

SCIT, University of Wollongong

CSIT110/CSIT810

2021 Session 3

Assignment 3 (10%) due on Saturday 14th Aug 2021 at 00:00AM

Objectives

- Able to write clear code with comments and follow coding convention
- Able to use variables with meaningful names and correct data types
- Able to define functions and class objects

Marking criteria:

- Total mark is 10. Deduct 1 mark for each day late.
- More than 3 days late will result in a zero mark.
- Code must be able to run with no errors: 0 mark for the whole assignment if there is an error is thrown.
- Correct file format (.py extension): 0 mark for the whole assignment if file submission is not in correct format.
- Use submission template for file submission.
- Save your file with this naming format
<name>_<uow_id>_<favourite_cartoon_character>_a3.py

Question 1	correctness, completeness and consistency with the assignment specification	2 marks
Question 2	correctness, completeness and consistency with the assignment specification	2 marks
Question 3	correctness, completeness and consistency with the assignment specification	3 marks
Question 4	correctness, completeness and consistency with the assignment specification	3 marks
Overall	comments include name, student number, subject code; clear code and follow coding convention; use variables with meaningful names and correct data types	Deduct up to 1 mark

Submission Instruction: Assignment 1 submission is on Moodle. Put all your python code into a single python file (file extension .py) and submit it.

Assignment questions: there are 4 assignment questions.

Write clear code with **comments** and follow **coding conventions**. Comments should include **your name, student number** and **subject code** on top of your code. Please also add this information to the variables as stated in the template Your code must work **exactly** like the provided examples given the input in the examples.

```
name = "John Snow"
student_num = "1234567" # Student number
subject_code = "CSIT110" # CSIT110 or SP121

...
```

Question 1. Define a function that satisfies the following specifications.

Function name	process_subscription
Parameters	-
Return Value	-
Detailed description	<p>With the use of a dictionary-typed variable, the function must display the six plans available for subscription and take subscription orders. Your program should work exactly as the following examples. The text in bold indicates user input.</p> <p>Lines 2 to 7 has two columns. The two columns are 39 characters and 7 characters in width.</p> <p>There is a space after each colon in the prompts.</p> <p>The bullets points before the list of selections is a space, a hyphen and a space.</p>

. Example 1: The user subscribes to 2 plans

<pre>Packages available for subscription(price/mth) Standard Food Delivery Plan \$13.40 Premium Food Delivery Plan \$25.40 Basic Commute Plan \$39.80 Premium Commute Plan \$69.80 All Access Plan(Lite) \$79.60 All Access Plan(Premium) \$108.60 Subscribe to Standard Food Delivery Plan? (Y/N): Y Subscribe to Premium Food Delivery Plan? (Y/N): N Subscribe to Basic Commute Plan? (Y/N): N Subscribe to Premium Commute Plan? (Y/N): N Subscribe to All Access Plan(Lite)? (Y/N): Y Subscribe to All Access Plan(Premium)? (Y/N): N Your selection: - Food Delivery Plan(\$21.40) - All Access Plan(Lite) (\$9.56) Total cost \$30.96</pre>

Example 2: The user subscribes to 0 plan.

<pre>Packages available for subscription(price/mth) Standard Food Delivery Plan \$13.40 Premium Food Delivery Plan \$25.40 Basic Commute Plan \$39.80 Premium Commute Plan \$69.80</pre>

All Access Plan(Lite)	\$79.60
All Access Plan(Premium)	\$108.60
<p>Subscribe to Standard Food Delivery Plan? (Y/N): Y</p> <p>Subscribe to Premium Food Delivery Plan? (Y/N): N</p> <p>Subscribe to Basic Commute Plan? (Y/N): N</p> <p>Subscribe to Premium Commute Plan? (Y/N): N</p> <p>Subscribe to All Access Plan(Lite)? (Y/N): N</p> <p>Subscribe to All Access Plan(Premium)? (Y/N): N</p>	
<p>Your selection:</p> <ul style="list-style-type: none"> - None 	
Total cost \$0.00	

Important requirement:

Your program must use string addition to produce the exact output as illustrated in the above example. You may assume the input are of letters n and y.

