CSC 222: Chapter 14 Quiz Name ___Solution___________________

1. Complete the body of the following function so that it returns a pseudorandom number between `l` and `h` inclusive. Make your function handle the case where l > h.

```
#include <cstdlib>

int random_between(int l, int h) {
    if (l > h) {
        int temp = l;
        l = h;
        h = temp;
    }

    return l + (rand() % (h + 1 - l));
}
```

2. Complete the body of the following function so that it swaps the two cards at indices `i1` and `i2` (*note*: Assume all other member functions in Cards.h work as expected).

```
void Deck::swap_cards(int i1, int i2) {
    Card temp_card = cards[i1];
    cards[i1] = cards[i2];
    cards[i2] = temp_card;
}
```

3. Complete the body of the following function so that it returns the index of the lowest card between `l` and `h` (*note*: Assume all other member functions in Cards.h work as expected).

```
int Deck::find_lowest_between(int l, int h) {
    int lowest = l;
    for (int i = l; i <= h; i++) {
        if (!cards[i].is_greater(this->cards[lowest])) {
            lowest = i;
        }
    }
    return lowest;
}
```

4. Given the header file `Time.h` and doctest file `test_times.cpp`, write code for `Time.cpp` for the two constructors and the `to_string` member function so that the doctests pass.

```
#include <string>
#include "Time.h"

Time::Time() {
    seconds = 0;
}

Time::Time(int s) {
    seconds = s;
}

string Time::to_string() const {
    int h = seconds / 3600;
    int m = (seconds % 3600) / 60;
    int s = seconds % 60;

    string time_str = std::to_string(h) + ":";
    time_str += (m > 9) ? std::to_string(m) : "0" + std::to_string(m);
    time_str += ":";
    time_str += (s > 9) ? std::to_string(s) : "0" + std::to_string(s);

    return time_str;
}
```

```
/* Cards.h */

#include <string>
#include <vector>
using namespace std;

enum Suit {NONE, CLUBS, DIAMONDS, HEARTS, SPADES};
enum Rank {JOKER, TWO=2, THREE, FOUR, FIVE, SIX, SEVEN, EIGHT, NINE, TEN,
           JACK, QUEEN, KING, ACE};

struct Card
{
    Rank rank;
    Suit suit;

    Card();
    Card(Suit, Rank);
    string to_string() const;
    bool equals(const Card&) const;
    bool is_greater(const Card&) const;
};

struct Deck
{
    vector<Card> cards;

    Deck();
    void print() const;
    int find(const Card&);
    int find_lowest_between(int, int);
    void swap_cards(int, int);
    void shuffle();
    void sort();
};
```

```
/* Time.h */

#include <string>
using namespace std;

struct Time
{
    int seconds;

    Time();
    Time(int);

    string to_string() const;
}
```

```
/* test_times.cpp */

#define DOCTEST_CONFIG_IMPLEMENT_WITH_MAIN
#include <doctest.h>
#include "Time.h"
using namespace std;

TEST_CASE("Test can create and render Times")
{
    Time t1;
    CHECK(t1.to_string() == "0:00:00");
    Time t2(7);
    CHECK(t2.to_string() == "0:00:07");
    Time t3(72);
    CHECK(t3.to_string() == "0:01:12");
    Time t4(7 * 3600 + 11 * 60 + 19);
    CHECK(t4.to_string() == "7:11:19");
}
```