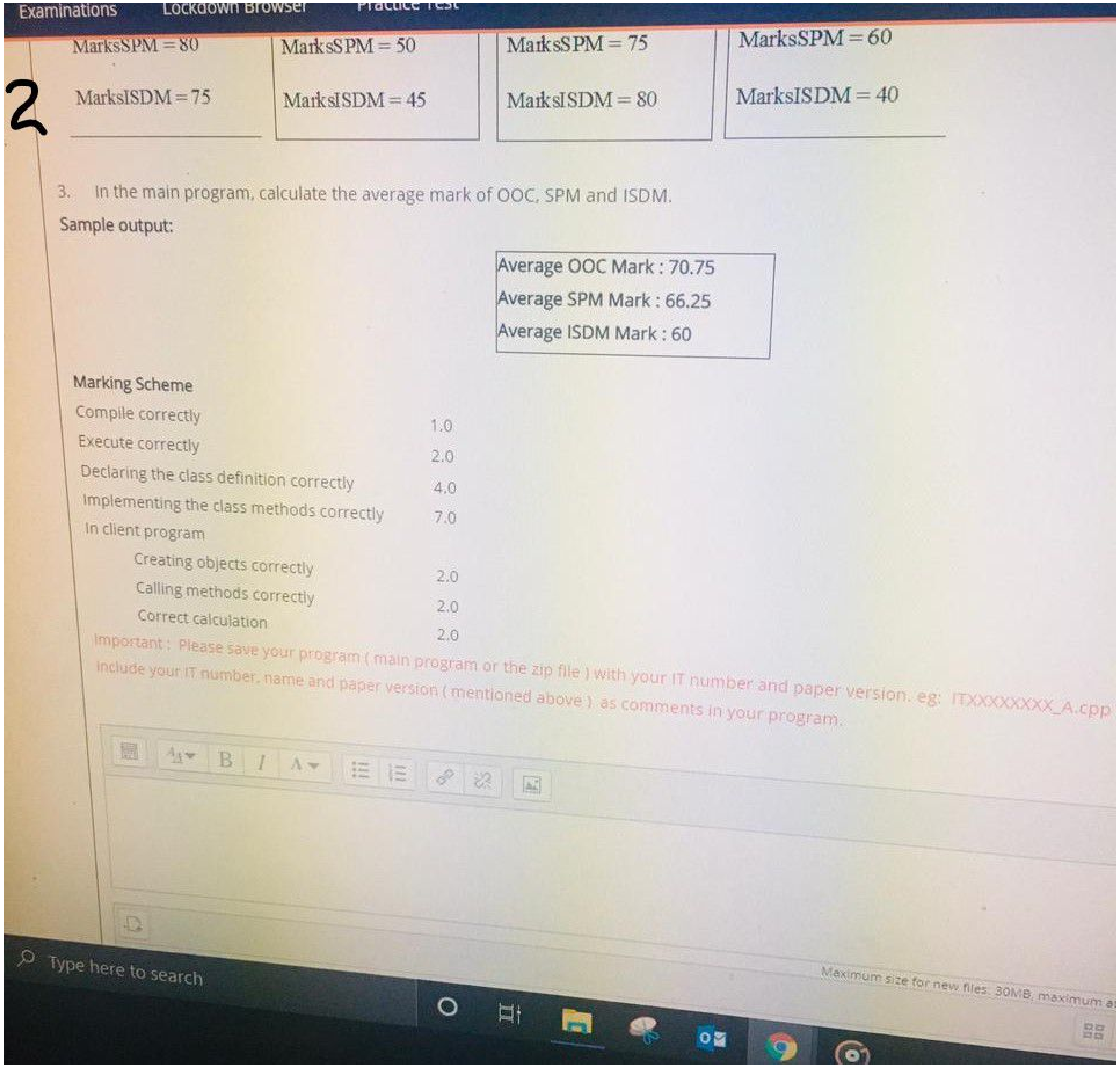


Examinations Lockdown Browser Practice Test

2

MarksSPM = 80	MarksSPM = 50	MarksSPM = 75	MarksSPM = 60
MarksISDM = 75	MarksISDM = 45	MarksISDM = 80	MarksISDM = 40

3. In the main program, calculate the average mark of OOC, SPM and ISDM.

Sample output:

```
Average OOC Mark : 70.75
Average SPM Mark : 66.25
Average ISDM Mark : 60
```