Question 2. Define a function that satisfies the following specifications.

Function	<code>generate_qns_from_list</code>
Parameters	1. A list. Each element of this list is a list of integers.
Return Value	1. A list of dictionary objects
Detail Information	<p>This function should convert the list parameter into a list of dictionaries. Each dictionary will have two string-typed keys – "qns" and "ans".</p> <p>The value for the "qns" key will be a str of the integers taken from each list element in the list parameter. In the str, the integers are separated with the characters " + ".</p> <p>The value for the "ans" key will be the sum of all the numbers in the list.</p> <p>You should skip the lists that contain less than 2 integers. There is no need to print anything out.</p>

Example:

<pre>input_list = [[1,3,3], [2,5,-1],[3,2],[4,5,3],[0,23],[1,2,3,4]] generate_qns_from_list(input_list)</pre>
<p>The function should return the following list</p> <pre>[{"qns": "1 + 3 + 3", "ans": 7}, {"qns": "2 + 5 + -1", "ans": 6}, {"qns": "3 + 2", "ans": 5}, {"qns": "4 + 5 + 3", "ans": 12}, {"qns": "0 + 23", "ans": 23}, {"qns": "1 + 2 + 3 + 4", "ans": 10}]</pre>

Question 3a.

Define a class that satisfy the following specifications.

Class name	ShoppingCart
Class attribute	server_url: str
Instance attributes	account_id: str cart: dict
Parameter	account_id: str
Detailed information	<p>Assign the str "128.123.123.0" to the class attribute, server_url.</p> <p>The value of the instance attribute, account_id, should be assigned with the parameter of the same name.</p> <p>The instance attribute, cart, is an empty dictionary when a ShoppingCart object is instantiated.</p> <p>The keys in the dictionary are the product names, of str types, while the values, of int type, mark the quantity of the products in the cart.</p>

Question 3b

Write an instance method for the ShoppingCart class that satisfies the following specifications.

Method name	add_item_to_cart
Parameter	1. product_name: str 2. quantity: int
Return value	-
Detailed description	<p>The function should check if the item exists in the keys of the cart attribute and update the number of the items in the cart accordingly.</p> <p>That is, if the item does not exist in the cart, it should create a key-value pair for the car. Otherwise, it should add the quantity in the parameter to the existing value.</p> <p>You may assume that the quantity in the parameter will be more than 0.</p>

Question 3c

Write an instance method for the ShoppingCart class that satisfies the following specifications.

Method name	<code>remove_item_from_cart</code>
Parameter	<code>1. product_name: str</code> <code>2. quantity: int</code>
Return value	-
Detailed description	The function should update the cart attribute with the correct number of product. If the updated quantity is 0, the key value pair should be removed from the cart. You can assume that the product is available for removal and the quantity to be removed is less than or equal to the quantity in the cart.

Question 3d

Write a class method for the ShoppingCart class that satisfies the following specifications.

Method name	<code>get_url</code>
Parameter	-
Return value	<code>1. A str</code>
Detailed description	This method returns the class attribute <code>server_url</code>

Question 3e

Write an instance method for the ShoppingCart class that satisfies the following specifications.

Method name	<code>empty_cart</code>
Parameter	-
Return value	-
Detailed description	This method empties the instance attribute, <code>cart</code>

Question 3f

Write an instance method for the ShoppingCart class that satisfies the following specifications.

Method name	count_items
Parameter	-
Return value	1. int
Detailed description	This method returns the total number of items based on the existing items in the cart. If there is nothing in the cart, the return value should be 0.

