CIS 830: Homework Assignment 2

Introduction to Python Programming, Spring 2020

Due Date: Thursday, March 19th by 6:00 PM

Instructions:

- Please include your name on everything you turn in.
- Please attempt to complete all parts of the assignment.
- Please include comments in your code; it is a good habit and makes it easier to read
- If you are unfamiliar with playing cards, suits and ranks, please look at this Wikipedia page: https://en.wikipedia.org/wiki/Standard_52-card_deck#Rank_and_color

Part 1: Design A Basic Guess-a-Card Game (**3 chances, no hints, no suits**)

Using what we have learned so far, design a Python code that randomly chooses a card and gives the player (you!) three chances to guess that card correctly. In this part, the card should not have a suit and the player should not be given hints. If the player guesses the correct card within three turns, the code must tell them that they have won. If not, the code must tell them that they have lost.

Steps:

1. Create a Python program titled "HW2_Part1_YOURNAME.py"
2. At the top of the program, type your "pseudocode" as a multi-line comment.
    a. Multi-line comments begin and end with three quotes (" " " … " " ")
    b. Pseudocode is just meant to help you organize your thoughts before you have to think about programming syntax. Since it will be written as a comment, Python will not attempt to execute it.
3. After the multi-line comment, write the actual Python code.
    a. It should be able to produce both outcomes (player wins or loses) without displaying any errors.
4. Upload your program to Canvas in "Homework Assignment 2, Part 1" by the due date.

Part 2: Design an Advanced Guess-a-Card Game (**unlimited chances, hints, all 4 suits**)

Build upon the code you made in Part 1 to make the Guess-a-Card game more complex. Now, the randomly chosen card should have a randomly chosen suit. After each guess, the code should print two hints: 1) if the player has or has not guessed the correct suit and 2) if the rank the player guessed is too high, too low, or correct. The code should allow the player to continue guessing until they get the correct answer (this will require a loop).

Steps:

1. Create a Python program titled "HW2_Part2_YOURNAME.py"
2. At the top of the program, type your "pseudocode" as a multi-line comment.
3. After the multi-line comment, write the actual Python code.
    a. It should reach its only outcome (player wins) regardless of how many guesses are needed and without displaying any errors.
4. Upload your program to Canvas in "Homework Assignment 2, Part 1" by the due date.