

Program Design & Testing Document for Program 2

Lyell C Read

Problem Statement

This problem/assignment is asking me to construct a C++ program that:

- Allows the user to play Go Fish
- Uses classes to implement that game as follows:
 - Using a card class
 - Using a deck class that has an array of cards
 - Using a hand cards that has some cards
 - Using a player class that has a hand of cards
 - Using a Game class that has a deck and some players.
- The program should, to this end, construct a deck of 52 cards, shuffle said deck to ensure random order, deal cards into a player's hand, and play the game of GoFish.
- The game should offer a play again option
- The game should not leak any memory, or segfault.
- The game should be factored into a .cpp file for each class used, as well as a header file and a gofish.cpp for the main() function. This will be compiled with a makefile.
- Dat makefile must compile on flip.engr.oregonstate.edu

I assume:

- NOTHING! I will check every input, so nothing is assumed.

This I will achieve by:

- A program that uses classes to provide the user a fine game of Go Fish
- Deallocating memory when done.

Understanding the Problem

As described in the requirements, the problem asks me to create a program that can play a nice game of Go Fish using classes. The game will be neatly organize, and follow the typical rules of the original game.

Pseudo Code (Simplified)

Class: Card:

Suit

Number

Class: Deck:

Array of Card classes
Number_of_cards

Functions:
Form (populate the deck class array of cards)
Shuffle (shuffle deck order)
Pop (pop an element off the top, i.e. "draw")

Class: Hand:

Variable length array of cards
Hand_size

Functions:
Show hand
Is in hand (checks if a number is in this hand)
Have books?

Class: Player:

Player's hand
Array of books
Book array length

Functions:
Show books

Class: Game:

Deck of cards
Array of players (2)

Functions:
Print messages
Manage user input

Various Functions, Explained:

Deck.form:

For x in range 0..12
For y in range 0..3

Place card with values
x and y at proper offset, calculated using
some calculation of x and y.

Deck.shuffle:

Do a random shuffle on the deck

Deck.Pop:

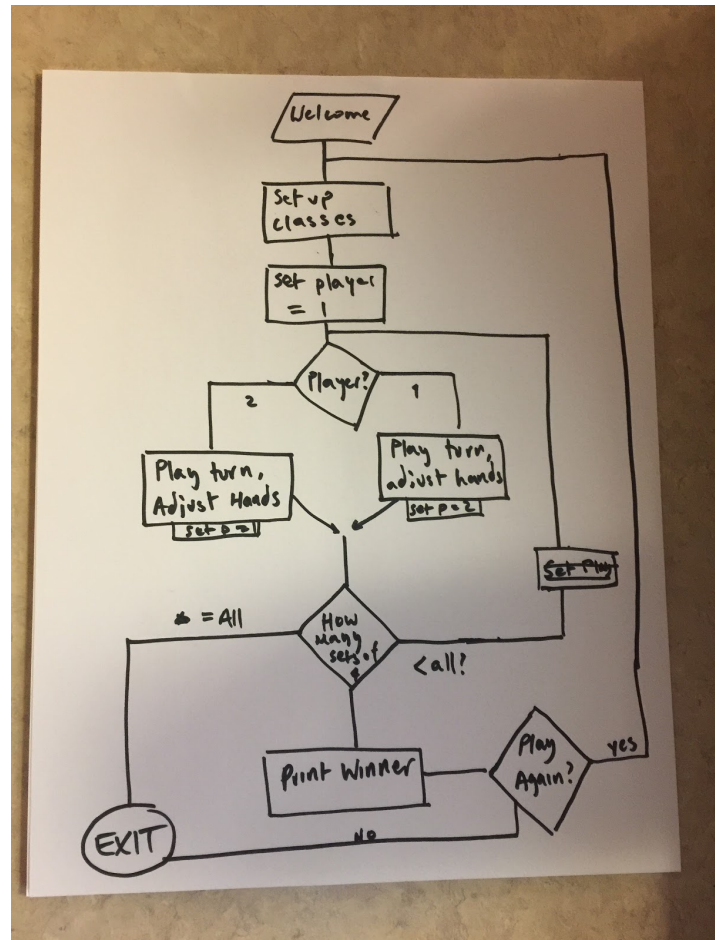
Remove the last element of the deck,
adjust size of deck to match

Hand.is in hand:

Look through hand to double check
that an int is in the hand (as the number, not
suit).

Hand.have books:

Checks through the hand to see if
there are four cards with the same number to
them. If so calls make book, and removes
from hand, adjusting accordingly.



Data Verification.

Note a 10 line table is far overkill here, as error checking only occurs on user input of a number from the list of cards in their hand. Therefore, I assume their hand is {2,3,3,7,8,11,12}

Ask for value from other player	What Should Happen	Does This Happen
""	Error - please enter again	
19	Error - out of range	
9ks0 2-	Error	
3	Game Proceeds	
9	Not in hand	
"Yote"	Error	