

# CS 115 - Introduction to Programming in Python

## Lab Guide 06

---

### Lab Objectives: 2D Lists / Classes

---

1. a) Write a function **formMatrix** which gets an integer number **n** and creates and returns a square 2-dimensional matrix of size  $n \times n$  according to the below rule:
  - The elements in the **upper triangular part** of the matrix will be assigned to the sum of their row and column indices
  - The elements in the **lower triangular part** of the matrix will be assigned to the difference of their row and column indices
  - The elements in the **major diagonal** of the matrix will be assigned to 0

(Assume n is 5 in the sample below).

0	1	2	3	4
1	0	3	4	5
2	1	0	5	6
3	2	1	0	7
4	3	2	1	0

- b) Write a main to input the size of the matrix and display the matrix in matrix form after calling the above function.

#### Sample Run:

Enter size of the matrix: 6

0	1	2	3	4	5
1	0	3	4	5	6
2	1	0	5	6	7
3	2	1	0	7	8
4	3	2	1	0	9
5	4	3	2	1	0

2. Write a class called `Passenger` which represents a Passenger object. Write a class (`Passenger.py`) which represents this object.

a) The class will store the following attributes:

- *passengerName*: name of the passenger
- *passengerSurname*: surname of the passenger
- *seatNo*: seat number of the passenger in the plane
- *fare*: ticket price

a) Your class should have an `init()` method that takes the values of all four attributes as parameters. The default value for *fare* is 1000 if not given.

b) In addition, the class will have the following methods:

- *get* methods for the name, surname and seat number of the passenger.
- *set* method for updating the *seatNo* with the new seat number.
- *calculateFare*: if the fare of the Passenger object is less than 1000 TL, then the function returns the fare increased by 5%, else the method returns the normal fare amount.

c) In addition to the above methods, your class should define the following special methods:

- `__str__` method which returns a string as follows:

```
Passenger Name:      John Gutttag      Seat: 9F
```

- `__repr__` method which returns a string as follows:

```
(Pearson D.      9F      1000TL)
```

2. Create an application that does the following:

a. Write a function `load_passengers` which reads passenger info from **passengers.txt** and creates and returns a list of passenger objects.

b. Input name and surname of a passenger and if he/she exists in the list, input new fare and update his/her fare with the new fare. If passenger does not exist, give a message as in the sample run.

b. Display the list of all passengers.

### Sample Run 1:

Enter name: Ruya

Enter surname: Yilmaz

Enter new price: 1220

Passenger Name: Yilmaz Ruya Seat: 08D  
UPDATED

Passenger List:

```
[(Ozer A.      14A   1500TL)
, (Yuksel A.   15C   525.0TL)
, (Kose Tas E. 09B   1100TL)
, (Yalcin M.   04C   3100TL)
, (Aksoy Z.    18D   1500TL)
, (Turan F.   11A   1500TL)
, (Sen U.      A02   913.5TL)
, (Yilmaz R.   08D   1220TL)
, (Ates O.     21F   4600TL)
, (Keskin A.   20F   2280TL)
, (Tas O.      04C   829.5TL)
, (Aktas S.    15C   3200TL)
, (Yildiz Y.   01C   8100TL)
, (Demir M.    15D   2500TL)
, (Ozdemir A.  22B   367.5TL)
, (Ozturk A.   215   1950TL)
, (Cakir B.    21C   4150TL)
, (Polat M.    07C   1800TL)
, (Gunes F.    16F   1850TL)
, (Yuksel F.   02D   2670TL)
]
```

### Sample Run 2:

Enter name: Elif

Enter surname: Sonmez

Passenger is NOT Found!

Passenger List:

```
[(Ozer A.      14A   1500TL)
, (Yuksel A.   15C   525.0TL)
, (Kose Tas E. 09B   1100TL)
, (Yalcin M.   04C   3100TL)
, (Aksoy Z.    18D   1500TL)
, (Turan F.   11A   1500TL)
, (Sen U.      A02   913.5TL)
, (Yilmaz R.   08D   1140TL)
, (Ates O.     21F   4600TL)
```

```
, (Keskin A. 20F 2280TL)
, (Tas O. 04C 829.5TL)
, (Aktas S. 15C 3200TL)
, (Yildiz Y. 01C 8100TL)
, (Demir M. 15D 2500TL)
, (Ozdemir A. 22B 367.5TL)
, (Ozturk A. 215 1950TL)
, (Cakir B. 21C 4150TL)
, (Polat M. 07C 1800TL)
, (Gunes F. 16F 1850TL)
, (Yuksel F. 02D 2670TL)
]
```