BLG 252E OBJECT ORIENTED PROGRAMMING Spring 2016

Homework 2

Due by 23:59 AM on Friday, Apr. 22, 2016

Submission through Ninova only!

No late submission will be accepted!

In this homework, you will implement some utility classes for an imaginary two-player game called **Pick-a-Card-Color**. The game engine interacts with an abstract class to simulate the players' moves. Each kind of move inherits from the *Base abstract class* and implements *virtual member functions*.

In **Pick-a-Card-Color**, each player has 4 cards with different colors (Red, Dark Red, Blue, and Green) available to him. In a single play of the game, the two players simultaneously pick cards. A player wins according to the following rules:

- If both players choose the same card color, this play ends in a tie.
- If one player chooses Red, and the other chooses Green, the player choosing Red wins this play.
- If one player chooses Dark Red, and the other chooses Green, the player choosing Green wins this play.
- If one player chooses Red, and the other chooses Blue, the player choosing Blue wins this play.
- If one player chooses Dark Red, and the other chooses Blue, the player choosing Dark Red wins this play.
- If one player chooses Green, and the other chooses Blue, the player choosing Green wins this play.

The outcome of each play of the game based on the players' choices is detailed in Table 1.1.

Table 1.1: Outcome of a play of the game based on the choices of the players.

Outcomes Based on the Choices of the Players		**Opponent's Choice**			
		Red	Dark Red	Blue	Green
Challenger's Choice	Red	There is a tie	Opponent wins	Opponent wins	Challenger wins
	Dark Red	Challenger wins	There is a tie	Challenger wins	Opponent wins
	Blue	Challenger wins	Opponent wins	There is a tie	Opponent wins
	Green	Opponent wins	Challenger wins	Challenger wins	There is a tie

Input: The program will read the players' choices from the text file, *deck.txt*. In this file, the first row shows the moves of the first player, and the second row shows the moves of the second player. Each color is abbreviated using its first letter (**(R)**ed, **(D)**ark Red, **(B)**lue, **(G)**reen). You should store these moves on a *heterogeneous list* (such as the one in the lecture slides, Pointers to Objects, OOP07). For example, if the first row is R, D, B, G, you need to create the relevant classes. For Red (R), the list must store the object of the Red class.

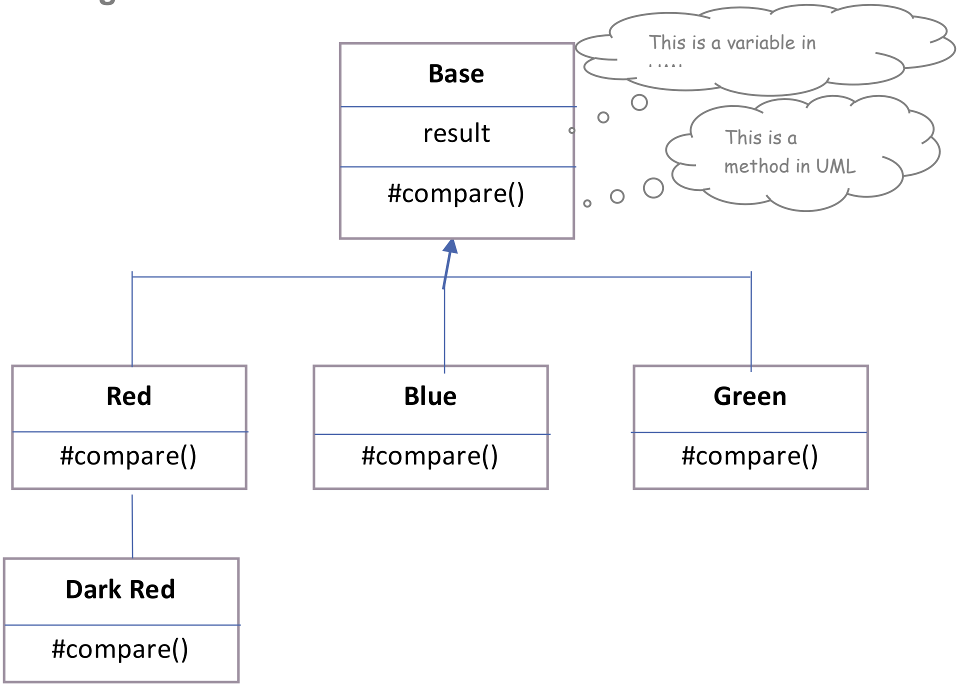
Output: The program will output the players' choices and the winner of each play. After the game is over, the program should output the total score of plays and the scores of each player as well.

The outcomes of loss, tie, and win are assigned the following scores: Lose(0), Tie(1), Win(2).

On screen:

```
Player1: D D B G G B D G R D
Player2: R D B R D R D R B G
Score1: 2 1 1 0 2 2 1 0 0 0
Score2: 0 1 1 2 0 0 1 2 2 2
Total Score1: Lose: 4 Tie: 3 Win: 3
Total Score2: Lose: 3 Tie: 3 Win: 4
Winner: Player2
```

1. Class Diagram



The Base class has a data member: *result (one out of this set { Lose, Tie, Win }).*
You can use **enum outcome { Lose, Tie, Win };**
The class also has a method: *compare().*

2. Implementation

You are asked:

1. to implement the unimplemented member functions of the Base class.

2. to define all classes corresponding to move types derived from the abstract class Base and compare() function.

3. to add insert() function. To add objects to **head** on the heterogeneous list, you should use insert() function as shown below.

   void HetList::insert(base* t){ … }

4. to add display() function to only show the results. This function is not a member of any class. You should pass necessary parameters as shown below.

   void displayResults(int, int, int, vector<int>&, …) {… }

3. Submission

->The aim of this homework is to understand inheritance and implement the classes and methods based on these concepts.

->Please submit all .h and .cpp files to **only** Ninova system **on time**. Late submission will not be accepted.

->Please use comments in your code to explain what you did and write your own code and also your program can be complied on ssh. Plagiarism and any other forms of cheating will have serious consequences as in the first assignment. If you have any questions about homework, you can ask Kübra Cengiz via e-mail (kcengiz@itu.edu.tr).