

SCIT, University of Wollongong

CSIT110/CSIT810

Session 2 2021

Examination (60%) due on Monday 7th June 2021 at 01:00 PM

Marking criteria:

- Total marks is 60.
- No errors should be raised unless otherwise specified.
- Correct file format (.py extension): 5 marks penalty if file submission is not in the correct format.
- Names of variables, functions and classes that do not meet specifications will result in 0 marks for each question. Spelling errors in attributes, variables, functions, class names which trigger errors in testing the question will be penalised.
- Use submission template for file submission.

Question 1	Correctness, completeness and consistency with the exam specifications.	15 marks
Question 2	Correctness, completeness and consistency with the exam specifications.	15 marks
Question 3	Correctness, completeness and consistency with the exam specifications.	5 marks (bonus)
Question 4	Correctness, completeness and consistency with the exam specifications.	10 marks
Question 5	Correctness, completeness and consistency with the exam specifications.	10 marks
Question 6	Correctness, completeness and consistency with the exam specifications.	10 marks
Overall	Script includes name, student number, subject code; clear code and follow coding convention; use variables with meaningful names and correct data types. No calling of functions outside of the main() block. Use the template given.	Deduct up to 1 mark for each issue Deduct up to 5 marks

Submission Instruction: Submission is on Moodle. Put all your python code into a single python file (<name>_<uow std no>_exam.py) and submit it.

Exam questions: there are 6 examination questions.

Write clear code with **comments** and follow **coding conventions**. The python file should include **your name, student number** and **subject code** on top of your code as provided as stated in the template.

You may assume that the parameters used for testing are of the right data type specified in each question, unless stated otherwise.

```
"""Examination"""
```

```
name = "John Snow"
student_uow_id = "1234567" # UOW student number
course_code = "CSIT110" # CSIT110 or SP420
```

Question 1: There are 5 parts to this question. Write the following **functions**.

Q1a.

Function name	question_1a
Parameter(s)/ argument(s)	1. A list object This list object contains string and integer objects.
Return value	A list object This list object contains first a string object, followed by an integer object
Detailed description	i. Return a list with a string object and an integer object. ii. The string object is a concatenation of the string objects in the list argument and a space between each of them . The string objects should be concatenated in the order from the smallest indexed to the largest index in the list iii. The integer object is the sum of all the integers in the list. Use isinstance(), type() or try and catch block to check if each object in the list is a string or integer object.
Example usage	<pre>x = question_1a([1, "ramen", "ayam penyet", 234, "Bulgogi", 111, 5])</pre>
x value	<pre>["ramen ayam penyet Bulgogi", 351]</pre> Single or double quotation does not matter as long as it is a string type object.

Q1b.

Function name	question_1b
Parameter(s)/ argument(s)	In this order, 1. an integer, x 2. an integer, N
Return value	A list of integers
Detailed description	return a list of 2*N number of x
Example usage	y = question_1b(15,3)
y value	[15,15,15,15,15,15]

Q1c.

Function name	question_1c
Parameter(s)/ argument(s)	In this order, 1. a list of string objects 2. a string object
Return value	An integer
Detailed description	return the number of times the given string appears in the given list
Example usage	z = question_1c(["kopi-o", "teh-o", "teh-si", "kopi-o"], "kopi-o")
z value	2

Q1d.

Function name	question_1d
Parameter(s)/ argument(s)	In this order, 1. a list of str objects 2. a str
Return value	1 integer
Detailed description	Count the total number of times the second string argument appears in the str objects in the list argument. Return this number. You may assume that each string element in the list will not contain more than one occurrence of the string argument.
Example usage	<code>xx = question_1d(["chicken rice", "nasi padang", "omu rice", "ice kacang"], "ice")</code>
xx value	3

Q1e.

Function name	sum_of_power
Parameter(s)/ argument(s)	In this order, 1. a list of N integers, list1 2. a list of N integers, list2
Return value	An integer
Detailed description	<p>Given:</p> <pre>list1 = [a1, a2, a3, b4, ...] list2 = [b1, b2, b3, b4, ...]</pre> <p>The sum of power is defined as such</p> $a1^{b1} + a2^{b2} + a3^{b3} \dots$ <p>Return the sum of power of the two lists. You can assume</p> <ul style="list-style-type: none">- the list parameters are of the same length and $N > 0$- the integers in the second list are ≥ 1.
Example usage	<code>yy = sum_of_power([1,2,3,4,5],[9,8,7,6,5])</code>
yy value	9665

Q2a.

