

Inspiring Excellence

Course Code:	CSE111
Course Title:	Programming Language II
Classwork No:	05
Topic:	OOP (HAS-A relationship and access modifier)
Number of tasks:	6

Design the program to get the output as shown.

Subtasks:

- 1. You will need to create 2 classes: **Teacher** and **Course**
- 2. Make all the variables in the Teacher class **private**.
- 3. Make all the variables in the Course class public.
- 4. Write the required codes in the Teacher and Course classes.

[You are not allowed to change the code below]

Write your code here for subtasks 1-4 **Output:** _____ t1 = Teacher("Saad Abdullah", "CSE") Name: Saad Abdullah t2 = Teacher("Mumit Khan", "CSE") Department: CSE t3 = Teacher("Sadia Kazi", "CSE") List of courses c1 = Course("CSE 110 Programming Language I") c2 = Course("CSE 111 Programming Language-II") CSE 110 Programming Language I c3 = Course("CSE 220 Data Structures") CSE 111 Programming Language-II c4 = Course("CSE 221 Algorithms") _____ c5 = Course("CSE 230 Discrete Mathematics") -----c6 = Course("CSE 310 Object Oriented Name: Mumit Khan Programming") Department: CSE c7 = Course("CSE 320 Data Communications") List of courses c8 = Course("CSE 340 Computer Architecture") t1.addCourse(c1) CSE 220 Data Structures t1.addCourse(c2) CSE 221 Algorithms CSE 230 Discrete Mathematics t2.addCourse(c3) ----t2.addCourse(c4) t2.addCourse(c5) t3.addCourse(c6) Name: Sadia Kazi t3.addCourse(c7) Department: CSE t3.addCourse(c8) List of courses _____ t1.printDetail() t2.printDetail() CSE 310 Object Oriented Programming t3.printDetail() **CSE 320 Data Communications CSE 340 Computer Architecture**

Please write the **Student** and **Department** class with the necessary properties so that the provided driver code generates the output given below. Make sure the **ID and CGPA** attributes in the **'Student'** class are private and cannot be accessed directly from outside of the class.

Driver Code	Output		
<pre>s1 = Student("Akib", 22301010, 3.29) s2 = Student("Reza", 22101010, 3.45) s3 = Student("Ruhan", 23101934, 4.00)</pre>	1=====================================		
<pre>print("1========="") cse = Department("CSE") cse.findStudent(22112233) print("2=========="")</pre>			
<pre>cse.addStudent(s1,s2,s3) print("3==========="") cse.details() print("4============="") cse.findStudent(22301010)</pre>	Department Name: CSE Number of student:3 Details of the students: Student name: Akib, ID: 22301010, cgpa: 3.29 Student name: Reza, ID: 22101010, cgpa: 3.45 Student name: Ruhan, ID: 23101934, cgpa: 4.0 4===================================		
<pre>print("5========="") s4 = Student("Nakib",22301010,3.22) cse.addStudent(s4) print("6==========="") s4.setId(21201220)</pre>			
cse.addStudent(s4) print("7==========="") cse.details() print("8=========="")	CGPA: 3.29 5===================================		
s5 = Student("Sakib", 22201010, 2.29) cse.addStudent(s5)			
<pre>print("9======="") cse.details()</pre>	Department Name: CSE Number of student:4 Details of the students: Student name: Akib, ID: 22301010, cgpa: 3.29 Student name: Reza, ID: 22101010, cgpa: 3.45 Student name: Ruhan, ID: 23101934, cgpa: 4.0 Student name: Nakib, ID: 21201220, cgpa: 3.22 8===================================		

Student name: Nakib, ID: 21201220, cgpa: 3.22 Student name: Sakib, ID: 22201010, cgpa: 2.29

Task 3

Class Description:

Spaceship: This class represents a spaceship. Each spaceship has a **name** and a **capacity** (the maximum weight it can carry).

Cargo: This class represents a piece of cargo. Each cargo item has a **name** and a **weight**. Both attributes should be **private** which means they cannot be accessed directly from outside of the class.

A Spaceship contains (HAS) Cargo. That means each spaceship can carry multiple cargo items, but the total weight of the cargo cannot exceed the spaceship's capacity.

