Homework #3 RELEASE DATE: 11/03/2015 DUE DATE: 11/23/2015, 12:00 (NOON)

As directed below, you need to upload your submission files to the github repository under the exact guidelines of the TAs.

Any form of cheating, lying or plagiarism will not be tolerated. Students can get zero scores and/or fail the class and/or be kicked out of school and/or receive other punishments for those kinds of misconducts. Discussions on course materials and homework solutions are encouraged. But you should write the final solutions alone and understand them fully. Books, notes, and Internet resources can be consulted, but not copied from.

Since everyone needs to write the final solutions alone, there is absolutely no need to lend your homework solutions and/or source codes to your classmates at any time. In order to maximize the level of fairness in this class, lending and borrowing homework solutions are both regarded as dishonest behaviors and will be punished according to the honesty policy.

Only English is allowed for writing any part of your homework. We do not accept any other languages.

1 Description

In this homework, you are asked to redo homework 1. That is, re-implement the old-maid card game, as shown in https://en.wikipedia.org/wiki/Old_maid_(card_game).

Sounds too simple, isn't it? Well, we hope you to consider two more things in this homework. First, write an OOP program, instead of an NOP one. Second, recall that in your homework 1, you were only asked to implement one particular variant (52 cards + 2 jokers). Now, we want to ask you to implement **two** different variants in this homework, and try to minimize your coding efforts by using **inheritance** properly. You can choose the two variants from https://en.wikipedia.org/wiki/Old_maid_(card_game), or design your own variant as long as it is sufficiently similar to the original game.

In particular, we hope that you have something like class OldMaid that describes the general old maid game, as well class VariantOne extends OldMaid and class VariantTwo extends OldMaid that describes the specific variants. The "shared" code should go to the parent class as much as possible, and the "specific" code goes to the child classes.

We will not restrict your design on the class structures other than having an inheritance structure somewhere like the above ones. You can freely add other classes if you need. For simplicity, you can just focus on implementing **four-player** games like homework 1. You can also freely design your outputs to show the game to the TAs. If you do not know what output to show, just follow the output in homework 1.

2 Requirements

- Write the code that implement the two variants along with an inheritance structure like the one discussed above. Note that readability of your source code would be worth 20 points out of 200 this time. That is, your grading TA would give you points based on the following qualitative measure:
 - 20 very readable
 - 16 readable
 - 12 mostly readable, but with some unreadable parts
 - 8 mostly unreadable, but with some readable parts
 - 4 unreadable
 - 0 very unreadable
- Write a short report with at **most** five A4 pages **IN ENGLISH** that contains the following items:
 - (1) your name and school ID
 - (2) your design of class structures and their reasons
 - (3) how a human player should play with your program for the two variants

- (4) how you tested the correctness of your program
- (5) the sample output from each variant of your program
- (6) any part that you implemented that is worth getting "bonus" points

You should submit your report in **PDF** format.

3 Submission File

Please submit your code with github as directed in the homework submission guide. Your directory structure (under hw3) should be

- src/*, your source code
- Makefile, where the TAs can use make on CSIE R217 linux machines to compile your code, and then make run to test your program as guided in your report
- a PDF file report.pdf, which is your report file written in English

Please do not include any other files (e.g. class files) in the repository. Otherwise you may lose some points.