

# CS 115 - Introduction to Programming in Python

## Lab 05

---

**Lab Objectives:** Tuples, Lists, Dictionaries

---

**Notes:**

- You should only use functionality covered in CS115 in your solution.
- Include a docstring for your functions.

**Q1:** In the file, `Lab5_Q1.py`, complete the following:

- a. Write a function `swap_pairs()` that takes a list as a parameter and switches the order of the pairs of items in the list. Your function should swap the order of the first two items, then swap the order of the next two, and so on.
- b. Write a script that initializes a tuple of two lists:

```
[2, 8, 5, 9, 4, 7, 6]
['ab', (2, 4), True, 5, [9, 4], (7, (9, (3, 6)))]
```

- c. and then call the `swap_pairs()` function to swap the pairs of items in each list and display it.

**Sample Run:**

```
original list: [2, 8, 5, 9, 4, 7, 6]
pairs swapped: [8, 2, 9, 5, 7, 4, 6]
```

```
original list: ['ab', (2, 4), True, 5, [9, 4], (7, (9, (3, 6)))]
pairs swapped: [(2, 4), 'ab', 5, True, (7, (9, (3, 6))), [9, 4]]
```

**Q2:** In a file, `Lab5_Q2.py`, complete the following:

- a. Write a function `remove_tuple_elements_containing_tuples()` which takes a tuple as a parameter and returns a new tuple containing the elements of the input tuple which are not tuples containing a tuple.

- b. Write a script that initializes a tuple with the below tuples:

```
(5, 'ab', (1, 4), 4.3, 'xyz', (2, 'a')),  
(5, 'ab', (1, 4, (3, 5)), 4.3, 'xyz', (2, 'a', (2, 7))),  
(5, 'ab', [1, 4, (3, 5)], 4.3, 'xyz', (2, 'a', [2, (7)])),  
(5, 'ab', (1, 'xyz', (2, 3, (4, 5, [6, 7])))),  
(5, 'ab', (1, 'xyz', [2, 3, (4, 5, (6, 7))]))
```

- c. and for each tuple element, display that tuple and find its elements which are not tuples containing a tuple using the `remove_tuple_elements_containing_tuples()` function. Display the returned tuple.

### Sample Run:

```
current tuple: (5, 'ab', (1, 4), 4.3, 'xyz', (2, 'a'))
```

```
tuple elements containing a tuple removed: (5, 'ab', (1, 4), 4.3, 'xyz',  
(2, 'a'))
```

```
current tuple: (5, 'ab', (1, 4, (3, 5)), 4.3, 'xyz', (2, 'a', (2, 7)))
```

```
tuple elements containing a tuple removed: (5, 'ab', 4.3, 'xyz')
```

```
current tuple: (5, 'ab', [1, 4, (3, 5)], 4.3, 'xyz', (2, 'a', [2, 7]))
```

```
tuple elements containing a tuple removed: (5, 'ab', [1, 4, (3, 5)], 4.3,  
'xyz', (2, 'a', [2, 7]))
```

```
current tuple: (5, 'ab', (1, 'xyz', (2, 3, (4, 5, [6, 7]))))
```

```
tuple elements containing a tuple removed: (5, 'ab')
```

```
current tuple: (5, 'ab', (1, 'xyz', [2, 3, (4, 5, (6, 7))]))
```

```
tuple elements containing a tuple removed: (5, 'ab', (1, 'xyz', [2, 3, (4,  
5, (6, 7))]))
```

**Q3:** Write a program `Lab5_Q3.py`, to track all courses and grades of a student (by her/his student number). Your program should store the list of students and course grades in a dictionary, where the key is the student number and the value is a list of tuples (course\_names and grades). Your program should define the following functions:

1. `add_student()`: takes a dictionary of students, a student id number and tuple containing the course name and float grade as parameters and creates a dictionary entry for the student and adds the first course(tuple) to the list of courses. Display error message if student already exists and success message if student is successfully added to the dictionary.
2. `add_course()`: takes a dictionary, student id number and course tuple (course\_name, grade), and adds the course to the list of courses that student takes. If the student is not in the dictionary, display an error message.
3. `find_student()`: takes a student number as a parameter and returns the list of courses and grades of the given student. Return None if the student is not in the list.

Your program should create a dictionary and implement the menu shown below.

**Sample Run: (User input is shown in red.)**

```
1)Add Student
2)Search Student
3)Add Course Grade
4)Show All Students
5)Quit
```

Enter Choice:**4**

All students: {}

```
1)Add Student
2)Search Student
3)Add Course Grade
4)Show All Students
5)Quit
```

Enter Choice:**1**

Enter student number: **1234**

Enter course: **econ101**

Enter grade: **98**

Student Added

```
1)Add Student
2)Search Student
3)Add Course Grade
```

4) Show All Students

5) Quit

Enter Choice: **3**

Enter student number: **1234**

Enter course: **cs115**

Enter grade: **95**

1) Add Student

2) Search Student

3) Add Course Grade

4) Show All Students

5) Quit

Enter Choice: **2**

Enter student number: **1234**

List of courses and grades: [('econ101', 98.0), ('cs115', 95.0)]

1) Add Student

2) Search Student

3) Add Course Grade

4) Show All Students

5) Quit

Enter Choice: **2**

Enter student number: **5678**

Student does not exist

1) Add Student

2) Search Student

3) Add Course Grade

4) Show All Students

5) Quit

Enter Choice: **1**

Enter student number: **9876**

Enter course: **cs115**

Enter grade: **80**

Student Added

1) Add Student

2) Search Student

3)Add Course Grade  
4)Show All Students  
5)Quit

Enter Choice:**4**

All students: {'1234': [('econ101', 98.0), ('cs115', 95.0)], '9876':  
[('cs115', 80.0)]}

1)Add Student  
2)Search Student  
3)Add Course Grade  
4)Show All Students  
5)Quit

Enter Choice:**3**

Enter student number: **9876**

Enter course: **chem101**

Enter grade: **92**

1)Add Student  
2)Search Student  
3)Add Course Grade  
4)Show All Students  
5)Quit

Enter Choice:3

Enter student number: **1234**

Enter course: **math101**

Enter grade: **86**

1)Add Student  
2)Search Student  
3)Add Course Grade  
4)Show All Students  
5)Quit

Enter Choice:**1**

Enter student number: **1234**

Enter course: **phys101**

Enter grade: **75**

Student already exists

- 1)Add Student
- 2)Search Student
- 3)Add Course Grade
- 4)Show All Students
- 5)Quit

Enter Choice:**4**

All students: {'1234': [('econ101', 98.0), ('cs115', 95.0), ('math101', 86.0)], '9876': [('cs115', 80.0), ('chem101', 92.0)]}

- 1)Add Student
- 2)Search Student
- 3)Add Course Grade
- 4)Show All Students
- 5)Quit

Enter Choice:**5**

Program Ended....