

# CS250 Assignment 3

## In-Between: the card game

**Date assigned:** Monday, February 23, 2015

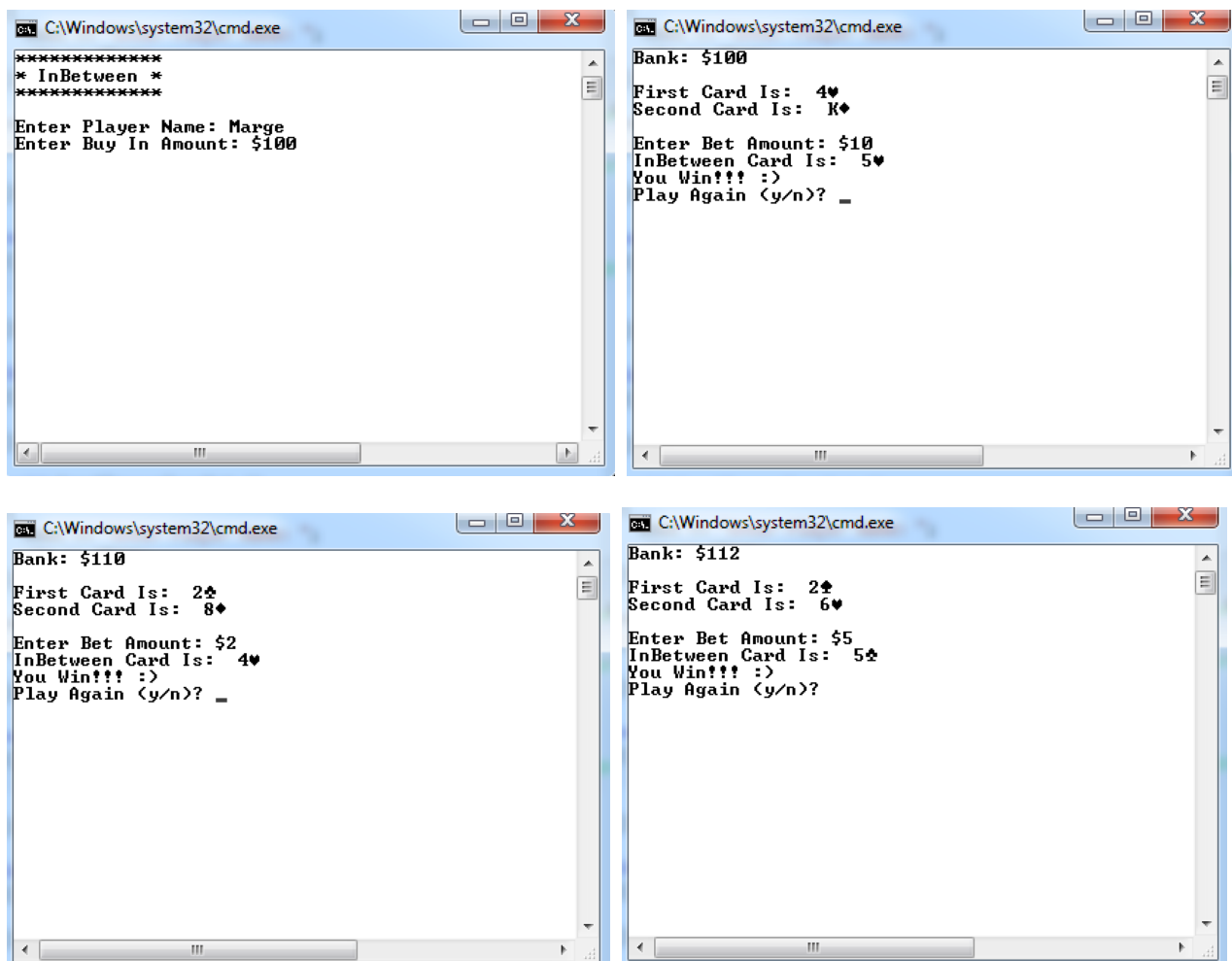
**Date due:** Wednesday, March 4, 2015

**Points:** 35

In-Between (AKA Acey Duecy) is a card game, typically played with multiple players, where two cards are displayed face up and players bet on whether the next card to be displayed falls in between the two cards. ([http://en.wikipedia.org/wiki/Acey\\_Ducey\\_%28card\\_game%29](http://en.wikipedia.org/wiki/Acey_Ducey_%28card_game%29)).