Your task is to design the **Spaceship** and **Cargo** class with necessary properties so that the given output is produced for the provided driver code.

Driver Code	Output		
<pre># Creating spaceships falcon = Spaceship("Falcon", 50000) apollo = Spaceship("Apollo", 100000) enterprise = Spaceship("Enterprise", 220000) print("1.====================================</pre>	1.====================================		

Design the **Student** and the **Usis** class so that the following output is produced.

Note:

- 1. A student's email, password, and login status are None by default while creating an object of the Student class.
- 2. Your code should satisfy the conditions mentioned in the output only.

Driver Code	Output		
rakib = Student("Rakib", 12301455, "CSE")	Student object is created!		
print("1***************")	1***********		
usis_obj = Usis()	USIS is ready to use!		
print("2**************")	2********		
usis_obj.login(rakib)	Email and password need to be set.		
print("3**************")	3***********		
usis_obj.advising(rakib)	Please login to advise courses!		
print("4**************")	4*************		
rakib.email = "rakib@hotmail.com"	5***********		
rakib.password = "1234"	Login successful!		
print("5**************")	6***********		
usis_obj.login(rakib)	You haven't selected any courses.		
print("6***************")	7***********		
usis_obj.advising(rakib)	You need special approval to take more than 3		
print("7**************")	courses.		
usis_obj.advising(rakib, "CSE110", "PHY111", "MAT110",	8*******		
"CSE260")	Advising successful!		
print("8**************")	9*********		
usis_obj.advising(rakib, "CSE110", "PHY111", "MAT110")	Name: Rakib		
print("9***************")	ID: 12301455		
print(usis_obj.individualDetails(rakib))	Department: CSE		
	Advised courses: CSE110, PHY111, MAT110		

Design the required class/es so that the following output is generated.

[Hint: If you have stops at A, B, and C the fare from A to B is \$100, A to C is \$200 and B to C is \$100]

Driver Code	Output	
t1 = Train('T1-Express','New York','Manhattan','Brooklyn','Boston') print("1============="") p1 =Passenger("Naruto") t1.addPassenger(p1) p2 = Passenger("Sasuke","Manhattan") p3 = Passenger("Hinata","Manhattan","Brooklyn") print("2==========="") t1.addPassenger(p2,p3) print("3============"") t1.allPassengerDetails() print("4============="") t2 = Train('Europe-Express','London','Paris','Brussels','Turkey') print("5============="") p4 =Passenger("Max","London","Brussels") p5 = Passenger("Eleven","Paris") p6 = Passenger("Mike","Brussels") t2.addPassenger(p4,p5,p6) print("6==========="") t2.allPassengerDetails()	Welcome aboard on T1-Express Start: New York Destination: Boston 1====================================	

Task 6

Design the required class/es so that the following output is generated. Read the following description:

- 1. You may assume that to board a bus, a student must have the bus pass, and his/her destination must match the route of the bus.
- 2. Additionally, the default maximum capacity of the bus is 2.

Driver Code	Output
st1 = BracuStudent("Afif", "Mirpur") print("1===========") st2 = BracuStudent("Shanto", "Motijheel") st3 = BracuStudent("Taskin", "Mirpur") st1.show_details() st2.show_details() print("2==========") st3.show_details() print("3===========") bus1 = BracuBus("Mirpur") bus2 = BracuBus("Azimpur", 5)	1=====================================
bus1.show_details() bus2.show_details() print("4==========") st2.get_pass() st3.get_pass() print("5===========") st2.show_details() st3.show_details() print("6===========") bus1.board() print("7==========")	Bus Route: Azimpur Passengers Count: 0 (Max: 5) Passengers On Board: [] 4====================================
bus1.board(st1, st2) print("8=========") st1.get_pass() st2.home = "Mirpur" st1.show_details() st2.show_details() print("9==========") bus1.board(st1, st2, st3) print("10=========") bus1.show_details()	You don't have a bus pass! You got on the wrong bus! 8====================================
bus 1.5110W_uctalis()	Bus Route: Mirpur Passengers Count: 2 (Max: 2) Passengers On Board: ['Afif', 'Shanto']