Atılım University
Computer Engineering Department
CMPE 225 Object Oriented Programming
2024-2025 Fall Term
Homework I
Due Date: 13.12.2024 – 23:59

Write a C++ program that provides the user with the alarm message given by sensors produced for type-1 diabetes according to the blood sugar levels of the patients and the relevant sensor information.

The program should contain a ***Sensor*** base class. There should be 3 different sensor classes derived from this class. These child classes should be created for ***Dexcom***, ***Abbott*** *and* ***Medtrum*** sensors.

The attributes that the Sensor class should contain are for the data listed below.

→ *User name (patient name)*
→ *Sensor model*
→ *Sensor usage life (days)*
→ *The number of days the sensor has been used*
→ *The patient's blood sugar value*

The Sensor class contains a pure virtual function called *checkBloodSugarLevel()*. Each child class displays messages to the user for low levels, normal levels and high levels for different blood sugar ranges. The messages that each sensor type prints for different blood sugar levels are given in the table below.

Sensor Type (Child Class)	*Glucose Level (Blood Sugar)*	*Message*
Dexcom	bloodSugar < 60	"Critical low blood sugar!"
	60<=bloodSugar <=110	"Normal blood sugar."
	bloodSugar>110	"High blood sugar!"
Abbott	bloodSugar < 80	"Critical low blood sugar!"
	80<=bloodSugar <=130	"Normal blood sugar."
	bloodSugar>130	"High blood sugar!"
Medtrum	bloodSugar < 75	"Critical low blood sugar!"
	75<=bloodSugar <=120	"Normal blood sugar."
	bloodSugar>120	"High blood sugar!"

The *displayRemainingUsage()* function in the Sensor class, calculates the remaining sensor usage days (sensor life - days the sensor is used) and prints it on the screen. Override this function in children's classes. Each sensor provides an additional extension of the usage time in addition to the remaining usage days; Dexcom 10 days, Abbott 7 days, Medtrum 5 days.

In addition to these, the class should contain getter and setter functions and a virtual destructor. Destructor function prints a message on the screen indicating that the objects are deleted from memory.

Create an array of pointers for the Sensor class in the main function and use it to dynamically create objects for each child class. After printing the outputs that you can display in the sample outputs on the screen, clear these created objects from memory.

Sample Run 1:

```
Naz's Blood Sugar Check:
Sensor Model: G7
Testing with blood sugar: 75
Normal blood sugar.
Remaining usage days: 15 days

Gizem's Blood Sugar Check:
Sensor Model: Libre2
Testing with blood sugar: 75
Critical low blood sugar!
Remaining usage days: 14 days

Yaren's Blood Sugar Check:
Sensor Model: S7-960
Testing with blood sugar: 110
Normal blood sugar.
Remaining usage days: 10 days

Sensor destructed for Naz
Sensor destructed for Gizem
Sensor destructed for Yaren
```

Sample Run 2:

```
Naz's Blood Sugar Check:
Sensor Model: G7
Testing with blood sugar: 243
High blood sugar!
Sensor usage has expired!

Gizem's Blood Sugar Check:
Sensor Model: Libre2
Testing with blood sugar: 135
High blood sugar!
Remaining usage days: 14 days

Yaren's Blood Sugar Check:
Sensor Model: S7-960
Testing with blood sugar: 122
High blood sugar!
Remaining usage days: 10 days

Sensor destructed for Naz
Sensor destructed for Gizem
Sensor destructed for Yaren
```

o The grading of the homework will be based on:

1. You have to upload your homework on time on Moodle.
2. Your file should be in the format that yourname_surname_hw1.cpp
3. Please do not upload .exe file, otherwise you will get zero.
4. Homework submitted via e-mail will be ignored.
5. This is not group work, so if anyone cheats or has a high similarity percentage (above 90%), will get zero.
6. If your code does not compile, your program will be evaluated over 80 pts.
7. Late submissions will not be graded.