ITEC 120: Principles of Computer Science I

Homework 11 – Monster Training League

>>>>>>>> Due Date: Wed., April. 25th at 10:00PM via Desire2Learn <<<<<<<

In doing this homework, remember to abide by the RU Honor Code.

Problem 1
50 points

The final battle is upon us, or at least that is what your boss says. The sales figures from the massive monster battle simulator have been astounding; your boss thinks that one more simulator will drive your competitors out of business.

The marketing department wants to combine the elemental monster battling capabilities from your previous best seller with the massive battles from your last game. They think that this combination will provide the best chance of another hit.

The game will be based around cooperative play where two players battle each other. Each player has N different monsters under their control that they use during battle. After each battle, the monsters are returned to perfect health so they can battle again.

A battle consists of two trainers taking the first monster from their repertoire and battling them against each other. When one monster is defeated, the next monster in the trainer's repertoire is brought in to continue the fight. The battle is over when a trainer runs out of monsters to fight the other trainer with. The battle is a draw if both trainers run out of monsters at the same time. Otherwise, the winner of the battle is the trainer that has at least one monster left alive.

Each monster has a certain number of HP, the amount of damage it deals, and a name. When a monster fights another monster, they hit each other (HP is reduced by the attack damage of the opposite monster) at the exact same time, and repeat the process until one monster's HP is less than 1. When one monster's HP is < 1, the trainer must bring out another monster to continue the battle, otherwise they will lose the battle.

Additionally, every monster will have an elemental type of normal, earth, fire, or water. Earth is strong against fire, but weak against water. Fire is strong against water, but weak against earth. Water is strong against earth, but weak against fire. Normal monsters are not stronger or weaker than any of the other types of monsters. Being strong against an elemental type gives the monster a 30% increase to its attack damage. Being weak against an elemental type reduces a monster's damage by 30%.

Input

Input will consist of one of three commands. The first command is addTrainer, the next is battle, and the last is quit.

The add trainer command is on a line by itself, the next line contains the trainer's name, a space, and the number of monsters that the trainer possesses. For the next N lines of the input (where N is the number of monsters that the trainer possesses), each line represents a monster in the specific trainer's repertoire (and the first line is the first monster in the trainer's repertoire, etc…). The format for a line of input that represents a monster is a string representing its name, a space, then another string representing its type (normal, earth, water, fire), a space, an integer representing its HP, a space, and an integer representing the amount of damage its attack deals.

You do not need to worry about incorrect input, the input given to your program is guaranteed to be correct.

The battle command is on a line by itself. The next line contains two Strings representing the names of the trainers that will battle.

The quit command stops your program.

Sample input file

```
addTrainer
Ash 3
Charizard fire 25 8
Squirtle water 20 5
Rocky earth 30 15
addTrainer
TeamRocket 4
Rat normal 10 2
Pidgey normal 10 5
Snake earth 12 8
Snake earth 12 8
addTrainer
Random 1
Dragon fire 50 15
battle
Ash TeamRocket
battle
TeamRocket Random
battle
Ash Random
quit
```

WARNING

Your program must work with file redirection due to the amount of input necessary to test solutions. You can test it yourself by typing java yourProgramName < inputFile.txt (this will work if you use only one Scanner in your program). If your program will not run in this manner, then you will receive at most a 25 on this assignment. Make sure you copy the sample input in this assignment, paste it into a text editor, save it as a plain text file in the same directory where your program is stored, and test running your program with the < filename.txt that you created.

Output

Your program must produce output that tells the user what is happening in the program. When a battle is commenced, the program must display which two trainer's are fighting. Next, it should announce the monster's names that are going to be battling. Whenever a particular monster is defeated, the event, trainer and monster name must be reported to the user. Whenever a replacement monster is brought in, its owner and the monster's name should be displayed. Whenever a draw between two monsters occurs, this should also be reported to the user.

Note: If you store your results in an int, the output will different than if you store them in a double.

Sample output file

```
Ash is fighting TeamRocket
Starting battle is Charizard versus Rat
TeamRocket's Rat lost
TeamRocket is bringing out Pidgey
TeamRocket's Pidgey lost
TeamRocket is bringing out Snake
*Draw*
Ash's Charizard lost
Ash is bringing out Squirtle
TeamRocket's Snake lost
TeamRocket is bringing out Snake
*End draw*
TeamRocket's Snake lost
Ash won the match
TeamRocket is fighting Random
Starting battle is Rat versus Dragon
TeamRocket's Rat lost
TeamRocket is bringing out Pidgey
TeamRocket's Pidgey lost
TeamRocket is bringing out Snake
TeamRocket's Snake lost
TeamRocket is bringing out Snake
TeamRocket's Snake lost
Random won the match
Ash is fighting Random
```

```
Starting battle is Charizard versus Dragon
Ash's Charizard lost
Ash is bringing out Squirtle
Ash's Squirtle lost
Ash is bringing out Rocky
Random's Dragon lost
Ash won the match
```

Implementation suggestions
For a program of this size (about 200 lines of java code for my solution, not counting any documentation), it is essential that you practice incremental implementation. That is, don't attempt to write the entire program at once, even though you may have a complete design. Here is a suggested order of implementation:

- Create the capability to read monsters from the command line.
- Create the capability to read multiple monsters and store them in an array
- Create the capability to associate multiple monsters in a Trainer
- Create the capability to store multiple trainers in an array.
- Create the capability to battle monsters.
- Create the capability to battle multiple monsters in succession.

Do not wait until two or three days before the assignment is due to start it. You will probably need to spend between five and fifteen hours working in order to complete this homework assignment. You will probably not be able to write, test, and document it at one time.

Constraints
Damage and HP must be stored as integers. If you use double variables, your answer will differ from the sample input. You must use at least four classes (including the driver class) in your solution. There are many opportunities to do so in this project. For instance, you might use a class to encapsulate the information for a monster, a class to model a trainer, and a class to battle two trainers.

The reference solution for this project is 206 lines of code without comments. Feel free to use less or more code in your program.

Submission requirements
You must submit the .java files that make up your program to Desire to Learn under the Homework #11 assignment. You may demo the homework day after it is due or you will not be able to contest the grade.

Grading Rubric
20 Points - Are classes designed / implemented properly?
15 Points - Do the commands work as expected?
10 Points - Is the program commented using inline and javadoc comments?
5 Points - Was a reasonable test case included in the submission?