

# Programming Assignment #1

Go Fish!

CS 3358.751, Summer II 2013

Instructor: Jill Seaman

**Due: Friday, 7/12/2012** (upload electronic copy by 11:30am)

---

## Problem:

You will write a program that simulates the card game "Go Fish". For the rules to this game, see wikipedia [here](#) (we are using the variation where players give just one card when asked, and lay down pairs). Basically the players take turns asking the other player for a card of a certain rank, which the other player must hand over if they have such a card. The players' goal is to collect matching pairs of cards (of the same rank), placing them on the table.

You will use 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.

The main program will simulate the playing of the game, with two players.

**Classes:** The classes needed to support the main program are listed here (click on the link to get the header file content):

[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

You will need to add an implementation file for each of the header files, and implement a new driver called `go_fish.cpp`.

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

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

```
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

. . .

Game over:
Joe has 14 books
Jane has 12 books
Joe is the winner
```

Your output can certainly be different as long as it describes the game.

### NOTES:

- Start by implementing the Card class and testing it. Then the Deck class. Then the Player. You should have a test driver for each class (but don't submit them).
- I recommend using a makefile, but it's not required. We will cover samples in class.
- To simplify this assignment, **implementing the game play in go\_fish.cpp is optional!** You must have a file called go\_fish.cpp with a main function, and at the very least it should demonstrate and/or test the functions from deck and player. The main requirement is that you implement card.cpp, deck.cpp, and player.cpp correctly.
- The purpose of this assignment is to remember how to compile and execute programs (objective iii), review classes and multi-file programs (objectives i and iv), and to get some practice using the vector template from the Standard Template Library.

- Some topics you may need to learn for this assignment (we will cover them in class):
    - rand (for random number generation)
    - enum types
    - assert
    - vectors
- 

### **Style:**

See the Style Guidelines document on the course website.

### **Logistics:**

Since there are multiple files for this assignment, you need to combine them into one file before submitting them. You should use the zip utility from the Linux/Unix command line:

```
[...]$zip assign1_XXXXXX.zip go_fish.cpp player.cpp player.h  
deck.cpp deck.h card.cpp card.h
```

This combines the 7 files into one zip file, assign1\_XXXXXX.zip. Then you should submit only assign1\_XXXXXX.zip.

Name your file **assign1\_XXXXXX.zip** where XXXXX is your TX State NetID (your txstate.edu email id). It should look something like this: assign1\_js236.zip.

**Submit:** an electronic copy only, using the Assignments tool on the TRACS website for this class.