You are to implement a single player version of this game. Two cards will be displayed, and the user will place a bet on whether the next card will fall in between the two cards. If the third card does fall between the first two cards, then the user will win ☺; otherwise, they lose their bet ☹

**Example:**

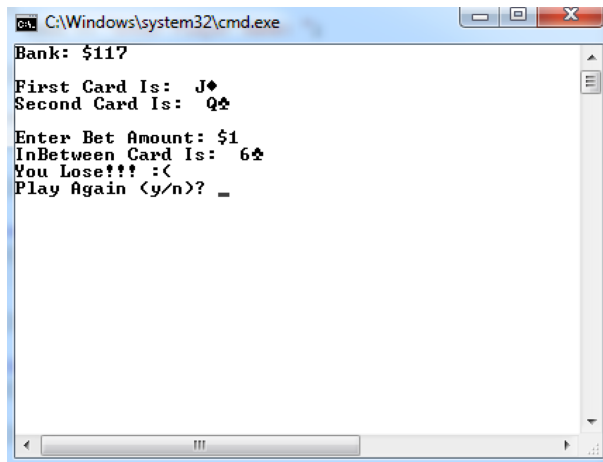


```
C:\Windows\system32\cmd.exe
*****
* InBetween *
*****
Enter Player Name: Marge
Enter Buy In Amount: $100

Bank: $100
First Card Is: 4♥
Second Card Is: K♦
Enter Bet Amount: $10
InBetween Card Is: 5♥
You Win!!! :)
Play Again (y/n)? _

C:\Windows\system32\cmd.exe
Bank: $110
First Card Is: 2♠
Second Card Is: 8♦
Enter Bet Amount: $2
InBetween Card Is: 4♥
You Win!!! :)
Play Again (y/n)? _

C:\Windows\system32\cmd.exe
Bank: $112
First Card Is: 2♠
Second Card Is: 6♥
Enter Bet Amount: $5
InBetween Card Is: 5♠
You Win!!! :)
Play Again (y/n)? _
```



```
C:\Windows\system32\cmd.exe
Bank: $117
First Card Is: J♦
Second Card Is: Q♣
Enter Bet Amount: $1
InBetween Card Is: 6♣
You Lose!!!! :(<
Play Again (y/n)? _
```

### Notes on gameplay:

- The game begins with the player buying into the game. The player must choose a value greater than 0.
- Next, two cards are dealt and displayed. When displaying the cards, always display the lowest card first and the highest card second. The card values from lowest to highest are 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K, A.
- Aces are always worth 10.
- The same card may be displayed twice!
- After both cards are displayed, the player places a bet. The minimum bet amount is \$1 and the maximum is the floor of half of the balance. All bets are to be whole numbers.
- Once the bank balance reaches zero, then the game should display the message “no more bets allowed” and exit.
- Let's assume that a player starts with a balance of \$100.
- If the player bets \$10 and wins, the resulting balance will be \$110 as the player wins their bet.
- If the player bets \$10 and loses, the resulting balance will be \$90.
- Play continues until the player decides not to play again or goes broke.

### Notes on design:

You are to use object-oriented design using two classes (Card and Gameplay). I have attached the interface of the two classes at the end of this assignment and placed them in the CS 250 public folder.

- For each class you will need a .h and a .cpp. You will also need a driver containing main function. Your project must have 5 files in total (Card.h, Card.cpp, Gameplay.h, Gameplay.cpp, and InBetween.cpp).
- You might find it helpful to set some constant arrays to store values for denomination, suit, value, and position. Place these constants at the top of Card.cpp.
- Your driver (InBetween.cpp) will create objects of those classes and use the objects to play the game.

### Goals for this assignment

1. Write an object-oriented program using two classes.
2. Practice using enums
3. Practice modular programming by using well-defined functions.

