

Kerry Ojakian's CSI 31 Class

Due Date: Sunday December 10 by 11pm in Dropbox

HW #3

General Instructions:

- Homework must be put in a your dropbox folder in subfolder called HW04.
- **Create a single Jupyter notebook and put these problems in there in order. Before a problem, put a header and leave space between problems.**

The Assignment

1. Using Tkinter, make a drawing of a face (keep it short and simple!). Your program must use at least the following methods of the Canvas class (all have exactly analagous syntax to `create_oval` as done in class): `create_rectangle`, `create_oval`, `create_line`.
2. For both of the following problems use the events '`< Enter >`' and '`< Leave >`'. Note - these 2 problems are short modifications of programs done in class, but we are binding events of entering and leaving instead of right clicking.
 - (a) Use tkinter to create a program that does the the following. A window opens with one Label widget. When the mouse is moved inside the Label, the word "INSIDE" should be printed to the console. When the mouse is moved outside the Label, the word "OUTSIDE" should be printed to the console.
 - (b) Modify the last program so that rather than printing to the console, the actual text in the Label changes so that it is "INSIDE" when the mouse is inside the label and "OUTSIDE" when the mouse is outside the Label
3. Ch. 9 Programming exercise #9.3 (page 328-329). Actually write the Student class in a file. Hint: Both methods in the child class should call the corresponding parent method first.

4. Use the Card class from our class assignment (if yours did not work, feel free to copy one from myself or a student). You will create a Deck class, which will use the Card class (i.e. a Has-A relationship). Draw a class diagram.

Create a class Deck that represents a deck of cards. Your class should have the following methods:

- constructor: Creates a new deck of 52 cards in a standard order.
- shuffle: Randomizes the order of the cards.
- dealCard: Returns a single card from the top of the deck and removes the card from the deck.
- cardsLeft: Returns the number of cards remaining in the deck.

Test your program by having it deal out a sequence of n cards from a shuffled deck where n is a user input. Try with various values of n to see that it works. And test the other methods.