Programming Project 1
50 Points
References: Referenced Textbook and Week 2, 3, & 4 handouts
Due date: Monday, Feb 13 2017, by 11:59pm

Skills Required:

1. Writing code that controls a user interface
2. Primitive data types
3. The use flow controls and control loops
4. Iterations: For, While, Do .. While statements
5. Controls: IF … Else and /or Switch statements
6. Simple message dialogs
7. Calling methods in the predefined Java classes
8. Displaying messages in a dialog boxes
9. Formatting output

Description:
You will write a java program simulates a Blackjack game with given rules. You need to precisely follow the specified rules in this project in your implementation in order to correctly simulate the game.

You must follow the exact rules to receive full credit on this program, as the code will be tested iteratively to satisfy the program requirements.

Task Specifications:

1. Create a new java project and name it Project PP1.
2. Inside the PP1 project, create a Blackjack class with a main method that prompts the user for input, and will run a simulated game over and over until the user types the word "STOP" in upper, lower, or mixed case. All user commands are shown in upper case, but any combination of upper, lower, or mixed case should work for all commands.
3. Create an application that plays a card game according to these rules. This game is similar to "Blackjack" but the cards have values of 1 through 11.
    a. First, write a code segment that deals a card with a random integer between 1 and 11 inclusive.
    b. The object is to accumulate cards having a sum of 21 or less.
    c. The player wins the game if the total of their cards is more than the total of the dealer's cards.

    d. The player goes first, and may choose as many cards as they want until they either choose to stop or their total exceeds 21.
    e. The player can only see one of the dealer's cards until the player has finished accepting new cards.
    f. The dealer wins a tie game.
    g. If the player chooses cards that have a total greater than 21 (called a "bust") then the game is over immediately, and the dealer shall not receive more than their first two cards.

4. Display the results of the deal to the user. Update this display after each new card is dealt.
5. Request input from the user. Use a dialog box that require the user to type: like a showInputDialog. If the user types "HIT" then a new card will be dealt using your code that generates a random number between 1 and 11 inclusive, and the results displayed. This choice will be offered until the user types "STAY" or the user is "BUSTED" as described below.
6. If the user enters "STAY" before they are "BUSTED" then the game will resolve the dealer total, and display the results of the game.
7. If after typing "HIT," the value of player is more than 21, the game terminates and displays "BUSTED." In this case, the value of the dealers numbers are resolved and displayed to the user before the game terminates.
8. When resolving the dealer's total, the application will "HIT" the dealer for totals of 17 or less. The dealer will "STAY" on totals of 18 or more. If the value of dealer becomes greater than 21, then the application terminates and displays "DEALER BUSTED." Otherwise, the application compares the final values of dealer and player, and reports the winner.
9. Note that it is possible to get two cards with a total of 22 on the initial deal. When this happens, regenerate the random numbers for the player or dealer so that they are not busted at the start of each game.
10. When the game terminates, the values of dealer and player should return to zero
11. The application should continue to play new games until the user types the text "STOP" when asked if they want to play another game.

Evaluation Criteria

1. All tasks must be completed to receive credit for this assignment
2. Program should not crash if the user enters incorrect data

Submission:

Copy the .java source files from the *src* folder in your *work space* to another folder that must be named following the provided naming format in this course, then zip and upload the file under this assignment answer in Canvas. You must read the project demo document in Canvas before the review time.

File Name: *FLLLLPP1.zip (F = first letter in your first name and LLLL = your last name)*

Grading Rubric – PP01

Name: ____

Requirements	**Comments**	**Max Points Allocated**	**Points Earned**
General Code Structure: Coding of the Java classes, Blackjack class with the expected methods (3) Use of Java comments in the source code (2) Indentations, good variable names/class members (2) Successful compilation (no compile error) and running of code (able to execute the program) (5)		12	
Input, Output, User Interface: Exception handling of the input values (e.g., program will not crash is no value is entered, empty space is entered, invalid value is entered) (3) Prompting user for various input values (3) Use of upper, lower or mixed case for STOP (3) Proper implementation of STAY, BUSTED, HIT (3) Use of dialog boxes, as appropriate (3) Formatting of the output values (3)		18	
Functionality: Generating random values from 1 through 11 inclusive (3) Accumulate cards having a sum of 21 or less (3) Logic for dealer winning the game and that of bust (3) When the game terminates, the values of dealer and player returns to zero (3) Application should continue to play new games until the user types the text "STOP" (3) Regenerate the random numbers for the player or dealer		30	

so that they are not busted at the start of each game (3) Calling methods, as appropriate (3) Use of conditional logic, as appropriate (5) Use of loop, as appropriate (4)			
Total		**50**	

Total: _____/50