Programming Design, Spring 2014
Homework 10

Instructor: Ling-Chieh Kung
Department of Information Management
National Taiwan University

Submission. To submit your work, please upload the following file to the online grading system at http://lckung.im.ntu.edu.tw/PD/.

1. Your .cpp file for Problems 1.

Each student must submit her/his individual work. No hard copy. No late submission. The due time of this homework is ***8:00am, May 19, 2013***.

Problem 0

(0 point) Please read Sections XXX of the textbook.[1] In any case, I strongly suggest you to read the textbook thoroughly before you start to do this homework.

Problem 1

(100 points) In a typical Role-Playing Game (RPG), a player plays the role of a character, who keep beating enemies (typically monsters, bad guys, or other players' characters). By beating enemies, one earns money to buy better weapons and earn experience points to advance to higher levels. Both better weapons and higher levels make one stronger to beat those stronger enemies.

In many RPGs, one can choose the occupation for her character(s). The occupation typically affect the ability of a character. For example, a warrior is typically physically strong but innocent about using magic. On the contrary, while a wizard may use magic to heal her teammates or hit enemies, she/he typically must hide from physical damages. Characters with different occupations have different attributes and behave differently. However, they are all characters.

In this program, you are give a class `Character` that defines some general features of an RPG character. From it, you need to create two new classes `Warrior` and `Wizard` so that in your program you may create warriors and wizards. They will beat monsters, earn experience points, advance their levels, and become stronger. To verify the correctness of your program, from time to time you will be required to print out the current status of these characters in your game.

The class `Character`

The class `Character` is provided in "PDSp14_hw10_class.txt". We assume that a character has her name (`string name`), her current level (`int level`), her accumulated experience points (`int exp`), and three ability levels: power, knowledge, and luck (`int power`, `int knowledge`, and `int luck`). All characters start at level 1 with no experience at all (but when they join your team, they may be at different levels). For all characters in our game, the number of experience points required for level k is $100(k-1)^2$. The number 100 is stored in `static const int expForLevel`.

There are three public member functions. To create a character, we must specify all its attributes except the experience point. In our game, a new character at level k always starts with $100(k-1)^2$ experience points. This is indicated in the constructor. Another public member function `print()` prints out the current status of a character following a specific format. The third public member function `beatMonster()` should be invoked when the character beats a monster. The parameter `int exp` is the number of experience points earns in this battle. This this, `beatMonster()` increments the accumulated

[1] The textbook is *C++ How to Program: Late Objects Version* by Deitel and Deitel, seventh edition.

experience points and checks whether she should be advanced to the next level. When this happens, a private member function `levelUP()` is invoked. Note that while a character's `level` will be incremented, her abilities will remain the same! This is because characters of different occupations get different improvements when advancing one level, but this is not defined for a general character (who has no occupation). One of the most important thing that you need to do is to implement the improvement of abilities in your `Warrior` and `Wizard` classes:

- When a warrior is advanced to the next level, her power, knowledge, and luck will be incremented by 10, 5, and 5. Therefore, a level-k warrior's three attributes are $10k$, $5k$, and $5k$ for all $k \in \mathbb{N}$.
- When a wizard is advanced to the next level, her power, knowledge, and luck will be incremented by 4, 9, and 7. Therefore, a level-k wizard's three attributes are $4k$, $9k$, and $7k$ for all $k \in \mathbb{N}$.

Of course, you may treat `Character` simply a reference and create `Warrior` and `Wizard` completely by yourself. Or you may modify `Character` to create `Warrior` and `Wizard`. It is up to you. Nevertheless, we suggest you to use inheritance to finish this homework. This will not only save you some time but also "advance you to the next level."

Input/output formats

In the input file, a line means an event in the world of our game. The event type depends on the first character:

- (A new warrior joins the team) If a line starts with `R`, a new warrior joins. `R` will be followed by a sequence of English letters representing her name and a positive integer as her current level. The three pieces of information are separated by two white spaces.
- (A new wizard joins the team) If a line starts with `D`, a new wizard joins. Her name and current level are provided in the same way as that for a warrior.
- (One beats a monster) If a line starts with `B`, it will be followed by either `R` or `D` indicating either warrior or wizard, a string as a name, and a positive integer as experience points. This means that the character of the name and the occupation beats a monster and earns those experience points. If after the battle she has enough experience points, she should be advanced to the next level.
- (Print out one character's status) If a line starts with `P`, it will be followed by either `R` or `D` indicating either warrior or wizard and a string as a name. The status of the character of the name should then be printed out in the format the same as the following example:

  ```
  Warrior Mikasa: Level 13 (13900/16900), 130-65-65
  ```

  As you expect, this character is a warrior and her name is Mikasa. She is currently at level 13 with 13900 experience points. To advance to the next level, she needs 16900 points. Her current power, knowledge, and luck are 130, 65, and 65. Note that there are five white spaces. Also note that the `print()` function in `Character` follows the format except that the occupation is not printed out.

At most ten characters join your team. They all have different names. They do not die nor leave your team. All lines of input will be effective. An example of input and output are in "PDSp14_hw10_input.txt" and "PDSp14_hw10_output.txt".

Grading criteria

- 70 points are given based on the correctness of your output, which should occur only for `P` lines. There are 35 `P` lines, each count for 2 points. The input file up to the tenth `P` line is provided in "PDSp14_hw10_testing.txt".
- 30 points will be based on how you write your program, including the logic and format. Please try to write a robust, efficient, and easy-to-read program.