CSPP51036
Homework 3
Abstract Classes/Object methods
Due: Mon. Nov 8

1. Write a program to compute and graph the histogram of N discrete data samples. The input data f[0..N-1] is divided in m "bins" of equal size delta = (max(f) - min(f))/m, where m is a user-adjustable parameter and the f's are type double. The histogram value at each bin location delta, 2*delta,... is just a count of the number of samples that fall in that bin. Use java graphics capabilities to draw the histogram to a graphics window (we will briefly cover this last step in Nov. 1st class)

- Java package: homework3.prob1
- Class name: Histogram
- Methods: constructor Histogram(double[] data, int num_bins), void drawGraph() (draws Histogram in a graphics window)

2. Create a Deck class that represents a regular deck of playing cards. Use the Deck class to create a very simple Poker application, introducing a Game, Player, and Bet abstraction as well. You are free to define these as you see most appropriate. The application can simply simulate a game among a specified number of players -- does not need to include a live player. Logic can be simplified -- for this exercise design is more important than algorithmic sophistication.

For example application steps would be the following:

Input number of players: 3

- Shuffling deck ...
- Dealing cards ...
- (Some simple logic to place bet ...)
- (some simple logic for each player to decide how many cards to draw (0 to 3))
- Dealing cards ...
- (some more betting logic)
- declare a winner

Deck and Card class Design:

- Class name: Deck
- Deck methods: Deck.Deck(), void Deck.sort(), void Deck.shuffle(),
  Card[] Deck.deal(int n) (remove the top n cards and return them as
  an array)

Class Deck should support the following actions:
1. sort [sort deck from lowest to highest numerically with diamonds > hearts > spades > clubs].
2. deal(Player p, int n) (deals n cards to Player p)
3. shuffle [random sort]
4. proper implementation of clone, .equals, and toString()

- Class name: Card
- Card class: static final ints for HEART, DIAMOND, SPADE, CLUB, (or
  enum Suit defined as a nested class of Card with values DIAMONDS,
  HEARTS, CLUBS, SPADES) int getSuit(), int getNumber(), constructor
  Card(int suit, int number).

A few hints on designing the rest of this application:

- Player class: This class should really keep of track of Player related details, like the chip count, cards in hand, etc.

- Bet class: This should keep track of bet related details, like player placing the bet, the bet amount, etc

- Game class: This should really be your driver, abstracting one game at a time. Here you have to keep track of the Deck, Players, and game flow among other things.

- Supplemental classes: Like HandEvaluator(which would evaluate the hand achieved by a player), etc

- You ABSOLUTELY need to have this application designed on paper (or doc) somewhere before you type a single line of code, otherwise you will either run into a dead-end or really sloppy buggy design.

- The game is called 5 card draw and it is a very famous Poker Game, if you don't know it, google it.