

Inspiring Excellence

Course Code:	CSE111	
Course Title:	Programming Language II	
Classwork No:	04	
Topic:	OOP (Instance method and overloading)	
Number of Tasks:	7	

Design the **Customer** class with the necessary properties so that the following output is produced.

[Hint:

- If the visitor's age is greater than 10, then the ticket price is 100 taka. Otherwise, 50 taka.
- A customer can't buy more than 3 tickets.]

Driver Code	Given Output		
<pre>print('1') customer1 = Customer() print('2') customer1.buyTicket('Bob', 23) customer1.buyTicket('Henry', 7) customer1.buyTicket('Alexa', 30) customer1.buyTicket('Jonas', 43) print('3') customer1.showDetails() print('4') customer2 = Customer() print('5') customer2.buyTicket('Harry', 60) customer2.buyTicket('Harry', 60) customer2.buyTicket('Tomas', 28) print('6') customer2.showDetails()</pre>	Welcome to ABC Memorial Park 2		

You are given a string containing a sentence. Your task is to implement the **SentenceAnalyzer** class with constructor overloading and method overloading to analyze the given sentence.

Driver Code	Given Output		
<pre>analyzer1 = SentenceAnalyzer()</pre>	1		
	Number of words in the sentence: 4		
analyzer1.set_sentence("That's an	2		
easy one")	3		
print("1")	Number of words in the sentence: 5		
<pre>analyzer1.get_word_count()</pre>	4		
print("2")	Number of words with 4 characters		
<pre>analyzer2 = SentenceAnalyzer("Like I</pre>	in the sentence: 4		
<pre>said it's easy")</pre>	5		
	Number of words with 5 characters		
print("3")	in the sentence: 0		
analyzer2.get_word_count()			
print("4")			
analyzer2.get_word_count(4)			
print("5")			
<pre>analyzer1.get_word_count(5)</pre>			

Design the **Student** class with the necessary properties so that the given output is produced for the provided driver code. Use constructor overloading and method overloading where necessary.

Hint:

- A student having cgpa>=3.5 and credit>10 is eligible for scholarship.
 - A student having cgpa >=3.7 is eligible for Merit based scholarship
 - A student with cgpa>=3.5 but <3.7 is eligible for Need-based scholarship.

Driver Code	Given Output	
print('')		
std1 = Student("Alif", 3.99, 12)	Alification of the state of the	
print('')	Alif is eligible for Merit-based scholarship.	
std1.checkScholarshipEligibility()	Name: Alif	
print('')	Department: CSE	
·	CGPA: 3.99 Number of Credits: 12	
std1.showDetails()	Scholarship Status: Merit-based scholarship	
print('')		
std2 = Student("Mim", 3.4)		
std3 = Student("Henry", 3.5, 15,"BBA")	Mim is not eligible for scholarship.	
print('')	Henry is eligible for Need-based scholarship.	
std2.checkScholarshipEligibility()		
print('')	Name: Mim	
std3.checkScholarshipEligibility()	Department: CSE CGPA: 3.4	
print('')	Number of Credits: 9	
std2.showDetails()	Scholarship Status: No scholarship	
print('')	N	
,	Name: Henry Department: BBA	
std3.showDetails()	CGPA: 3.5	
print('')	Number of Credits: 15	
std4 = Student("Bob", 4.0, 6, "CSE")	Scholarship Status: Need-based scholarship	
print('')		
std4.checkScholarshipEligibility()	Bob is not eligible for scholarship.	
print('')		
std4.showDetails()	Name: Bob	
· · · · · · · · · · · · · · · · · · ·	Department: CSE CGPA: 4.0	
	Number of Credits: 6	
	Scholarship Status: No scholarship	

Write the **Author** class with the required properties so that the given output is produced for the provided driver code:

Driver Code	Output		
# Write your code here	=======================================		
	A book can not be added without		
# Do not change the following lines of	author name		
code.			
a1 = Author()	Number of Book(s): 1		
print("======"")	Author Name: Anna Kavan		
al.addBook("Ice", "Science Fiction")	Science Fiction: Ice		
print("======"")			
a1.setName("Anna Kavan")			
al.addBook("Ice", "Science Fiction")	Number of Book(s): 2		
a1.printDetail()	Author Name: Humayun Ahmed		
print("======"")	Science Fiction: Onnobhubon		
a2 = Author("Humayun Ahmed")	Horror: Megher Upor Bari		
a2.addBook("Onnobhubon","Science			
Fiction")	Number of Book(s): 3		
a2.addBook("Megher Upor Bari", "Horror")	Author Name: Humayun Ahmed		
print("======"")	Science Fiction: Onnobhubon, Ireena		
a2.printDetail()	Horror: Megher Upor Bari		
a2.addBook("Ireena", "Science Fiction")			
print("======"")			
a2.printDetail()			
print("======"")			

Using the TaxiLagbe app, users can share a single taxi with multiple people.

Implement the design of the **TaxiLagbe** class with the necessary properties so that the given output is produced for the provided driver code:

[Hint:

- 1. Each taxi can carry a maximum of 4 passengers
- 2. The addPassenger() method takes the last name of the passenger and ticket fare for that person in an underscore (_)-separated string.]

Dear Walker! Welcome to TaxiLagbe. Dear Wood! Welcome to TaxiLagbe. Dear Matt! Welcome to TaxiLagbe. Dear Wilson! Welcome to TaxiLagbe. Trip info for Taxi number: 1010-01 This taxi can only cover the Dhaka area. Total passengers: 4 Passenger lists: Walker, Wood, Matt, Wilson Total collected fare: 505 Taka Taxi Full! No more passengers can be added. Trip info for Taxi number: 1010-01 This taxi can only cover the Dhaka area. Total passengers: 4 Passenger lists: Walker, Wood, Matt, Wilson Total collected fare: 505 Taka Total passengers: 4 Dear Ronald! Welcome to TaxiLagbe. Dear Parker! Welcome to TaxiLagbe. Trip info for Taxi number: 1010-02 This taxi can only cover the Khulna area.

1	class Scope:
2	<pre>definit(self):</pre>
3	self.x, self.y = 1, 100
4	<pre>def met1(self):</pre>
5	x = 3
6	x = self.x + 1
7	self.y = self.y + self.x + 1
8	x = self.y + self.met2() + self.y
9	<pre>print(x, self.y)</pre>
10	<pre>def met2(self):</pre>
11	y = 0
12	<pre>print(self.x, y)</pre>
13	self.x = self.x + y
14	self.y = self.y + 200
15	return self.x + y

Write the output of the	х	V
following code:		
q2 = Scope()		
q2.met1()		
q2.met2()		
q2.met1()		
q2.met2()		

1	class Test3:
2	<pre>definit(self):</pre>
3	self.sum, $self.y = 0$, 0
4	<pre>def methodA(self):</pre>
5	x, y = 2, 3
6	msg = [0]
7	msg[0] = 3
8	y = self.y + msg[0]
9	self.methodB(msg)
10	x = self.y + msg[0]
11	self.sum = x + y + msg[0]
12	<pre>print(x, y, self.sum)</pre>
13	<pre>def methodB(self, mg2, mg1=6):</pre>
14	x = 0
15	self.y = self.y + mg2[0]
16	x = x + 33 + mg1
17	self.sum = self.sum + x + self.y
18	mg2[0] = self.y + mg1
19	mg1 = mg1 + x + 2
20	<pre>print(x, self.y, self.sum)</pre>

Write the output of the	х	у	sum
following code:			
t3 = Test3()			
t3.methodA()			