An example of how your code should work given the following parameters.

```
newCart = ShoppingCart("1234567")
newCart.add_item_to_cart("fruit juice", 2)
newCart.add_item_to_cart("tissue box", 4)
newCart.add_item_to_cart("ice cream", 3)
# newCart.cart is now {"fruit juice": 2, "tissue box": 4, "ice cream": 3}
newCart.remove_item_from_cart("tissue box",1)
newCart.remove_item_from_cart("fruit juice",2)
# newCart.cart is now {"tissue box": 3, "ice cream": 3}
newCart.count_items() # returns 6
print(newCart.get_url()) # prints 128.123.123.0
newCart.empty_cart()
newCart.count_items() # returns 0
```

Question 4

Part 1

The checksum letter of a magic 7 digits number is calculated as such:

$$D = [(i_1 i_2 i_3 i_4 i_5 i_6 i_7) \cdot (2\ 7\ 6\ 5\ 4\ 3\ 2)] \bmod 11$$
$$= (2i_1 + 7i_2 + 6i_3 + 5i_4 + 4i_5 + 3i_6 + 2i_7) \bmod 11$$

Where i_x is the 1st to last of the 7 digits of the numbers and (2,7,6,5,4,3,2) are the weights.

The checksum is a letter which corresponds to the number d as shown in the look-up table below

d	10	9	8	7	6	5	4	3	2	1	0
Check digit	A	B	C	D	E	F	G	H	I	Z	J

Question 4a

Define a function that meets the following specifications.

Method name	get_magic_num_checksum																								
Parameter	<div>1. str</div> <div>This str contains the numerical series mentioned in the description above.</div>																								
Return value	<div>1. str</div> <div>An upper case letter</div>																								
Detailed description	<div>From the given string of 7 numbers, computer the number d as described in the section 'Part 2'. Return the letter which corresponds to the number d as shown in the look-up table below</div> <table><tr><td>d</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr><tr><td>Check digit</td><td>A</td><td>B</td><td>C</td><td>D</td><td>E</td><td>F</td><td>G</td><td>H</td><td>I</td><td>Z</td><td>J</td></tr></table>	d	10	9	8	7	6	5	4	3	2	1	0	Check digit	A	B	C	D	E	F	G	H	I	Z	J
d	10	9	8	7	6	5	4	3	2	1	0														
Check digit	A	B	C	D	E	F	G	H	I	Z	J														

Part 2

A typical vehicle registration number comes in the format **xxx #### y**:

- **x** – prefixes
- **####** – Numerical series (from 1 to 9999, without leading zeroes)
- **y** – Checksum
 - The checksum letter is calculated by converting the letters into numbers, *i.e.*, where A=1 and Z=26, potentially giving seven individual numbers from each registration plate. However, only two letters of the prefix are used in the checksum. For a three-letter prefix, only the last two letters are used; for a two-letter prefix, both letters are used; for a single letter prefix, the single letter corresponds to the second position, with the first position as 0. For numerals less than four digits, additional zeroes are added in front as placeholders, for example "1" is "0001". SBS 3229 would therefore give 2, 19, 3, 2, 2 and 9 (note that "S" is discarded); E 12 would give 0, 5, 0, 0, 1 and 2. SS 108 would be given as 19, 19, 0, 1, 0, 8.
 - Each individual number is then multiplied by 6 fixed numbers (9, 4, 5, 4, 3, 2). These are added up, then divided by 19. The remainder corresponds to one of the 19 letters used (A, Z, Y, X, U, T, S, R, P, M, L, K, J, H, G, E, D, C, B), with "A" corresponding to a remainder of 0, "Z" corresponding to 1, "Y" corresponding to 2 and so on. In the case of SBS 3229, the final letter should be a P; for E 23, the final letter should be a H. SS 11 back letter should be a T. The letters F, I, N, O, Q, V and W are not used as checksum letters.

Question 4b

Define a function that meets the following specifications.

Function name	<code>get_car_plate_checksum</code>
Parameter	<ol style="list-style-type: none"><code>str</code> <p>This <code>str</code> contains the prefixes and numerical series mentioned in the description above.</p>
Return value	<ol style="list-style-type: none"><code>str</code> <p>An upper case letter</p>
Detailed description	<p>Compute and return the checksum from the string parameter. The computation logic is described in the section titled 'Part 1'.</p> <p>The checksum is one character in length. The return value is case insensitive.</p> <p>You should use the try and except blocks to find out if a character in a string is an integer or not. The input string may contain 1-3 letters for prefixes while there can be 1 to 4 digits for the numerical series that follows.</p>