ITU, Faculty of Computer and Informatics,
Asst. Prof. Dr. Gülşen CEBİROĞLU ERYİĞİT
Asst. Prof. Dr. Hatice KÖSE
Asst. Prof. Dr. Sanem SARIEL TALAY
Ress. Asst.s: Ahmet Aycan ATAK, G. Selda UYANIK, Neziha AKALIN

05.03.2014

BLG252E Object Oriented Programming

HOMEWORK – I, Due Date: 24 March 2014, Monday, 23:00

In this homework, you will design and implement two classes reflecting a trainee and a sample cardiovascular fitness activity (e.g. Running) machine: treadmill. Please go through below descriptions of these classes and the test program. The test program should be smoothly working with your implementation.

- Implement only necessary methods.
- Do not write extra constructors or destructors.
- Be careful with the methods/attributes which are supposed to be *constant*, *static*.
- Do not use *friend* relations in your homework and make sure that all of your class attributes are *private*.

➢ Trainee Class

A trainee is a person employing a personal trainer’s fitness program. Selected related attributes are:

name (*string* Supposed not to be changed), age (*int*), gender (*bool* 1: female, 0: male. Supposed not to be changed), height (*int* in cm.s), weight (*int* in gram s)

- dailyActivityDuration (*int* in minute s): Required activity duration per exercise. This attribute is determined and fixed by the personal trainer. Its value will be initialized by the constructor. It is supposed that all trainees are loyal to their trainers meaning: once started a trainee completes whole required duration (do not preempt his/her exercise).

- treadmillList (*Treadmill* **) Trainee should be associated with list of pointers to treadmills he/she can use. This list has a fixed size: upper limit of 3. But is a dynamic array and related space will be taken by the constructor.

Note: treadmillList is the list of treadmills he/she has permission to use but they are not necessarily all available

- numOfTreadmills (*int)* attribute keeps number of added treadmill pointers to *treadmillList* list, must be initialized to 0 by the constructor.

- totalCaloriesBurned (*int* in kilocalories) will be updated by exercise routine

And, required functionalities:

- Constructor: will set the name, age, gender, height and weight attributes according to its input parameters. dailyActivityDuration may or may not be specified by the caller (if not given, will be set to 30). Also, totalCaloriesBurned and numOfTreadmills should be set to 0 and memory space should be taken for treadmillList .

- looseWeight: updates trainee’s weight according to the totalCaloriesBurned property: decreases weight by 1.5 gr per each 1 kcal burned, also you can round up (may not be realistic but implement this one) inorder not to have uninteger value for weight given in grams. After weight update, totalCaloriesBurned should be reset to 0. If totalCaloriesBurned is 0, prints out a message and does not alter weight.

- exercise: should check list of treadmills and take the first marked as available, if none is available nor he/she does not have any related treadmills in his/her use prints out an error message and cannot perform any exercise. If an available treadmill is found, calls run function of treadmill with required input parameters (gender, age, weight and duration of itself). Return value of this run function will be burned calories. Exercise then adds this number to totalCaloriesBurned.

- addTreadmill (Treadmill *): adds treadmill pointer given by the input parameter to the *treadmillList.*

- Destructor

➢ Fitness Machine Class: Treadmill

Sample cardio fitness activity is an indoor activity of running and related machine is treadmill. Some simplified attributes:

- numOfTreadmills: will hold the number of treadmills in the gym, will be initialized to 0 **once.** And will help to assign each newly created treadmill object its *id* value.
- id: will be assigned by the default constructor. *id* of the created object will be one greater than number of existing treadmill objects.
- availability: Boolean variable representing machine's availability.

And, required functionalities:

- Constructor: sets *availability* to true, assigns *id* value properly, adjusts *numOfTreadmills* value.
- isAvailable: returns status of *availability*.
- getId: returns *id* of object
- run (*bool, int, int, int*): should set *availability* to false, calculates and returns total calories burned by trainee using this machine. Input parameters in order are: gender, age, weight of caller trainee and duration of exercise. Function uses below formula and multiplies kcal/min by duration to get total calories burned:

Female: kcals/min = (-20.4022 + 0.4472 x 100 - 0.1263 x weight(kg) + 0.074 x age) / 4.184
Male: kcals/min = (-55.0969 + 0.6309 x 100 + 0.1988 x weight(kg) + 0.2017 x age) / 4.184

*Heart Rate taken as 100, Very abstract calculation .

