Inspiring Excellence

Course Title: Programming Language II
Course Code: CSE 111
Lab Assignment no: 5

Task 1

Write a class called Circle with the required constructor and methods to get the following output.

Subtasks:

1. Create a **class** called Circle.

2. Create the required **constructor**. Use **Encapsulation** to protect the variables. [**Hint:** Assign the variables in **private**]

3. Create **getRadius()** and **setRadius()** method to access variables.

4. Create a **method** called area to calculate the area of circles.

[You are not allowed to change the code below]

# Write your code here for subtasks 1-5 c1 = Circle(4) print("First circle radius:" , c1.getRadius()) print("First circle area:" , c1.area()) c2 = Circle(5) print("Second circle radius:" , c2.getRadius()) print("Second circle area:" , c2.area())	**Output:** First circle radius: 4 First circle area: 50.26548245743669 Second circle radius: 5 Second circle area: 78.53981633974483

Task 2

Write a class called Triangle with the required constructor and methods to get the following output.

Subtasks:

1. Create a **class** called Triangle.

2. Create the required **constructor**. Use **Encapsulation** to protect the variables. [**Hint:** Assign the variables in **private**]

3. Create **getBase(), getHeight(), setBase()** and **setHeight()** methods to access variables.

4. Create a **method** called area to calculate the area of triangles.

[You are not allowed to change the code below]

# Write your code here for subtasks 1-5 t1 = Triangle(10, 5) print("First Triangle Base:" , t1.getBase()) print("First Triangle Height:" , t1.getHeight()) print("First Triangle area:" ,t1.area()) t2 = Triangle(5, 3) print("Second Triangle Base:" , t2.getBase()) print("Second Triangle Height:" , t2.getHeight()) print("Second Triangle area:" ,t2.area())	**Output:** First Triangle Base: 10 First Triangle Height: 5 First Triangle area: 25.0 Second Triangle Base: 5 Second Triangle Height: 3 Second Triangle area: 7.5

Task 3

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 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()	**Output:** ========== 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 4

Design the program to get the output as shown.

Subtasks:

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

Hints:

- Create a list in team class to store the player's name in that list
- Use constructor overloading technique for Team class

[You are not allowed to change the code below]

# Write your code here for subtasks 1-4 b = Team() b.setName('Bangladesh') mashrafi = Player("Mashrafi") b.addPlayer(mashrafi) tamim = Player("Tamim") b.addPlayer(tamim) b.printDetail() a = Team("Australia") ponting = Player("Ponting") a.addPlayer(ponting) lee = Player("Lee") a.addPlayer(lee) a.printDetail()	**Output:** ========== Team: Bangladesh List of Players: ['Mashrafi', 'Tamim'] ========== ========== Team: Australia List of Players: ['Ponting', 'Lee'] ==========

Task 5

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 6

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 7

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 passenger! 7========== You don't have bus pass! You got on 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']

Task 8

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

Read the following description:

- The Library class has two dictionaries: one contains borrower information(the name of borrowers and the number of books they borrowed) and the other contains book availability information (book type and their remaining number)
- A reader cannot borrow more than 5 books.
- If a book's availability is 0 in the Library, then the reader cannot borrow that book.
- The readerInfo method in the Reader class prints the type and the number of all books borrowed if no parameter is passed, else it prints the number of books borrowed of the specific type mentioned in the parameter. You may use the default argument for this.

Driver Code	Output
L1=Library('Dhaka',{'Arts':15,'Fiction':135,'Politics':2,'Science':11,' Poetry':15}) L1.details() print("1----------") r1=Reader('Aladdin') r1.borrow(L1,'Arts','Fiction','Fiction','Politics') print("2----------") r1.borrow(L1,'Politics','Fiction') print("3----------") r1.readerInfo() print("4----------") r1.readerInfo('Fiction') print("5----------") L1.details() print("6----------") r2=Reader('Jasmine') r2.borrow(L1,'Politics','Poetry') print("7----------") r2.readerInfo() print("8----------") L1.details()	Dhaka Library details Borrower details: {} Books availability: {'Arts': 15, 'Fiction': 135, 'Politics': 2, 'Science': 11, 'Poetry': 15} 1---------- Arts book is borrowed successfully. Fiction book is borrowed successfully. Fiction book is borrowed successfully. Politics book is borrowed successfully. 2---------- Politics book is borrowed successfully. You cannot borrow more than 5 books. 3---------- Aladdin, you have 5 book(s) with you. Books on Arts: 1 Books on Fiction: 2 Books on Politics: 2 4---------- Aladdin, you have 2 Fiction book(s) with you. 5---------- Dhaka Library details Borrower details: {'Aladdin': 5} Books availability: {'Arts': 14, 'Fiction': 133, 'Politics': 0, 'Science': 11, 'Poetry': 15} 6---------- Politics books are not available at the moment. Poetry book is borrowed successfully. 7----------

	Jasmine, you have 1 book(s) with you. Books on Poetry: 1 8---------- Dhaka Library details Borrower details: {'Aladdin': 5, 'Jasmine': 1} Books availability: {'Arts': 14, 'Fiction': 133, 'Politics': 0, 'Science': 11, 'Poetry': 14}