hw05

▼ Hide Assignment Information

Instructions

hw05

Focus

- Dynamic memory
- But this does *not* involve copy control. No destructor, copy constructor or assignment operator.

Update

- Do **NOT** make the Noble class a friend of the Warrior class.
- Do **NOT** put a pointer in the Warrior class to the Noble class.

Problem:

Building on previous assignments, we will be reading a file of commands to create Nobles and Warriors, and sending them off to battle.

Key differences:

- Each time a warrior or a noble is defined, we will create it on the heap.
- We will keep track of the nobles in a vector of pointers to nobles.
- We will keep track of *all warriors* using a vector of pointers to warriors.

The input file will be named "`nobleWarriors.txt`".

Commands

- Noble. Create a Noble on the heap.
- Warrior. Create a Warrior on the heap.
- Hire. Call the Noble's hire method.
- Fire. Call the Noble's fire method.
- Battle. Call the Noble's battle method.
- Status. The status command shows the nobles, together with their armies, as we did previously. In addition, it will show separately the warriors who do not currently have a employer
- Clear. Clear out all the nobles and warriors that were created.

Our application is going to rely on each Noble having a unique name and each Warrior having a unique name. Otherwise, how would we be sure who we were hiring (or firing). Note that this is not a requirement of the Noble and Warrior classes themselves, just of this particular *use* of them, i.e. our application.

Whenever you are displaying a Noble or a Warrior, you will use the **output operator** for the class.

Handle errors!

Previously we promised that all of the commands we gave you in the input would be valid. Now we would like you to take some responsibility for checking the input. First, we still guarantee that the format of the file will be correct. That means that the Warrior command will always have a name and a strength. The Battle command will always have two names. The Status command will not have any other information on it than just the word Status.

However, **you will need to detect and report any issues indicating inconsistencies**, such as:

- Noble command: if a Noble with a given name already exists.
- Warrior command:if a Warrior with a given name already exists.
- Hire command: If a Noble tries to hire a Warrior and either of them do not exist or the Warrior is already hired.
- Fire command: If a Noble tries to fire a Warrior and either the Noble does not exist or does not have the Warrior by that name in his army.
- Battle command: If a Noble initiates a battle with another Noble, but one or the other does not exist.

We have not specified the format of these error messages, so we'll leave that up to you. (You get to be creative!)

Example input file and the corresponding output are attached.

Due Date

Oct 19, 2022 11:59 PM

Attachments

nobleWarriors.txt (643 Bytes)

output.txt (1.29 KB)

Download All Files

Submit Assignment

Files to submit *

(0) file(s) to submit

After uploading, you must click Submit to complete the submission.

Add a File **Record Audio** **Record Video**

Comments

Submit **Cancel**