CS2321 Lab 4

Lab Instructions:

Save the code you write for each exercise in this lab as a *library* -- that is, a textfile with a .py extension containing only executable python code (i.e. no angle-bracket prompts, etc). Name each file according to the exercise number (e.g. ex1.py, ex2.py, etc.) and save them to a directory containing the report file, when completed, compress them together in a single zip file to be submitted on D2L.

For this lab, make sure you include breezypythongui.py, your python modules, and image files (in their folders), all compressed to the zip file.

Each function should have a docstring explaining what the function does.

Any Follow-up Questions and their Answers should be included in a **docstring** in the `main()` function.

e.g. the structure for a module contains a GUI class should look like:

```
'''
   poker.py
   Doc-string explaining what this module does
'''

from breezypythongui import EasyFrame

# Other imports, such as math, random, etc., as needed

class Poker(EasyFrame):
   '''
       This is the doc-string explaining what this class does.
   '''
   def __init__(self):
       '''
            Doc-string for each of the methods, including the constructor.
       '''
       EasyFrame.__init__(self, title = 'A proper title to be displayed')

       #Create and place widgets properly.

   #definitions of event-handling methods, make sure the first parameter is
      self.
   def method1(self, ......):
       '''
            Doc-string for each of the methods, including the constructor.
       '''
   def method2(self, ......):
       '''
            Doc-string for each of the methods, including the constructor.
       '''
```

```
def main():
      Poker().mainloop()

if __name__ == "__main__":
      main()
```

Lab Deliverable: Once all your programs run correctly, collect their code and the results of their test-cases in a nicely-formatted **PDF** file exported from Word Processing document (e.g. MS Word or LibreOffice are fine) to be included in the submission on D2L.

This **report** should consist of each lab exercise, clearly **labeled** in order, consisting of code, then copy/pasted text output, or, for GUI, screen-captured, of its four test-cases.
In this lab, take series of screen captures of your GUIs and insert them into the report.

Paired Programming:
We will work today's lab assignments in pairs -- on a single computer in one partner's account. One person will start out as the *typist*, the other as the *verifier*. These roles will switch.
For each problem, partners should decide upon their proposed algorithm to solve the given problem *before* the typist begins to type. Sketch it out on a sheet of paper, perhaps.
For **ten** minute periods, the typist will type the code, while the other verifies and suggests corrections (typist has final decision). Under no circumstances may the verifier ever touch the mouse or keyboard. (Note: the *instructor* may not touch your input devices either!)
On the instructor's ten-minute signal, partners will trade responsibilities. This should allow both partners to benefit from each other's strengths. Future paired-programming labs will be with different partners, to spread the gained experience around.
Each partner should post the resulting code in their own D2L folder. You may transmit partnership-generated code to the other partner (only!) by email or thumb-drive.

Exercise

This project will build a GUI based Poker (simplified) game, It should have a **Poker** class to hold the controls, and display cards, scores, etc.

It should also controls the deck of cards and two hands (lists of cards), one for the user, another for the computer.

Each hand has five cards, and Card is defined as a class. You can modify the existing Card class to include the card's image name.

So these are suggested classes: **Poker**, **Deck** and **Card**.

The rules can be as simple as comparing the values of the five cards from each hand to determine who wins the Hand, and 2 out of 3 Hands determines the winner of the Game.

You can definitely go further to apply some well known game rules.

Images of cards, 52 face images and one back image will be included.