

CS 115 - Introduction to Programming in Python

Lab 05

Lab Objectives: Tuples, Lists, Dictionaries

Notes:

- Upload your solutions as **a single .zip file** to the Lab05 assignment for your section on Moodle. You must use the following naming convention: Lab05_Surname_FirstName.zip where Surname is your family name and FirstName is your first name.
- You should only use functionality covered in CS115 in your solution.
- Include a docstring for your functions.

- 1) Write a program (**Lab5Q1.py**) by completing the following:
- a. `fill_list()` : takes `n` and returns a list of `n` random integer values between 2 and 50.
 - b. `eliminate_values()` : takes a list as a parameter and keeps one copy of each element by eliminating the repeated values. **NOTE:** your function should update the list passed as a parameter and not create a new list.

Your program should use the functions defined above to produce the output shown below in the sample run. Your program should validate that the digit input is a single, odd digit.

Sample Run:

```
Enter the number of elements:30
Original list:
[19, 3, 11, 37, 10, 30, 12, 26, 39, 20, 24, 46, 13, 35, 25, 2, 14, 35, 35,
44, 32, 45, 42, 24, 17, 38, 15, 51, 20, 4]
List with multiples removed:
[19, 3, 11, 37, 10, 30, 12, 26, 39, 20, 24, 46, 13, 35, 25, 2, 14, 44,
32, 45, 42, 17, 38, 15, 51, 4]
```

- 2) Write a program `Lab5Q2.py`, which includes the following:
- a. Write a function `sumT` which receives a tuple and returns a new tuple containing only the numeric elements in the tuple passed as a parameter.
 - b. Write a script that initializes a tuple of mixed type values and using the `sumT` function, displays the new tuple containing numeric elements and also displays the sum of the tuple.

Sample Run:

```
Original tuple:
((3, 5), 5, False, 2.5, 'cs115', [1, 2, 3], 7.8)
New Tuple:
(5, 2.5, 7.8)
Sum of new tuple is 15.3
```

3) A car rental company stores a dictionary of car brands and their plate numbers. Write a Python program that does the following:

- Define a **function**, `import_cars()` that takes a file name and an empty dictionary as parameters. The function should add all car brands to the dictionary as keys, and for each key, a **tuple** of plate numbers (values) of cars with that brand.
- Define a **function**, `find_by_city()` that takes a string city code and a car dictionary as parameters. The function should return:
 1. a **list** containing the brand of the cars with the given city code (**no duplicates**)
 2. the percentage of the cars with the given city code in the dictionary.
- Write a **script** that uses the functions defined above to read the cars into a dictionary. The script should then input a city code from the user and display the brands of the cars with the given city code, and the percentage of cars from the given city.

Note: the city code is the first two characters of the plate number.

Sample file: cars.txt (sample only, contents of the file may change!!)

Sample Run:

```
Enter city code to search: 06
Models with city code 06 ['Honda', 'Toyota', 'Fiat']
Percentage of cars from 06 : 50.0
```