Worksheet 1

Miscellaneous Exercises

Given the following class definition, answer the following questions. It is VERY IMPORTANT to try to answer the questions without running the code through the interpreter. Run the code only to verify your answers.

```
class Person:
  def __init__(self, name="Messi"):
    self.name = name
```

Problem 1:
Does the following code result in an error? if yes, why? and if no, what is the output?

```
p = Person()
print(p.name)
```

Problem 2:
What is the output of the following code?

```
p = Person()
p.name = "CR7"
print(p.name)
```

Problem 3:
What is the output of the following code?

```
p = Person("CR7")
print(Person().name)
```

Continue to the next page...

Problem 4:
For the following Python expressions, answer whether they evaluate to True or False.

```
p1 = Person(), p2 = Person(); p1 is p2 # True or False?
p1 = Person("CR7"); p2 = Person("CR7"); id(p1) != id(p2) # True or False?
id(Person) == id(Person()) # True or False?
```

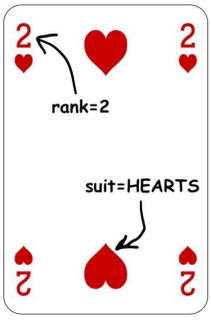
Capstone Project: MaxHandWins

Problem 5:
As an exercise, try to brainstorm all the different objects involved in MaxHandWins. There is not one single solution to this and there is usually more than one way to design your classes.

Problem 6:
Create a class `PlayingCard` that represents a playing card object. A playing card can be defined by two attributes:

- **`rank`:** This is the number that you see on the playing card.
- **`suit`:** This is one of four categories that represent the suit, or the shape that appears on the playing card. The four categories are <**SPADES**, **HEARTS**, **DIAMONDS**, **CLUBS**>

👉PRO TIP 👈
Can you use enum classes to represent the suit of a playing card?
More information about Enums can be found here.



Note:
Assume the following ranks for picture cards:
Jack => 11
Queen => 12
King => 13
Ace => 14

Problem 7:

One of the objects in MaxHandWins is the “player”. Define a class `Player` that has the following attributes:

- **`name`:** a string that represents the name of the player.
- **`hand`:** a list that represents the set of cards that this player has.

The `hand` attribute **MUST** be initialized to an empty list.

Problem 8:

Create a class that represents a Deck of Playing Cards. Let’s name this class `Deck`. For the sake of this problem and throughout the project, you can assume that we have a single 52-card deck. Class `Deck` has one attribute `cards` that is a list of `PlayingCard`s. At initialization, this list MUST be initialized with all 52 playing cards.