# PROG 10082 – Assignment #3

**Due Date:**

Thursday, November 6 @ 11:00 pm
This assignment is worth **4%** of your final grade.

**Description:**

This assignment is based on material covered up to and including while loops with a focus of decision and repetition structures. You will be completing a Java program that is a simplified version of blackjack called Sheridan21.

**Rules:**

- In Sheridan21, the goal is for the player's total value of all cards to be greater than the dealer's total value of all cards without going over 21. If the player's total is over 21 then the player "busts" and the **dealer automatically wins**. If the player decides to stop drawing cards with a total less than or equal to 21, the dealer will draw cards until the dealer's total is at least 17. After the total of all the dealer's cards is at least 17, the game concludes as follows:
    - If the dealer's total is greater than 21 then the **PLAYER WINS**
    - If the dealer's total is less than or equal to 21 and **less than** the player's total then the **PLAYER WINS**
    - If the dealer's total is less than or equal to 21 and **greater than** player's total then the **DEALER WINS**
    - If the dealer's total is less than or equal to 21 and **equal to** the player's total then the **PLAYER AND DEALER TIE (known as a PUSH)**
- Use the java.util.Random class to generate a random integer between 1 and 10 to represent a card being drawn. In Sheridan21, ***Aces will always have a value of 1*** and there are no Jacks, Queens, or Kings. All other cards have a point total matching their numerical value.

**Gameplay:**

1. The player initially receives two cards, while the dealer initially receives one card.
2. While the player's total is less than or equal to 21 and the player has not decided to stop taking cards
    - The player is asked whether she would like another card:
        - If the player says yes, then another card is drawn and is added to the player's total (by generating another random number between 1 and 10)
            - If the new card results in the player's total being greater than 21, then the **DEALER WINS** and the game immediately ends.
        - If the player says no, then move to step 3.
3. The dealer will repeatedly draw cards until the dealer's total is at least 17. Once the dealer's total is at least 17 the game will end with one of the following results:
    - If the dealer's total is greater than 21 then the **PLAYER ALWAYS WINS**
    - If the dealer makes a hand (has a total between 17 and 21), the result is as follows:
        - If the dealer's total is less than the player's total then the **PLAYER WINS**
        - If the dealer's total is greater than the player's total then the **DEALER WINS**
        - If the dealer's total is equal to the player's total then the **PLAYER PUSHES**

Below are examples of program output when playing Sheridan21 (text in **black** represents user input):

**Example 1 (Dealer wins, player busts):**

```
> run Sheridan21
Player draws 8
Dealer draws 1
Player draws 6
Player total is 14
Would you like another card? (yes/no)
yes
Player draws 9
Player total is 23
Sorry you busted. Dealer wins!
```

**Example 2 (Dealer wins, player doesn't draw, dealer total higher than player total):**

```
> run Sheridan21
Player draws 9
Dealer draws 10
Player draws 9
Player total is 18
Would you like another card? (yes/no)
no
Dealer draws 10
Dealer total is 20
Dealer wins!
```

**Example 3 (Player wins, player draws two cards, dealer busts):**

```
> run Sheridan21
Player draws 7
Dealer draws 9
Player draws 4
Player total is 11
Would you like another card? (yes/no)
yes
Player draws 5
Player total is 16
Would you like another card? (yes/no)
yes
Player draws 3
Player total is 19
Would you like another card? (yes/no)
no
Dealer draws 3
Dealer total is 12
Dealer draws 4
Dealer total is 16
Dealer draws 6
Dealer total is 22
Player wins!
```

**Example 4 (Dealer wins, player draws two cards, dealer total is higher than player total):**

```
> run Sheridan21
Player draws 3
Dealer draws 10
Player draws 3
Player total is 6
Would you like another card? (yes/no)
yes
Player draws 2
Player total is 8
Would you like another card? (yes/no)
yes
Player draws 10
Player total is 18
Would you like another card? (yes/no)
no
Dealer draws 5
Dealer total is 15
Dealer draws 5
Dealer total is 20
Dealer wins!
```

**Example 5 (Player push (tie), player draws one card, dealer total equals player total):**

```
> run Sheridan21
Player draws 1
Dealer draws 6
Player draws 10
Player total is 11
Would you like another card? (yes/no)
yes
Player draws 8
Player total is 19
Would you like another card? (yes/no)
no
Dealer draws 5
Dealer total is 11
Dealer draws 8
Dealer total is 19
Player push!
```

**Example 6 (Player wins, player draws no cards, player total greater than dealer total):**

```
> run Sheridan21
Player draws 9
Dealer draws 7
Player draws 10
Player total is 19
Would you like another card? (yes/no)
no
Dealer draws 10
Dealer total is 17
Player wins!
```

**Notes:**

- Your program should use self-documenting variables of the appropriate type.
- For reference, here is a link to the rules of blackjack: http://wizardofodds.com/games/blackjack/
- For simplicity, some of the standard rules of blackjack do not apply to Sheridan21:
    - Aces always have a point value of 1 (there is no option to choose whether an ace has a point value of 1 or 11)
    - There are no "face cards" (jacks, queens, kings)
    - This is a friendly game, there is no betting! Therefore the player cannot do things such as split or double-down

**Submission:**

You will be submitting one Java file, **Sheridan21.java**. Your program must have the following structure:

```
/*
     Name: [Your Name]
     Assignment 3
     Due Date: [Date Due]

     Description: [A brief description of what this program does –
                indicate here whether you have implemented the bonus]
*/

public class Sheridan21 {
     public static void main(String[] args) {
          // Implement your game
     }
}
```

Testing is a vital and often overlooked component of the software development process. Don't forget to test your program numerous times so that each scenario is handled correctly!

**Bonus (5% maximum):**

Modify your program so that after each game, the user has the option to play again or quit. If the user quits, thank the user for playing. Please indicate in the comments if you decide to implement the bonus.

**Evaluation:** This assignment is out of a total of **20 marks** with the following breakdown:

| | |
|---|---|
| Correct game logic | **/15** |
| User input/program output | **/3** |
| Coding conventions | **/2** |
| **TOTAL** | **/20** |