

[Content](#) [Announcements](#) [Assignments](#) [Discussions](#) [Quizzes](#) [Grades](#) [Zoom](#) [More Tools](#) ▼ [Course Reports](#) ▼ [Help](#) ▼[Assignments](#) > [hw06](#)

# hw06

[▼ Hide Assignment Information](#)**Instructions**

## hw06: Cyclic Association and Separate Compilation

### Focus

- Cyclic Association
- Separate compilation
- Namespace
- Operator overloading

### Problem:

We are revisiting the world of Nobles and Warriors (hoping you aren't sick of it...), now adding one new feature: not only can warriors be fired, they can also runaway!!! Of course, when a warrior runs away, he has to inform his noble. Which means the warrior **has to know who hired him** and be able to communicate with him. Which means, obviously, that he will need a pointer to his noble. Good thing we learned about forward class declarations!

Of course your Noble and Warrior classes should have all the same functionality that they had in the prior exercises, e.g. nobles can fire warriors.

Take this opportunity to clean up any "sloppy" code from before. For example, your battle method should not have redundant code. Make good use of small methods.

You will provide two solutions:

- The first is a single file, hw06-single.cpp, that solves the problem of the cyclic association and leaves no method or function definitions inside the class definition, just the prototypes and the field definitions.
- The second splits the first into separate files for separate compilation.
  - In case you haven't covered how to do this yet, you will in plenty of time to finish the assignment. Meanwhile, get to work on the first part above!
  - None of the **code** should change between the first and second implementations
  - There should be separate header and implementation files for each class.
  - And for the separate compilation solution, place the code in the WarriorCraft namespace.

### Test Code and Output

For your convenience, we have attached test code and corresponding output.

### Turn in

Six files:

- hw06-single.cpp

- Noble.h
- Noble.cpp
- Warrior.h
- Warrior.cpp
- and also (for the grader's convenience) the attached test program as hw06.cpp.



Start Date

Oct 19, 2022 9:00 AM

Due Date

Nov 2, 2022 11:59 PM

Attachments

-  [output.txt](#) (1.14 KB)
-  [hw06-testcode.cpp](#) (1.84 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