Week 04 Tuple, List, Iterables

Part 1 Tuple

In this exercise, please try to come out with the answer without using IDLE/Python first. Then type in the expressions into IDLE to verify your answers. The objective is for you to understand why and how they work. If there is an error, specify the error and the cause of error.

Expressions	**Output**

`tup_a = (10, 11, 12, 13)`
`print(tup_a)`
`tup_b = (“CS”, 1010)`
`print(tup_b)`
`tup_c = tup_a + tup_b`
`print(tup_c)`
`print(len(tup_c))`	

`print(11 in tup_a)`
`print(14 in tup_b)`
`print(“C” in tup_c)`
`print(tup_b[1])`
`tup_d = tup_b[0]*4`
`print(tup_d)`
`print(tup_b[1] * 4)`	

`tup_e = tup_d[1:]`
`print(tup_e)`
`tup_f = tup_d[::-1]`
`print(tup_f)`
`tup_g = tup_d[1:-1:2]`
`print(tup_g)`
`tup_h = tup_d[-1:6:-2]`
`print(tup_h)`	

`tup_i = (1)`
`print(tup_i)`
`tup_j = (1,)`
`print(tup_j)`
`print(tup_i * 4)`
`print(tup_j * 4)`	

`print(min(tup_a))`
`print(max(tup_a))`
`print(min(tup_c))`
`print(max(tup_c))`
`print(min(tup_e))`
`print(max(tup_e))`	

`for i in tup_b:`
`  print(i)`	

`for i in range(5):`
`  print(i)`	

`for i in range(2,5):`
`  print(i)`	

`for i in range(2,5,2):`
`  print(i)`	

`for i in range(5,1,-1):`
`  print(i)`	

`for i in range(5,6,-1):`
`  print(i)`	

Part 2 List

In this exercise, please try to come out with the answer without using IDLE/Python first. Then type in the expressions into IDLE to verify your answers. The objective is for you to understand why and how they work. If there is an error, specify the error and the cause of error.

Expressions	**Output**

`lst_a = [“CS”, 1010]`
`print(lst_a)`
`lst_b = [“E”,(“is”, “easy”)]`
`print(lst_b)`
`lst_c = lst_a + lst_b`
`print(lst_c)`	

`tup_a = (“CS”, 1010)`
`tup_a[1] = 2030`
`lst_a[1] = 2030`
`print(lst_a)`	

`lst_a.append(“E”)`
`print(lst_a)`
`lst_a.extend(“easy”)`
`print(lst_a)`	

`cpy_b = lst_b[:]`
`print(cpy_b)`
`cpy_b[1] = “is hard”`
`print(cpy_b)`
`print(lst_b)`	

`lst_d = [1, [2], 3]`
`cpy_d = lst_d[:]`
`print(cpy_d)`
`print(lst_d)`
`lst_d[1][0] = 9`
`print(cpy_d)`
`print(lst_d)`	

`print(lst_d == cpy_d)`
`print(lst_d is cpy_d)`
`print(lst_d[1] == cpy_d[1])`
`print(lst_d[1] is cpy_d[1])`	

Part 3 List Mutation

Try the follow and notice the difference between the left and right column in each row.

`x = 1`
`y = x`
`x = 2`
`print(x)`
`print(y)`
	`lstx = [1,2,3]`
	`lsty = lstx`
	`lstx[0] = 999`
	`print(lstx)`
	`print(lsty)`

`a = 4`
 	
`def foo(x):`
`    x = x * 2`
`    print(x)`
 	
`print(a)`
`foo(a)`
`print(a)`
	`lsta = [1,2,3]`
	
	`def foo2(lst):`
	`    lst[0] = lst[0]*2`
	`    lst[1] = lst[1]*2`
	`    print(lst)`
	
	`print(lsta)`
	`foo2(lsta)`
	`print(lsta)`

Part 4 Data

We can use tuple/list to create a data that consists of mixed types. In this exercise, you are hired by NUS to improve the LumiNUS system due to high rate of complain. You are provided with an implementation of the records for each class as follows:

```
def make_module(code, units):
  return (code, units)

def make_units(lec, tut, lab, hw, prep):
  return (lec, tut, lab, hw, prep)

def get_module_code(mod):
  return mod[0]

def get_module_units(mod):
  return mod[1][0] + mod[1][1] + mod[1][2] + mod[1][3] + mod[1][4]
```

Due to Personal Data Protection Act (PDPA), you are to respect abstraction barriers. For instance, to get the course code from a module mod, you cannot simply use `mod[0]`. You must call the function `get_module_code(mod)`, otherwise, you will get penalized by the PDPC.

A. Write a constructor `make_empty_schedule()` that returns an empty schedule.
B. Write a function `add_class(mod, schedule)` that returns a new schedule with the added module mod.
C. Write a function `total_scheduled_units(schedule)` that computes and returns the total number of units from all modules in the given schedule.
D. Write a function `drop_class(mod, schedule)` that returns a new schedule with a particular module mod dropped from the given schedule.
E. Write a function `credit_limit(schedule, max_credit)` that takes in a schedule and the maximum credit and returns a new schedule that has total number of units less than or equal to `max_credit` by removing modules from the specified schedule.