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**

Task 1

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

Task 2

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

`s1 = Student("Akib", 22301010, 3.29)`
`s2 = Student("Reza", 22101010, 3.45)`
`s3 = Student("Ruhan", 23101934, 4.00)`
`print("1==========")`
`cse = Department("CSE")`
`cse.findStudent(22112233)`
`print("2==========")`
`cse.addStudent(s1,s2,s3)`
`print("3==========")`
`cse.details()`
`print("4==========")`
`cse.findStudent(22301010)`
`print("5==========")`
`s4 = Student("Nakib",22301010,3.22)`
`cse.addStudent(s4)`
`print("6==========")`
`s4.setId(21201220)`
`cse.addStudent(s4)`
`print("7==========")`
`cse.details()`
`print("8==========")`
`s5 = Student("Sakib",22201010,2.29)`
`cse.addStudent(s5)`
`print("9==========")`
`cse.details()`
	1==========
	Student with this ID doesn't exist, Please give a valid ID
	2==========
	Welcome to CSE department, Akib
	Welcome to CSE department, Reza
	Welcome to CSE department, Ruhan
	3==========
	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==========
	Student info:
	Student Name: Akib
	ID: 22301010
	CGPA: 3.29
	5==========
	Student with the same ID already exists, Please try with another ID
	6==========
	Welcome to CSE department, Nakib
	7==========
	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==========
	Welcome to CSE department, Sakib
	9==========
	Department Name: CSE
	Number of student:5
	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 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

`# Creating spaceships`
`falcon = Spaceship("Falcon", 50000)`
`apollo = Spaceship("Apollo", 100000)`
`enterprise = Spaceship("Enterprise", 220000)`
`print("1.==========")`
`# Creating cargo`
`gold = Cargo("Gold", 20000)`
`platinum = Cargo("Platinum", 25000)`
`dilithium = Cargo("Dilithium", 50000)`
`trilithium = Cargo("Trilithium", 70000)`
`neutronium = Cargo("Neutronium", 80000)`
`print("2.==========")`
`# Loading cargo onto spaceships`
`falcon.load_cargo(gold)`
`falcon.load_cargo(platinum)`
`falcon.display_details()`
`print("3.==========")`
`apollo.load_cargo(gold)  # Apollo will not`
	1.==========
	2.==========
	Spaceship Name: Falcon
	Capacity: 50000
	Current Cargo Weight: 45000
	Cargo: ['Gold', 'Platinum']
	3.==========
	Spaceship Name: Apollo
	Capacity: 100000
	Current Cargo Weight: 20000
	Cargo: ['Gold']
	4.==========
	Warning: Unable to load Neutronium inside Falcon. Exceeds capacity by 75000.
	5.==========
	Spaceship Name: Enterprise
	Capacity: 220000
	Current Cargo Weight: 200000
	Cargo: ['Dilithium', 'Trilithium', 'Neutronium']

```
reach its total capacity
apollo.display_details()
print("4.==================================")
falcon.load_cargo(neutronium)  # This should
exceed Falcon's capacity
print("5.==================================")
enterprise.load_cargo(dilithium)
enterprise.load_cargo(trilithium)
enterprise.load_cargo(neutronium)  # This
should not exceed Enterprise's capacity
enterprise.display_details()
```

Task 4

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

Task 5

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====================== Naruto welcome aboard 2====================== Sasuke welcome aboard Hinata welcome aboard 3====================== Name: Naruto,Start: New York,Destination: Boston,Fair: $300 Name: Sasuke,Start: Manhattan,Destination: Boston,Fair: $200 Name: Hinata,Start: Manhattan,Destination: Brooklyn,Fair: $100 4====================== Welcome aboard on Europe-Express Start: London Destination: Turkey 5====================== Max welcome aboard Eleven welcome aboard Mike welcome aboard 6====================== Name: Max,Start: London,Destination: Brussels,Fair: $200 Name: Eleven,Start: Paris,Destination: Turkey,Fair: $200 Name: Mike,Start: Brussels,Destination: Turkey,Fair: $100

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) 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==========") 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()	1========== Student Name: Afif Lives in Mirpur Have Bus Pass? False Student Name: Shanto Lives in Motijheel Have Bus Pass? False 2========== Student Name: Taskin Lives in Mirpur Have Bus Pass? False 3========== Bus Route: Mirpur Passengers Count: 0 (Max: 2) Passengers On Board: [] Bus Route: Azimpur Passengers Count: 0 (Max: 5) Passengers On Board: [] 4========== 5========== Student Name: Shanto Lives in Motijheel Have Bus Pass? True Student Name: Taskin Lives in Mirpur Have Bus Pass? True 6========== No passengers! 7========== You don't have a bus pass! You got on the wrong bus! 8========== Student Name: Afif Lives in Mirpur Have Bus Pass? True Student Name: Shanto Lives in Mirpur Have Bus Pass? True 9========== Afif boarded the bus. Shanto boarded the bus. Bus is full! 10========== Bus Route: Mirpur Passengers Count: 2 (Max: 2) Passengers On Board: ['Afif', 'Shanto']