CS211 – Programming Practicum **Fall 2013**

Programming Project 11

Due: Thursday, 11/7/13 at 11:59 pm

Modification of Existing Code

For this program, you are to take an existing program and make some modifications to it. This skill is highly important since maintenance (or modification) of existing code is an extremely common task for many software engineers and computer programmers.

For this lab you are to take a simple blackjack program that uses a single deck of cards and player between two players and make it so it can use any number of decks of cards and allow up to 7 players. The original program is called bjOriginal.c and can be found at:
 http://www.cs.uic.edu/bin/view/CS211/ProjectsF13

Decks of Cards

A deck of cards contains 52 different cards. The cards are divided into 4 suits where each suit contains 13 cards of different ranks. The ranks are:

- Ace
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- Jack
- Queen
- King

The suits in a deck of cards are:

- Clubs
- Diamonds
- Hearts
- Spades

In a single deck of cards, there will be 4 cards of each rank: one from the Clubs suit. one from the Diamonds suit, one from the Hearts suit and one from the Spades suit. Each card is uniquely identified by its rank and its suit, for example: the 8 of Spades, King of Clubs, or the Ace of Spades.

When playing a game with multiple decks of cards, you start with some multiple of 52 cards: 104 cards when playing with two decks, 156 cards when playing with three decks, 208 cards when playing with four decks, etc. When using X decks, you have X copies of each card. Thus when playing to 4 decks, you have 4 Queen of Hearts.

The number of decks to be used in the modified program will be given using a command line argument. The program is to default to just using one deck of cards. If the command line argument/flag of –d is given, immediately following this command line argument another command line argument will be given indicating the number of desk to use. If this second command line argument is not an integer value of 1 or greater, print out a meaningful error message and quit the program. For the following examples, assume the executable program is stored in a file called bj.exe in the current directory and $ is the command prompt.

Examples using a single deck of cards:

```
$ ./bj.exe
$ ./bj.exe  -d 1
```

Examples using multiple decks of cards (3 in the first example 12 in the second):

```
$ ./bj.exe   -d  3
$ ./bj.exe  -d  12
```

Examples of an error (no value after the –d flag, value is not >= 1, value is not an integer):

```
$ ./bj.exe  -d
$ ./bj.exe  -d 0
$ ./bj.exe  -d junk
```

Multiple Players

A game of blackjack has multiple players. One player is always the dealer. There must be at least one other player. The other players play against the dealer and NOT against each other. A good web site to show how the players play against the dealer can be found at:
http://www.greatdaygames.com/games/play/blackjack/blackjack.aspx
This web site includes many things about blackjack that we are ignoring (like betting for different amounts of money/points, allowing the user to "split" or "double", or keeping track of win/lose amounts for multiple hands).

The number of non-dealer players in the modified program will be given using a command line argument. The program is to default to having 1 player play against the dealer. If the command line argument/flag of –p is given, immediately following this command line argument another command line argument will be given indicating the number of non-dealer players to use. Your program must allow up to 6 players to play against the dealer (for a total of 7 players including the dealer). If this second command line argument is not an integer value of 1 or greater, print out a meaningful error message and quit the program. If this second command line argument is an integer value greater than 6, you can either allow this number of non-dealer players or print out a meaningful error message and quit the program. For the following examples, assume the executable program is stored in a file called bj.exe in the current directory and $ is the command prompt.

Examples using 1 non-dealer player:

```
$ ./bj.exe
$ ./bj.exe  -p 1
```

Examples using multiple non-dealer players (3 in the first example 5 in the second):

```
$ ./bj.exe   -p  3
$ ./bj.exe  -p  5
```

Examples of an error (no value after the –p flag, value is not >= 1, value is not an integer):

```
$ ./bj.exe  -p
$ ./bj.exe  -p 0
$ ./bj.exe  -p junk
```

Use of Multiple Command Line Arguments

Your program must allow the user to specify both the –d and the –p command line flags in the same command line argument list. The –d flag can come before or after the –p flag; however, IMMEDIATELY following the flag must be the integer number specifying the number being used for that flag.

Examples using 2 decks of cards and 4 players:

```
$ ./bj.exe  -d 2 -p 4
$ ./bj.exe  -p 4 -d 2
```

Examples of errors:

```
$ ./bj.exe   -d -p  3 5
```

Program Submission

You are to submit the programs for this lab via the Assignments Page in Blackboard.

To help the TA, name your file with your net-id and the assignment name, like:

- ptroy1LabX.c