Assignment 2 Peer Reviews Due: Thursday, 1/24/19, 11:59pm Assignment 2 Back Evaluations: Sunday, 1/27/19, 11:59pm

Go to Peerceptiv and complete your 3 peer reviews by Thursday at midnight. After Thursday at midnight, you will evaluate how helpful your peer reviews are by Sunday at midnight.

Assignment #3: Conditionals and Loops (150 pts) Design Due: Sunday, 1/27/19, 11:59pm Design Peer Reviews Due: Thursday, 1/31/19, 11:59pm Code and Design Back Evaluations: Sunday, 2/03/19, 11:59pm

Make sure you demo Assignment #2 within two weeks of the due date to receive full credit. If you go outside the two-week limit without permission, you will lose 50 points. If you fail to show up for your demo without informing anyone, then you will automatically lose 10 points.

(10 pts) Assignment 2 Reflective Post-Peer Review Survey: http://oregonstate.qualtrics.com/jfe/form/SV_4YmgauZa5JyDgFf

Problem Statement

We are going to create a game called TwentyOne, which is similar to the Black Jack and War card games. This game has a dealer/computer and 1-4 players, and the game will generate random numbers from 1 to 11 to simulate the values in cards. All players start with a specified amount of money, which they determine at the start of the game, and the goal is to see if you can get closer than the dealer to twenty-one points without going over.

With each player's turn, they first bet against the dealer, and then play against the dealer before moving on to the next player. An initial random number, 1-11, is generated, and after each random number is generated, the player decides whether they want to generate another to add to their total points trying to get as close to 21 as possible without going over. The dealer is also given random numbers, but the strategy you use for determining whether they get another random number is your choice. (3) The rules for the game play are below.

- If the player busts, the player automatically loses their bet, and the dealer doesn't need to play against that player.
- If the dealer busts, the player automatically wins their bet.
- If the dealer ties with the player, then the player doesn't win or lose their bet.
- If the player's bet is more than their bank, then you need to continue to re-prompt for a new bet until a good value is given.
- If a player's bank is 0, they get skipped.
- If all players' banks are 0, then the game is automatically over!
- After all players take a turn against the dealer, then the players can collectively cash out or play again.

Example game play below:

How many players do you have (1-4)? 2

Player 1, how much money are you starting with? 20.00 Player 2, how much money are you starting with? 50.00

Player 1, how much do you bet you can beat the dealer? 10.00 Player 1, you got 10. Do you want to go again (0-no or 1-yes)? 1 Player 1, you got 8. Do you want to go again (0-no or 1-yes)? 0 Your total points are 18.

The dealer gets 11. The dealer gets 11. The dealer busts! You win! Player 1 has 30.00.

Player 2, how much do you bet you can beat the dealer? 10.00 Player 2, you got 8. Do you want to go again (0-no or 1-yes)? 1 Player 2, you got 9. Do you want to go again (0-no or 1-yes)? 0 Your total points are 17.

The dealer gets 8. The dealer gets 9. The dealer has 17 points. It's a tie! Player 2 has 50.00.

Do you want to play again (0-no or 1-yes)? 0

Design Document – Due Sunday 1/27/19, 11:59pm Refer to the Example Polya Document - Polya template.pdf

Step 1: Understanding the Problem/Problem Analysis. (15 pts)

Do you understand everything in the problem? List anything you do not fully understand, and make sure to ask a TA or instructor about anything you do not understand.

- What are the user inputs/requirements, program outputs, etc.? (5 pts)
- What assumptions are you are making? (5 pts)
- What are all the tasks and subtasks in this problem? (5 pts)

Step 2: Program Design. (25 pts)

- What does the overall big picture of this program look like? (flowchart or pseudocode) (20 pts)
 - What data do you need to create, when do you read input from the user, what is the dealer strategy going to be?
 - What are the decisions that need to be made in this program?
 - What tasks are repeated?
- What kind of bad input are you going to handle? (5 pts)

Based on your answers above, list the **specific steps or provide a flowchart** of what is needed to create this program to play TwentyOne. Be very explicit!!!

Step 4: Program Testing. (10 pts)

Create a test plan with the test cases (bad, good, and edge cases). What do you hope to be the expected results?

• What are the good, bad, and edge cases for ALL input in the program? Make sure to provide enough of each and for all different inputs you get from the user.

Program Code – Due Sunday 2/03/19, 11:59pm

(80 pts) Write the program to play the above game.

- Your user interface must provide clear instructions for the user and information about the data being presented.
- You are not allowed to use global variables (those declared outside of many or any other function).
- Your program must handle bad input of the same type expected from the user, e.g. user enters 3 instead of 1 or 0, but you do not have to handle bad input of a different type, e.g. user enters 3.4 instead of 1 or 0.
- You must have at least 2 functions.
- Make sure you follow the style guidelines, have a program and function headers with appropriate comments, and be consistent.

(10 pts) Extra Credit

Handle *all* bad user input, even data of the wrong type.

(10 pts) Extra Credit 2

Change the game play so that the dealer plays against all players in a turn, rather than playing only again one player at a time.

(10 pts) Assignment 3 Predictive Pre-Peer Review Survey:

http://oregonstate.qualtrics.com/jfe/form/SV_1ApIXp0UOpcxWrX

Electronically submit your Design Document by the design due date and your C++ program (.cpp file, not your executable!!!) by the code due date using Peerceptiv.