

# Go Fish!

## EE 312

**100 points**

**Due: Friday 4/12/2018 at 6:00pm**

---

The goal of this assignment is to use some pre-defined classes to create more classes and to use these classes to play a simple card game. We will use the basic rules of [Go Fish](#). The requirement is that your program work with two players (starting with 7 cards).

### The basic game

5 cards are dealt to each player, or **seven if there are only two players**, and the remaining card pack is placed between the players.

The player whose turn it is to play asks another player for their cards of a particular rank. For example, "Agatha, do you have any threes?". A player may only ask for a rank of which they already hold at least one card. The recipient of the request must then hand over all cards of that rank. If the call was successful, the same player has another turn. If the player who was asked has no cards of that rank, they say "Go fish" (or simply "Fish"), and the asking player draws the top card from the pack. The turn then passes to the player who was asked.

When one player has two of the same cards of a given rank, they form a *pair*, and the cards are placed face up on the table. The game ends when all twenty-six books are formed, and the player who won the most books wins.

If the player whose turn it is has no cards left in hand, the game is not over, but they simply draw the top card from the pack and the turn passes to the next player in the rotation, or, in the case of a two-player game, the other player.

(from Wikipedia)

We will be using the vector class from the Standard Template Library (STL) to hold the players cards and to hold the "books" that are formed from pairing cards. Each player will have a hand and a collection of books.

---

Some classes that we will use for this project. You will need to implement most of these classes before you can get to the "Go Fish" game.

[card.h](#) - simulates one playing card (a suit and a value).

[deck.h](#) - simulates a deck of 52 cards

[player.h](#) - simulates a player in the game

[card\\_demo.cpp](#) - a demo of a main program that will deal some cards

---

For the output of the program, you should print each play to a text log file (you may want to echo to standard output while debugging) and the eventual winner.

**Note: The computer will play the hands of both players. There is no input from people.**

e.g. Joe asks - Do you have a J?

Jane says - Go Fish

Joe draws 4d

Jane asks - Do you have a 4?

Joe says - Yes. I have a 4.

Jane books the 4..

Jane asks - Do you have a 6?

Joe says - Go Fish

Jane draws Jc

etc.

Your output can certainly be different as long as it describes the game. Make sure the results of the game are clear.

---

Notes:

- 1) Start by implementing the Card class and testing it. Then the Deck class. Then the Player.
- 2) If you are unsure about how the flow of the game should proceed, use your best judgement and document your decision.
- 3) Start early!

---

## Notes:

- You may do this program with a partner from the same lecture section. You both must be able to attend one discussion section or checkout. If you use a partner, please hand in one copy with both of your names on each file.
- You must use GitHub to store your files.
- If you run into an ambiguous situation regarding the card game, use your best judgement and document your solution.
- Your output should go to the file **gofish\_results.txt**.
- **You must name your C++ source files** `card.h`, `card.cpp`, `deck.h`, `deck.cpp`, `player.h`, `player.cpp`, and `go_fish.cpp`.
  - Hand in a zipped file named `prog06_XXXXXX.zip` where XXXXXX is your UT EID number.
  - Include a makefile that will compile your files. You can use [this one](#).
- Appropriate values should be stored in constant variables.
- Be sure to follow the documentation standards for the course.

**Turn in:** Turn in a zipped file named `prog06_XXXXXX.zip` where XXXXXX is your UT EID to Canvas.

Updated: 3/25/18 rlp