Write a function that meets the following specifications.

Function name	collate_order
Parameter(s)/ argument(s)	1. A list of str.
Return value	A dictionary
Detailed description	Using the string objects in the list argument as keys and the number of each unique string occurrence as values, add the key-value pairs to a dictionary and return the dictionary.
Example usage	<pre>zz = collate_order(["chicken rice", "nasi padang", "omu rice", "ice kacang", "nasi padang", "kopi-o", "teh-o", "teh- si", "kopi-o"])</pre>
zz value	<pre>{"chicken rice":1, "nasi padang":2, "omu rice":1, "ice kacang":1, "kopi-o":2, "teh-o":1, "teh-si":1}</pre>

Q2b.

Write an Exception class that meets the following specifications.

Class name	FoodTooSpicyError This class inherits the Exception class
Parameter(s)/ argument(s)	1. a str, x Store the str argument as an instance attribute with the name of your choice.
Return value	No return values.
Instance Method	<pre>__str__</pre> Define a <code>__str__</code> dunder instance method that returns the uppercase of the str instance attribute initialised in the constructor. This instance method does not take in any parameter.

Q2c

Write a function that meets the following specifications.

Function name	question_2c										
Parameter(s)/ argument(s)	-										
Return value	No return values.										
Error handling	Function should raise an exception. The type of exception raised is described below.										
Detailed description	<p>Using the following text as the prompt, get an integer from the user input. "Please enter the Scoville Heat Units of your dish: " There is a space after the colon symbol.</p> <p>You may assume that the user input will be a valid integer.</p> <p>Raise the FoodTooSpicyError with the str state below.</p> <table border="1"> <thead> <tr> <th>Condition</th><th>Str obj used to raise the exception.</th></tr> </thead> <tbody> <tr> <td>$x < 200$</td><td>"This is a false alarm."</td></tr> <tr> <td>$200 \leq x < 700$</td><td>"No kick at all but we have customers who cannot take much spiciness."</td></tr> <tr> <td>$700 \leq x < 25000$</td><td>"Slight chance of a tummy ache."</td></tr> <tr> <td>$25000 \leq x$</td><td>"Some kind of habanero detected!"</td></tr> </tbody> </table> <p>Do not forget the punctuation at the end of the statements. The marker is aware that the __str__ dunder method in the Exception class will convert the str to uppercase.</p>	Condition	Str obj used to raise the exception.	$x < 200$	"This is a false alarm."	$200 \leq x < 700$	"No kick at all but we have customers who cannot take much spiciness."	$700 \leq x < 25000$	"Slight chance of a tummy ache."	$25000 \leq x$	"Some kind of habanero detected!"
Condition	Str obj used to raise the exception.										
$x < 200$	"This is a false alarm."										
$200 \leq x < 700$	"No kick at all but we have customers who cannot take much spiciness."										
$700 \leq x < 25000$	"Slight chance of a tummy ache."										
$25000 \leq x$	"Some kind of habanero detected!"										

Q2d

Write a function that meets the following specifications.

Function name	get_bill
Parameter(s)/ argument(s)	<p>1. A dict</p> <p>The keys of the dictionary are str objects and they are the names of food items. The values are float objects and they are the prices of the food items in.</p>
Return value	1. A float.
Detailed description	Return the sum of the prices in the dictionary. There is no need to format the float.
Example Usage	<pre>bill = get_bill({ "chicken rice":2.5,"nasi padang":3.2,"omu rice":6.5,"ice kacang":1.8, "kopi-o":0.9, "teh-o":0.9, "teh-si":1.1})</pre>
bill value	16.900000000000002

Q2e

Write a function that meets the following specifications.

Function name	has_long_name
Parameter(s)/ argument(s)	a str
Return value	a boolean or a None type object
Detailed description	<p>Return True if the length of the string argument is longer than 20.</p> <p>Return False if the length of the string argument is less than or equals to 20 but not equal to 0.</p> <p>Return the None object if an empty str is received.</p>

Q3

Write a function that meets the following specifications.

