

## Python class exercise

### Hooman Ileayi

**Imported Restaurant:** Using your latest Restaurant class, store it in a module. Make a separate file that imports Restaurant. Make a Restaurant instance, and call one of Restaurant's methods to show that the import statement is working properly.

```
:class Restaurant
def __init__(self, name, cuisine_type):
    self.name = name
    self.cuisine_type = cuisine_type

def describe_restaurant(self):
    print(f"{self.name} is a {self.cuisine_type} restaurant.")

def open_restaurant(self):
    print(f"{self.name} is now open!")
```

OR

```
from restaurant import Restaurant

my_restaurant = Restaurant("Pizza Palace", "Italian")
my_restaurant.describe_restaurant()
```

**Imported Admin:** Start with your work from Exercise 9-8 (page 173). Store the classes User, Privileges, and Admin in one module. Create a separate file, make an Admin instance, and call show\_privileges() to show that everything is working correctly.

```
from user import Admin

my_admin = Admin("John", "Doe")
my_admin.privileges.show_privileges()
```

**Multiple Modules:** Store the User class in one module, and store the Privileges and Admin classes in a separate module. In a separate file, create an Admin instance and call show\_privileges() to show that everything is still working correctly.

```
from user import User

from admin import Privileges, Admin

my_admin = Admin("John", "Doe")

my_admin.privileges.show_privileges()
```

**Dice:** Make a class Die with one attribute called sides, which has a default value of 6. Write a method called roll\_die() that prints a random number between 1 and the number of sides the die has. Make a 6-sided die and roll it 10 times. Make a 10-sided die and a 20-sided die. Roll each die 10 times.

```
import random

class Die

def __init__(self, sides=6):
    self.sides = sides

def roll_die(self):
    print(random.randint(1, self.sides
```

**Lottery:** Make a list or tuple containing a series of 10 numbers and 5 letters. Randomly select 4 numbers or letters from the list and print a message saying that any ticket matching these 4 numbers or letters wins a prize.

```
import random

lottery_numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 'A', 'B', 'C', 'D', 'E']
```

```
winning_numbers = random.sample(lottery_numbers, 4)
```

```
print(f"Any ticket matching these 4 numbers or letters wins a prize:  
{winning_numbers}")
```

**Lottery Analysis:** You can use a loop to see how hard it might be to win the kind of lottery you just modeled. Make a list or tuple called `my_ticket`. Write a loop that keeps pulling numbers until your ticket wins. Print a message reporting how many times the loop had to run to give you a winning ticket.

```
import random
```

```
lottery_numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 'A', 'B', 'C', 'D', 'E']
```

```
my_ticket = [1, 'A', 5, 'B']
```

```
num_attempts = 0
```

```
:while True
```

```
num_attempts += 1
```

```
winning_numbers = random.sample(lottery_numbers, 4)
```

```
:if winning_numbers == my_ticket
```

```
break
```

```
print(f"It took {num_attempts} attempts to win the lottery with ticket  
{my_ticket}.")
```

**Python Module of the Week:** One excellent resource for exploring the Python standard library is a site called *Python Module of the Week*. Go to <https://pymotw.com> and look at the table of contents. Find a module that looks interesting to you and read about it, perhaps starting with the `random` module.

- `random()`: returns a random floating-point number between 0 and 1.
- `randint(a, b)`: returns a random integer between a and b, inclusive.
- `choice(seq)`: returns a random item from the given sequence.
- `shuffle(seq)`: shuffles the items in the given sequence in place.
- `sample(seq, k)`: returns a list containing k unique random items from the given sequence.

