions

MAKE SURE THAT :

- Before choice Show all cars or Sell a car, the user HAS TO INPUT (BUY) a car !

- The user has to input 0 to exit the menu !

Example I/O

Example Input	Expected Output
10000 500 3 1 0 Audi 2009 5700	Make : Audi Year : 2009 Price : 5700
5000 350 1 2 3 0 BMW 2007 4500	Capital before sell : 500 Capital after sell : 5350
15000 1000 1 3 3 3 0 Audi 2005 3500 BMW 2007 2000 Mercedes 2010 3000	Make : Audi Year : 2005 Price : 3500 Make : BMW Year : 2007 Price : 2000 Make : Mercedes Year : 2010 Price : 3000