San Francisco Bay University

CS360 - Programming in C and C++
Homework Assignment #3

1. Create a program to shuffle and deal a deck of cards. The program should consist of class *Card*, class *DeckOfCards* and a main program. Class *Card* should provide:

   a. Data members *face* and *suit* of type *int*.
   b. A constructor that receives two *ints* representing the face and suit and uses them to initialize the data members.
   c. Two *static arrays* of *strings* representing the faces and suits.
   d. A *toString* function that returns the *Card* as a *string* in the form *"face of suit."* You can use the + operator to concatenate strings.

   Class *DeckOfCards* should contain:

   a. An *array* of *Cards* named *deck* to store the *Cards*.
   b. An integer *currentCard* representing the next card to deal.
   c. A default constructor that initializes the *Cards* in the deck.
   d. A *shuffle* function that shuffles the *Cards* in the deck. The shuffle algorithm should iterate through the *array* of *Cards*. For each *Card*, randomly select another *Card* in the deck and swap the two *Cards*.
   e. A *dealCard* function that returns the next *Card* object from the deck.
   f. A *moreCards* function that returns a *bool* value indicating whether there are more *Cards* to deal.

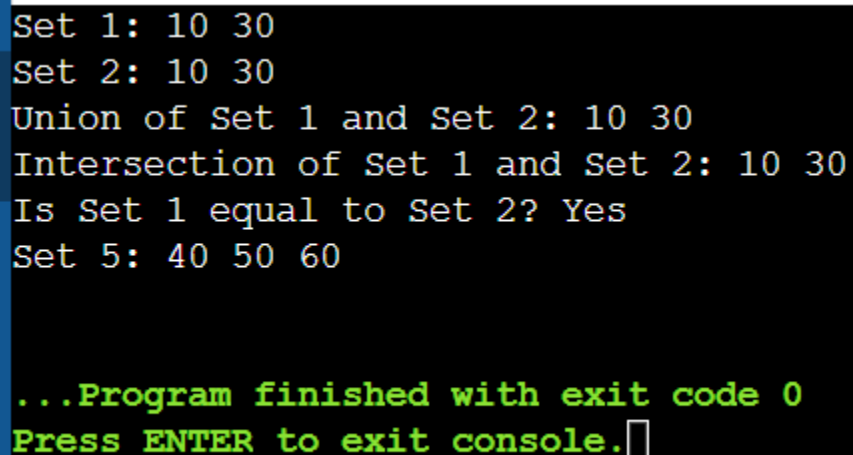   The main program should create a *DeckOfCards* object, shuffle the cards, then deal the 52 cards.

Here is the output of my code and the main code is in the cpp file

```
King of Clubs
3 of Spades
Queen of Clubs
10 of Clubs
7 of Diamonds
Jack of Spades
6 of Diamonds
10 of Hearts
King of Hearts
4 of Spades
6 of Hearts
5 of Diamonds
8 of Diamonds
7 of Clubs
2 of Hearts
8 of Clubs
9 of Diamonds
5 of Clubs
7 of Hearts
Ace of Clubs
Jack of Diamonds
9 of Hearts
King of Spades
3 of Clubs
Jack of Hearts
10 of Diamonds
5 of Hearts
8 of Hearts
Queen of Diamonds
4 of Diamonds
Queen of Spades
2 of Clubs
4 of Hearts
8 of Spades
7 of Spades
Ace of Hearts
3 of Diamonds
Ace of Spades
```

2. Create class *IntegerSet* for which each object can hold integers in the range *0* through *100*. Represent the set internally as a *vector* of *bool* values. Element *a[i]* is *true* if integer *i* is in the set. Element *a[j]* is *false* if integer *j* is not in the set. The default constructor initializes a set to the so-called "empty set," i.e., a set for which all elements contain *false*.

   a. Provide member functions for the common set operations. For example, provide a *unionOfSets* member function that creates a third set that is the set-theoretic union of two existing sets (i.e., an element of the result is set to *true* if that element is *true* in either or both of the existing sets, and an element of the result is set to *false* if that element is *false* in each of the existing sets).
   b. Provide an *intersectionOfSets* member function which creates a third set which is the set-theoretic intersection of two existing sets (i.e., an element of the result is set to *false* if that element is *false* in either or both of the existing sets, and an element of the result is set to *true* if that element is *true* in each of the existing sets).
   c. Provide an *insertElement* member function that places a new integer *k* into a set by setting *a[k]* to *true*. Provide a *deleteElement* member function that deletes integer *m* by setting *a[m]* to *false*.
   d. Provide a *printSet* member function that prints a set as a list of numbers separated by spaces. Print only those elements that are present in the set (i.e., their position in the *vector* has a value of *true*). Print --- for an empty set.
   e. Provide an *isEqualTo* member function that determines whether two sets are equal.
   f. Provide an additional constructor that receives an array of integers and the size of that array and uses the array to initialize a set object.

Now write a main program to test your *IntegerSet* class. Instantiate several *IntegerSet* objects. Test that all your member functions work properly.

Here is my code output and the main code is in the cpp file


```
Set 1: 10 30
Set 2: 10 30
Union of Set 1 and Set 2: 10 30
Intersection of Set 1 and Set 2: 10 30
Is Set 1 equal to Set 2? Yes
Set 5: 40 50 60


...Program finished with exit code 0
Press ENTER to exit console.
```