Before return, sets *availability* to true.

- unsetAvailability: will set *availability* to false. For test purposes. (Check main)
- Destructor

➢ You are not allowed to use STL members.

➢ For any questions, contact to Ress. Asst. G. Selda Uyanık via seldauyanik@itu.edu.tr.

➢ **Test Program:** Initially, it is suggested for you to through below C++ program which includes hints for your implementation. Afterwards, please use it for testing your whole program/codes.

```
#include<iostream>

// Other includes

using namespace std;

int Treadmill::numOfTreadmills;

int main(int argv, char * argc[])
{

         Treadmill::resetNumOfTreadmills();             // Resetting number of treadmills to 0.

         // Creating Treadmills for the gym:
         Treadmill tM1;
         Treadmill tM2;
         Treadmill tM3;
         Treadmill tM4;                                 // Check constructor/destructor messages
         {
                  Treadmill tM5;
         }
         Treadmill tM6;

         // Creating a Trainee: (dailyActivityDuration is 30 minutes as default) // Check Constructor messages
         Trainee trainee1 ("Sally Brown", 18, 1, 170, 55000);
         // Trainee trainee2 = trainee1;                              // When uncommented, should work properly

         // Try a daily exercise before treadmill/s are related with this trainee:
         trainee1.exercise();                                             // Check error message

         // Adding treadmill pointers to tradmill list of Trainee
         if (!trainee1.addTreadmill(&tM1)) cout << "Add Operation 1 is not successfull" << endl;
         if (!trainee1.addTreadmill(&tM2)) cout << "Add Operation 2 is not successfull" << endl;
         if (!trainee1.addTreadmill(&tM3)) cout << "Add Operation 3 is not successfull" << endl;

         // Should print out error message since max list size is 3.
         if (!trainee1.addTreadmill(&tM4)) cout << "Add Operation 4 is not successfull" << endl << endl;

         // Try to loose weight before exercise :
         trainee1.looseWeight();                                          // Check error message

         // SAMPLE Workout:
         trainee1.exercise();

         // Update Weight:
         trainee1.looseWeight();

         // What if all treadmills are unavailable?
         tM1.unsetAvailability();
         tM2.unsetAvailability();
         tM3.unsetAvailability();

         trainee1.exercise();                                             // Check error message

         return 0;
```

➢ Test Program's Output:

```
Treadmill with id: 1 is created
Treadmill with id: 2 is created
Treadmill with id: 3 is created
Treadmill with id: 4 is created
Treadmill with id: 5 is created
Treadmill with id: 5 is deleted
Treadmill with id: 5 is created

A female trainee is created:

AGE                     : 18
NAME                    : Sally Brown
GENDER                  : 1
HEIGHT                  : 170
WEIGHT                  : 55000
DURATION                : 30

Sorry, no treadmills are related to trainee: Sally Brown, Can NOT exercise

Add Operation 4 is not successfull

Sorry, NO workout = NO LOSS

An available treadmill is found having id: 1
Exercisizing for 30 minutes ...
Total Kilocalories burned: 134

***** Weight UPDATE: *****
Previous weight was     : 55
NOW                     : 54.799 kg...

Sorry, NO treadmills are AVAILABLE to: Sally Brown, Can NOT exercise
```

Due to: 24 March 2014, Monday, 23:00

Submission NOTES:

- Submissions are through Ninova system and has a **strict deadline**, no assignments submitted after deadline is accepted.
- Source codes should be uploaded as one compressed file named after your student number as "*student_number*.zip"
- You should implement your program in C++ programming language.
- Be sure that your program works and produces reasonable and expected results.
- Be sure to include clear code comments.
- This is *not* a group assignment and getting involved in any kind of cheating induces negative grades.