**To complete this assignment you must submit the following:**

**1. An electronic copy of your program on Grace**

- a) Add a project named **03\_InBetween** to your assignment solutions folder. It is vital that you name your solution and your project correctly!
- b) Type your program (fully documented/commented) into the project. You need to follow the coding standards from the CS250 Web page. These coding standards have been modified to include additional C++ language features introduced in CS250, so please be sure to read the new coding standards. Make sure that you include the hours you worked on the assignment in your header comments.
- c) Pay attention to the example output. Your program's output must look **exactly** like the sample output. The spacing and newlines in your output must match exactly.
- d) Make sure that your program builds without errors & warnings and runs correctly. If you get any errors or warnings, double check that you typed everything correctly. Be aware that C++ is case-sensitive. You will lose 10% if there are any warnings and 70% if your program does not build successfully.
- e) Once you are sure that the program works, it is time to submit your program. You do this by logging on to Grace and placing your complete solution folder in the **CS250-01 Drop** folder.
- f) The solution must be in the drop folder by the time class starts on the day the assignment is due. Anything submitted after that will be considered late.

**2. A hard copy of your program**

- a) The hard copy must be placed on the instructor's desk by the time class starts on the day that it is due.
- b) The hard copy must be printed in color, double-sided, and stapled in the upper left corner if your solution contains multiple pages.
- c) Your tab size must be set to 2 and you must not go past column 80 in your output.

**Remember, if you have any problems, come to me straight away with your project on a flash drive or on Grace. Good Luck!!!! ☺**

```

#ifndef CARD_H
#define CARD_H

class Card
{
public:

    static enum Suit {HEARTS = 0, DIAMONDS, CLUBS, SPADES};
    static enum Denomination {TWO = 0, THREE, FOUR, FIVE, SIX, SEVEN, EIGHT,
                               NINE, TEN, JACK, QUEEN, KING, ACE};

    // constructor to initialize the card to an initial value
    Card ();

    // set the mCardId to a random number between 0-51 inclusive
    // must also call setCard to set mSuit and mDenomination
    void draw ();

    // return the value of the suit
    Suit getSuit ();

    // print the suit to the screen using '\3', '\4', '\5', '\6'
    void printSuit ();

    // return the denomination
    Denomination getDenomination ();

    // print the denomination to the screen
    void printDenomination ();

    // print both the denomination and the suit
    // reuse functions!
    void printCard ();

    // return the value of the card
    // 2, 3, 4, 5, 6, 7, 8, 9, 10, 10, 10, 10, 10
    int getCardValue ();

    // return the position of the card within a suit
    // 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13
    int getPositionValue ();

    // return mCardId
    int getCardId ();

private:
    int mCardId;
    Suit mSuit;
    Denomination mDenomination;

    // set the value of mSuit based on the value of mCardId
    void setSuit (Suit);

    // set the value of mDenomination based on the value of mDenomination
    void setDenomination (Denomination);

    // set mCardId, mSuit, and mDenomination
    // reuse functions!
    void setCard (int);
};

#endif

```

```

#ifndef GAMEPLAY_H
#define GAMEPLAY_H

#include "Card.h"
#include <string>

using namespace std;

class GamePlay
{
public:
    GamePlay ();

    // clear the screen between plays
    void clearScreen ();

    // display the heading
    void printHeading ();

    // display a message to enter the name and read it in
    void enterName ();

    // return mName
    string getName ();

    // display a message to enter an initial buy in amount
    void enterBank ();

    // return mBank
    int getBank ();

    // allow the user to place a bet amount. Validate the bet amount
    void placeBet ();

    // display the message to play again. Keep asking while user does not
    // input y or n
    char playAgain ();

    // display the passed in message and the passed in card
    void printCard (string message, Card cCard);

    // reorder cCard1 and cCard2 so that cCard1 is the smaller value and
    // cCard2 is the larger one
    void orderCards (Card &cCard1, Card &cCard2);

    // update mBank and mbIsWinner depending on whether the player won or not
    void updateResult (Card cCard1, Card cCard2, Card cInBetweenCard);

    // display a results message. Either:
    // - You Win!!! :)
    // - You Lose!!! :(
    void printResult ();

private:
    string mName;
    int mBank;
    int mBet;
    bool mbIsWinner;
};

#endif

```