

CS 115 - Introduction to Programming in Python

Lab Guide 08

Lab Objectives: Recursion. Searching and Sorting.

1. Download the class, `Passenger`, that represents a passenger on a Flight.
 - The class stores the following attributes: `passengerName`, `passengerSurname`, `seatNumber`, `fare`.

Note: seat numbers always start with 2-digit row number followed by the single character alphabetic seat.
2. Update the class, `Passenger` to include the following:
 - `__eq__` method: `Passengers` are equal if their first name, last name and seat number are the same.
 - `__lt__` method: A `Passenger` is less than another if their surname comes before the other alphabetically. If the `Passengers` being compared have the same surname, `Passengers` in lower row/seats are less than passengers in higher row/seats based on the alpha-numeric flight number. For example, `Passenger Smith` is less than `Passenger Wilson`. If both `Passengers` are `Smith`, the `Passenger` in seat `03F` is less than the `Passenger` in `04A`. If both `Passengers` are `Wilson`, the `Passenger` in seat `08A` is less than the `Passenger` in `08B`.
3. Create a script, `PassengerApp.py`, that includes the following functions:
 - `load_passengers()` : takes a filename as a parameter and returns a list of `Passengers` contained in the file. Each line in the file contains the data used to create a `Passenger` object. The list should not contain duplicate `Passengers`.
 - `min_index()` : Takes a list of `Passengers` and a starting index and returns index of the `Passenger` in the list with the lowest (minimum) seat number, starting from the starting index to the end of the list.
 - `sort_passengers_by_seat()` : takes a list of `Passengers` as a parameter and sorts the `Passengers` by `seatNumber` using the *selection sort algorithm*. You should make use of the `min_index` function defined above.
 - `search_passenger_by_seat()` : takes a list of `Passengers` and a `seatNumber` as parameters and finds and returns the `Passenger` in the given `seatNumber` (returns `None` if no matching `Passenger` found).
 - `search_passenger_by_surname()` : takes a list of `Passengers`, a surname and `n`, the number of elements in the list to search as parameters. Searches the `n` element list using a **recursive** linear search algorithm and returns a list containing all passengers with the matching surname.
4. Create an application that does the following:
 - Loads the `Passengers` in the file, `passengers.txt`, into a list and displays the list.
 - Sorts the list of `Passengers` by `seatNumber` and displays the sorted list.
 - Inputs a `seatNumber` from the user and displays the `Passenger` in the given seat. Display appropriate message if no matching `Passenger` found.
 - Inputs a surname and displays the `Passengers` with the given surname. Display appropriate error if no matching `Passengers` found.
 - Sorts the list using default sort (surname/seat number) and display the sorted list.

Sample Run:

duplicate - passenger Aysel Keskin 20F not added...

Original list of Passengers:

[Ozer, A. (14A) Fare: 1500.00TL
, Yuksel, A. (15C) Fare: 515.00TL
, Kose Tas, E. (unassigned) Fare: 1100.00TL
, Yalcin, M. (04C) Fare: 3100.00TL
, Keskin, A. (20F) Fare: 2280.80TL
, Aksoy, Z. (18D) Fare: 1500.00TL
, Turan, F. (11A) Fare: 1500.75TL
, Sen, U. (unassigned) Fare: 896.36TL
, Yilmaz, R. (08D) Fare: 11400.00TL
, Ates, O. (21F) Fare: 4600.00TL
, Tas, O. (unassigned) Fare: 813.70TL
, Aktas, S. (unassigned) Fare: 3200.00TL
, Yildiz, Y. (01C) Fare: 8100.00TL
, Demir, M. (15D) Fare: 2500.00TL
, Ozdemir, A. (22B) Fare: 360.50TL
, Ozturk, A. (unassigned) Fare: 1950.00TL
, Cakir, B. (21C) Fare: 4150.00TL
, Polat, M. (07C) Fare: 1800.00TL
, Gunes, F. (16F) Fare: 1850.25TL
, Yuksel, F. (02D) Fare: 2670.50TL
,
]

Passengers sorted by seat:

[Yildiz, Y. (01C) Fare: 8100.00TL
, Yuksel, F. (02D) Fare: 2670.50TL
, Yalcin, M. (04C) Fare: 3100.00TL
, Polat, M. (07C) Fare: 1800.00TL
, Yilmaz, R. (08D) Fare: 11400.00TL
, Turan, F. (11A) Fare: 1500.75TL
, Ozer, A. (14A) Fare: 1500.00TL
, Yuksel, A. (15C) Fare: 515.00TL
, Demir, M. (15D) Fare: 2500.00TL
, Gunes, F. (16F) Fare: 1850.25TL
, Aksoy, Z. (18D) Fare: 1500.00TL
, Keskin, A. (20F) Fare: 2280.80TL
, Cakir, B. (21C) Fare: 4150.00TL
, Ates, O. (21F) Fare: 4600.00TL
, Ozdemir, A. (22B) Fare: 360.50TL
, Ozturk, A. (unassigned) Fare: 1950.00TL
, Tas, O. (unassigned) Fare: 813.70TL
, Kose Tas, E. (unassigned) Fare: 1100.00TL
, Aktas, S. (unassigned) Fare: 3200.00TL
, Sen, U. (unassigned) Fare: 896.36TL
,
]

Enter seat to search: 16F

Gunes, F. (16F) Fare: 1850.25TL

Enter surname to search: Yuksel

List of matching passengers:

[Yuksel, F. (02D) Fare: 2670.50TL

, Yuksel, A. (15C) Fare: 515.00TL

]

Passengers sorted by surname/seat number:

[Aksoy, Z. (18D) Fare: 1500.00TL

, Aktas, S. (unassigned) Fare: 3200.00TL

, Ates, O. (21F) Fare: 4600.00TL

, Cakir, B. (21C) Fare: 4150.00TL

, Demir, M. (15D) Fare: 2500.00TL

, Gunes, F. (16F) Fare: 1850.25TL

, Keskin, A. (20F) Fare: 2280.80TL

, Kose Tas, E. (unassigned) Fare: 1100.00TL

, Ozdemir, Å. (22B) Fare: 360.50TL

, Ozer, A. (14A) Fare: 1500.00TL

, Ozturk, A. (unassigned) Fare: 1950.00TL

, Polat, M. (07C) Fare: 1800.00TL

, Sen, U. (unassigned) Fare: 896.36TL

, Tas, O. (unassigned) Fare: 813.70TL

, Turan, F. (11A) Fare: 1500.75TL

, Yalcin, M. (04C) Fare: 3100.00TL

, Yildiz, Y. (01C) Fare: 8100.00TL

, Yilmaz, R. (08D) Fare: 11400.00TL

, Yuksel, F. (02D) Fare: 2670.50TL

, Yuksel, A. (15C) Fare: 515.00TL

]