CS 115
Assignment 2
Due – Thursday Oct 11th, 2018 7:00pm

1. [10%] A pseudo-random number generator is a computational facility for generating a sequence of seemingly random numbers. As our digital computers are deterministic devices, it cannot exhibit truly random behavior. Therefore, the numeric sequence generated by a pseudo-random number generator is in fact deterministic, but statistically "random" enough for performing computational simulation. A very popular algorithm for generating pseudo-random numbers is the Linear Congruential Method. The implementation of a linear congruential random number generator is provided to you as a C++ module:

[random.h]

[random.cpp]

Read the interface specification embedded in random.h to find out how the module is intended to be used. As a simple example, the following lines of client code initialize the module, and print out an unsigned int random number followed by another random number in the range 0 to 9 (inclusive):

```
#include <iostream>

using namespace std;

#include "random.h"

int main() {
  initializeWithDefaultSeed();
  cout << nextRandom() << endl;
  cout << nextRandomInRange(10) << endl;
  return 0;
}
```

Although the implementation provided to you is functionally complete, there are two (2) flaws.

- The preconditions are not properly enforced by assertions.
- The global generator state and helper functions are not properly encapsulated.

Fix these two flaws.
Hint: Consult the set of lecture notes entitled Program Organization Principles.

2. [55%] A deck of playing cards contains 52 cards. Each card belongs to one of four suits (i.e., Spades, Hearts, Diamonds, and Clubs). Each card also has a rank (i.e., ace, king, queen, jack, 10, 9, ..., 2). There is exactly one card with a given suit and a given rank.

Develop a playing card module (card.h/card.cpp) that offers the following services.

a. a record type (i.e., struct) for modeling a card

b. a function that returns the string representation of a given card

Remarks
The string representation of a card is its suit followed by its rank, with a white space in between. For example, the following are the string representation of some cards:

"Spade 5" "Heart ace" "Diamond king"

c. a function that compares if one card is more valuable than another card

Remarks
A card is more valuable than another card if the former has a higher rank than the latter, or if the two cards are of the same rank but the former belongs to a more valuable suit. Ranks are ordered, from higher to lower, as follows: ace, king, queen, jack, 10, 9, ..., 2. We follow the convention of Contract Bridge, and consider Spades the most valuable, followed by Hearts, and then Diamonds, and lastly Clubs.

d. a function that tests if two given cards are the same in suit and in rank

e. an array type, in the form of a typedef, for representing a deck of cards

f. a function that initializes a given deck array with the 52 standard playing cards

g. a function that randomly shuffles a given deck of cards

Remarks
To produce a random shuffling of an array A of 52 playing cards, follow the algorithm below:

```
for i stepping from 0 up to 51 do:
   let j be a random number between i and 51 (inclusive)
   swap A[i] and A[j]
```

h. a function that prints a given deck of playing cards to an output file stream

Remarks
The output must assume the following format. The output is composed of 52 lines, each containing the string representation of a card as specified in part b.

i. a function that reads a given deck of playing cards from an input file stream

Remarks
You may assume that the input is generated by the printing function as specified in part h.

The interface of your module must be carefully specified in the header file (in terms of Purpose, Argument(s), Precondition, Return, and Side Effect). Avoid passing card, deck and stream arguments by value. Qualify argument types properly to avoid unintended side effects. When appropriate, supply proper assertions to check for specified preconditions. There is, however, no need to use assertions to guarantee initialization.
Hint: Consult the set of lecture notes entitled Abstract Data Types via C++ Records.

3. [10%] Develop a C++ program, shuffle, that prints a randomly shuffled deck of playing cards to file output. The output of the program shuffle should follow the format specified in part h of the previous question.

4. [25%] Develop a C++ filter, game, that reads from file input a deck of playing cards (assumed to have been generated by the program shuffle) and simulates the following game using the input deck.

The game proceeds in three (3) rounds. In each round, the top two (2) cards of the remaining deck is dealt to the two players. The player who gets the more valuable card scores one point. At the end of the three rounds, the player who gets the highest score wins the game.

The output of the program should be the result of the game. The set of cards dealt to each player (i.e., their hands) should be printed first. This is followed by the score of each player. Lastly, the result of the game is reported. A sample output is given below.

```
Hand of Player 1:
Diamond 3
Club king
Spade 9

Hand of Player 2:
Spade 7
Heart king
Heart 5

Score of Player 1: 1
Score of Player 2: 2

Winner: Player 2
```

Submission Requirements

- random.h
- random.cpp
- card.h
- card.cpp
- shuffle.cpp
- game.cpp
- Screenshot of output from game.cpp