Function name	question_3											
Parameter(s)/ argument(s)	a function reference, fcn_arg											
Return value	an integer											
Detailed description	<p>In a try and except block, call the function given in the parameter.</p> <p>The function, fcn_arg, does not take in any arguments.</p> <p>Return an integer corresponding to the following conditions.</p> <table><tr><th>Condition</th><th>Return value</th></tr><tr><td>ZeroDivisionError was raised. An error raised when a method or variable of an instance which was referenced did not exist</td><td>An integer object -1</td></tr><tr><td>KeyError was raised. This occurs when an argument that has the right type but an inappropriate value</td><td>An integer object 1</td></tr><tr><td>All other errors</td><td>An integer object 999</td></tr><tr><td>If no error was raised</td><td>Return the integer which the function, fcn_arg returns</td></tr></table>		Condition	Return value	ZeroDivisionError was raised. An error raised when a method or variable of an instance which was referenced did not exist	An integer object -1	KeyError was raised. This occurs when an argument that has the right type but an inappropriate value	An integer object 1	All other errors	An integer object 999	If no error was raised	Return the integer which the function, fcn_arg returns
Condition	Return value											
ZeroDivisionError was raised. An error raised when a method or variable of an instance which was referenced did not exist	An integer object -1											
KeyError was raised. This occurs when an argument that has the right type but an inappropriate value	An integer object 1											
All other errors	An integer object 999											
If no error was raised	Return the integer which the function, fcn_arg returns											

Q4 An angle can be measured in radians or degrees.

Q4a

Write a class that fulfils the following specifications.

Class name	Angle
Parameter(s)/ argument(s) for the constructor __init__	1. A float, angleInDegrees
Detailed description	Using the argument, initialise an instance attribute of the same name.
Example usage	<pre>right_angle = Angle(90.01) acute_angle = Angle(23.1) print(right_angle.angleInDegrees()) print(acute_angle.angleInDegrees())</pre>
Example usage output	<pre>90.01 23.1</pre>

Q4b.

Define an **instance method** for the Angle class as follows.

Method name	get_degrees
Method type	Instance method
Parameter(s)/ argument(s)	- No parameter for this method
Return value	A float
Detailed description	Return the stored instance attribute angleInDegrees
Example usage	<pre>right_angle = Angle(90.01) acute_angle = Angle(23.1) print(right_angle.get_degrees()) print(acute_angle.get_degrees())</pre>
Example usage output	<pre>290.01 23.1</pre>

Q4c.

Define an **instance method** for the Angle class as follows.

Method name	get_radians
Method type	Instance method
Parameter(s)/ argument(s)	- No parameter for this method
Return value	A float
Detailed description	<p>Return the stored angle value converted to radians.</p> <p>Note:</p> $\text{angleInRadians} = \text{math.pi} * \text{angleInDegrees} / 180$ <p>You will have to import the math library to compute the angle in radians. You do not have to format, e.g. rounding up or down, the price.</p>
Example usage	<pre>right_angle = Angle(90.01) acute_angle = Angle(23.1) print(right_angle.get_radians()) print(acute_angle.get_radians())</pre>
Example usage output	<pre>1.570970859720096 0.40317105721069013</pre>

Q4d.

Define a **static method** for the Angle class as follows.

Method name	deg2rad
Method type	static method
Parameter(s)/ argument(s)	A float An angle measured in degrees.
Return value	A float An angle in radians.
Detailed description	Convert the parameter from degrees to radians. $\text{angleInRadians} = \text{math.pi} * \text{angleInDegrees} / 180$ You will have to import the math library to compute the angle in radians. You do not have to format, e.g. rounding up or down, the price.
Example usage	<pre>print(Angle.deg2rad(90)) print(Angle.deg2rad(45))</pre>
Example usage output	<pre>1.5707963267948966 0.7853981633974483</pre>

Q4e.

Define a **class method** for the Angle class as follows.

Method name	from_radians
Method type	class method
Parameter(s)/ argument(s)	A float, angleInRadians
Return value	An instance of the Angle class
Detailed description	<p>Return an instance of the Angle class</p> <p>You will have to convert angle in radians to angle in degrees before instantiating the Angle object.</p> <p>Note: $\text{angleInDegrees} = 180 * \text{angleInRadians} / \text{math.pi}$</p>
Example usage	<pre>right_angle = Angle.from_radians(1.571) acute_angle = Angle.from_radians(1.571) print(right_angle.get_degrees()) print(acute_angle.get_degrees())</pre>
Example usage output	<pre>90.01166961505233 44.97718691776963</pre>

Q5a.

