Course Code:	CSE111
Course Title:	Programming Language II
Classwork No:	06
Topic:	OOP (HAS-A relationship)
Number of tasks:	4

Task 1

Design the program to get the output as shown.

[You are not allowed to change the code below]

# *Write your code here*	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.

Driver Code	Output

``python
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

Design the **Vaccine** and **Person** class so that the following expected output is generated.

[N.B: Students will get vaccines on a priority basis. So, age for students doesn't matter]

Driver Code	Output
# Write your code here astra = Vaccine("AstraZeneca", "UK", 60) modr = Vaccine("Moderna", "UK", 30) sin = Vaccine("Sinopharm", "China", 30) p1 = Person("Bob", 21, "Student") print("===============================") p1.pushVaccine(astra) print("===============================") p1.showDetail() print("===============================") p1.pushVaccine(sin, "2nd Dose") print("===============================") p1.pushVaccine(astra, "2nd Dose") print("===============================") p1.showDetail() print("===============================") p2 = Person("Carol", 23, "Actor") print("===============================") p2.pushVaccine(sin) print("===============================") p3 = Person("David", 34) print("===============================") p3.pushVaccine(modr) print("===============================") p3.showDetail() print("===============================") p3.pushVaccine(modr, "2nd Dose")	=============================== 1st dose done for Bob =============================== Name: Bob Age: 21 Type: Student Vaccine name: AstraZeneca 1st dose: Given 2nd dose: Please come after 60 days =============================== Sorry Bob, you can't take 2 different vaccines =============================== 2nd dose done for Bob =============================== Name: Bob Age: 21 Type: Student Vaccine name: AstraZeneca 1st dose: Given 2nd dose: Given =============================== =============================== Sorry Carol, Minimum age for taking vaccines is 25 years now. =============================== =============================== 1st dose done for David =============================== Name: David Age: 34 Type: General Citizen Vaccine name: Moderna 1st dose: Given 2nd dose: Please come after 30 days =============================== 2nd dose done for David

Task 4

```python
1   class msgClass:
2       def __init__(self):
3           self.content = 0
4   class Q5:
5       def __init__(self):
6           self.sum = 1
7           self.x = 2
8           self.y = 3
9       def methodA(self):
10          x, y = 1, 1
11          msg = []
12          myMsg = msgClass()
13          myMsg.content = self.x
14          msg.append(myMsg)
15          msg[0].content = self.y + myMsg.content
16          self.y = self.y + self.methodB(msg[0])
17          y = self.methodB(msg[0]) + self.y
18          x = y + self.methodB(msg[0], msg)
19          self.sum = x + y + msg[0].content
20          print(x," ", y," ", self.sum)
21      def methodB(self, mg1, mg2 = None):
22          if mg2 == None:
23              x, y = 5, 6
24              y = self.sum + mg1.content
25              self.y = y + mg1.content
26              x = self.x + 7 +mg1.content
27              self.sum = self.sum + x + y
```

28	` self.x = mg1.content + x +8`
29	` print(x, " ", y," ", self.sum)`
30	` return y`
31	` else:`
32	` x = 1`
33	` self.y += mg2[0].content`
34	` mg2[0].content = self.y + mg1.content`
35	` x = x + 4 + mg1.content`
36	` self.sum += x + self.y`
37	` mg1.content = self.sum - mg2[0].content`
38	` print(self.x, " ",self.y," ", self.sum)`
39	` return self.sum`

What is the output of the following code sequence?	x	y	sum
q = Q5()			
q.methodA()			