Write a **class** that fulfils the following specifications.

Class name	Dish
Parameter(s)/ argument(s) for the constructor __init__	In this order, 1. name - a str 2. pax - an int 3. spiceLevel - an int 4. cost - a float
Detailed description	<ol style="list-style-type: none">1. Assign the first three parameters, name, pax and spiceLevel to instance attributes of the same names, i.e. name, pax and spiceLevel2. Using the argument, cost, instantiate an instant attribute, price. The price of the Dish object is 150% <u>more than</u> the cost in the dictionary. That is 2.5 times of the cost. You do not have to format, e.g. rounding up or down, the price.
Example usage	<pre>main_course1 = Dish("Fish and Chips",8.6,1,200) print(main_course1.name) print(main_course1.price) print(main_course1.pax) print(main_course1.spiceLevel)</pre>
Example usage output	<pre>Fish and Chips 21.5 1 200</pre>

Q5b.

Write a **class method** for the Dish class that fulfils the following specifications

Method name	from_dict
Method type	Class method
Parameter(s)/ argument(s)	A dictionary with the following key value pairs 1. Key - "name", value - a str 2. Key - "cost", value - a float 3. Key - "pax", value - an integer 4. Key - "spiceLevel", value - an integer
Return value	An instance of the Dish class
Detailed description	Return an instance of the Dish class using the values of the keys "name", "cost", "pax" and "spiceLevel" in the dictionary.
Example usage	<pre>new_dish = Dish.from_dict({"name": "Ayam penyet", "cost":4.0, "pax":2, "spiceLevel":900}) print(new_dish.name) print(new_dish.price) print(new_dish.pax) print(new_dish.spiceLevel)</pre>
Example usage's instance attributes values	<pre>new_dish.name # "Ayam penyet" new_dish.price # 10.0 new_dish.pax # 2 new_dish.spiceLevel # 900</pre>

Q5c:

Write a **static** method for the Dish class that fulfils the following specifications

Method name	get_cost_from_price
Method type	Static method
Parameter(s)/ argument(s)	A float, price
Return value	A float
Detailed description	Calculate the cost with the float argument, price. Recall that the cost is 0.4 times the price. You do not need to format the float.
Example usage	cost_price = Dish.get_cost_from_price(10)
value of cost_price	4 in float type

Q5d:

Write an **instance method** for the Dish class that fulfils the following specifications

Method name	<code>__str__</code>
Method type	instance method
Parameter(s)/ argument(s)	- No parameter for this method
Return value	A str object
Detailed description	<p>Return the following str object</p> <p><code>"name_of_dish (\$price) is for # pax. [xxxSHU]"</code></p> <p>With the bold words replace as such</p> <p>Name_of_dish - instance attribute, name, of the dish.</p> <p>price - instance attribute, price, of the dish in two decimal places.</p> <p># - instance attribute, pax, of the dish.</p> <p>xxx - instance attribute, spiceLevel, of the dish</p>
Example usage	<pre>main_course1 = Dish("Fish and Chips",8.6,1,200) print(main_course1)</pre>
Console display	<pre>Fish and Chips (\$21.50) is for 1 pax. [200SHU]</pre>

Q6a.

Write the following **class**:

Class name	SideDish This class inherits the Dish class from question 5
Parameter(s)/ argument(s)	1. name - a str 2. cost - a float 3. pax - an int 4. spiceLevel - an int
Detailed description	Use the parameters to instantiate the instance by calling the parent class constructor.

Q6b.

Class name	PremiumDish This class inherits the Dish class
Parameter(s)/ argument(s)	1. name - a str 2. cost - a float 3. pax - an int 4. spiceLevel - an int 5. side_dish - a sideDish instance First, call the parent constructor with the relevant values. The cost of the premium dish is described below. Next, assign the side_dish argument to an instance attribute with the same name, ie side_dish <u>The cost of the premium dish</u> is the sum of the argument, cost, and the cost of the side_dish. Hint: Using the side_dish's price as the argument, you can get the cost of the side dish by calling the static method defined in Q5c.

-----End of Paper-----