Marking Scheme

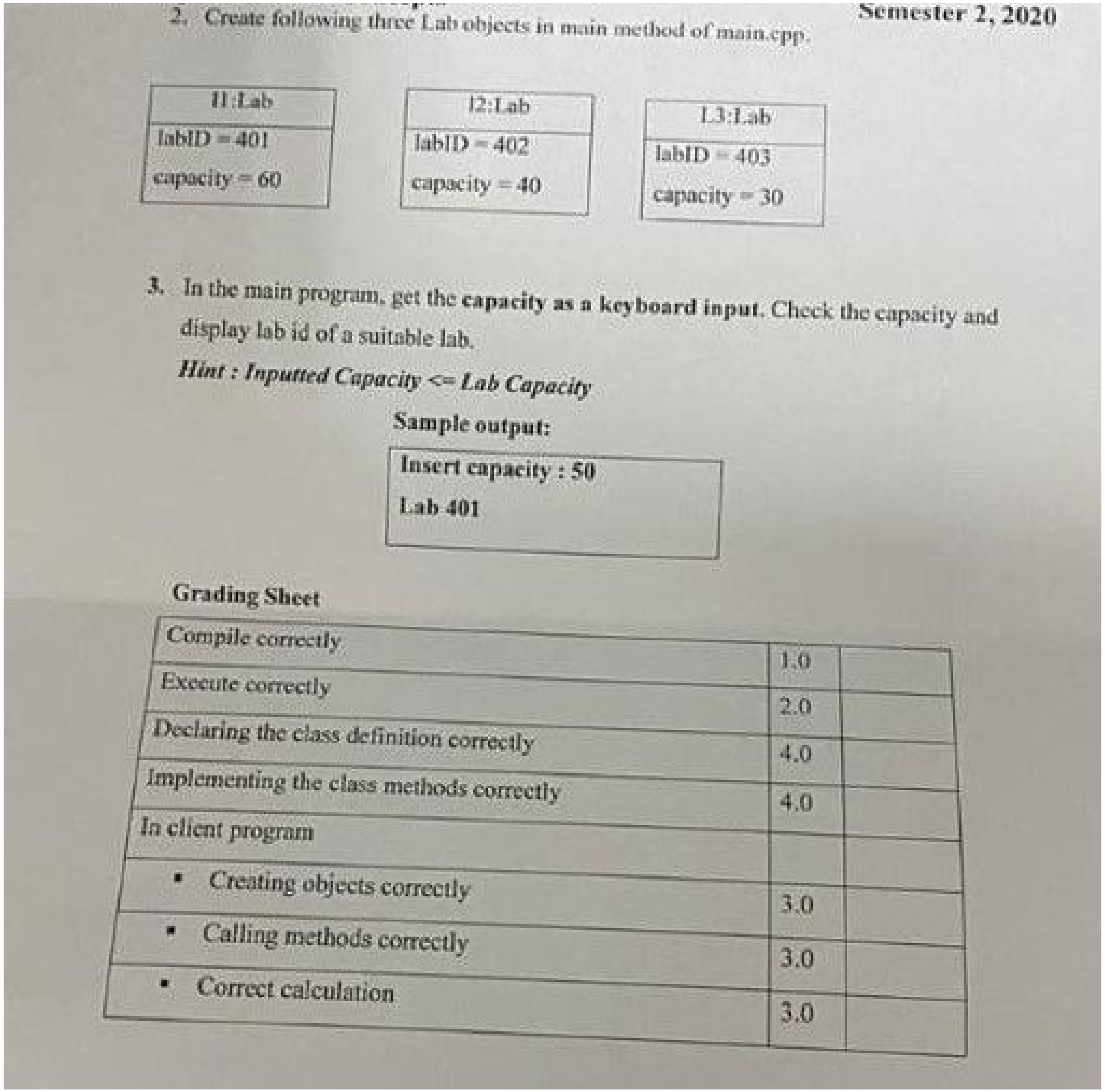
Compile correctly	1.0
Execute correctly	2.0
Declaring the class definition correctly	4.0
Implementing the class methods correctly	7.0
In client program	
Creating objects correctly	2.0
Calling methods correctly	2.0
Correct calculation	2.0

Important : Please save your program (main program or the zip file) with your IT number and paper version. eg: ITXXXXXXXX_A.cpp

Include your IT number, name and paper version (mentioned above) as comments in your program.

Maximum size for new files: 30MB, maximum a

Type here to search



2. Create following three Lab objects in main method of main.cpp.

l1:Lab

labID = 401
capacity = 60

l2:Lab

labID = 402
capacity = 40

L3:Lab

labID = 403
capacity = 30

3. In the main program, get the **capacity as a keyboard input**. Check the capacity and display lab id of a suitable lab.

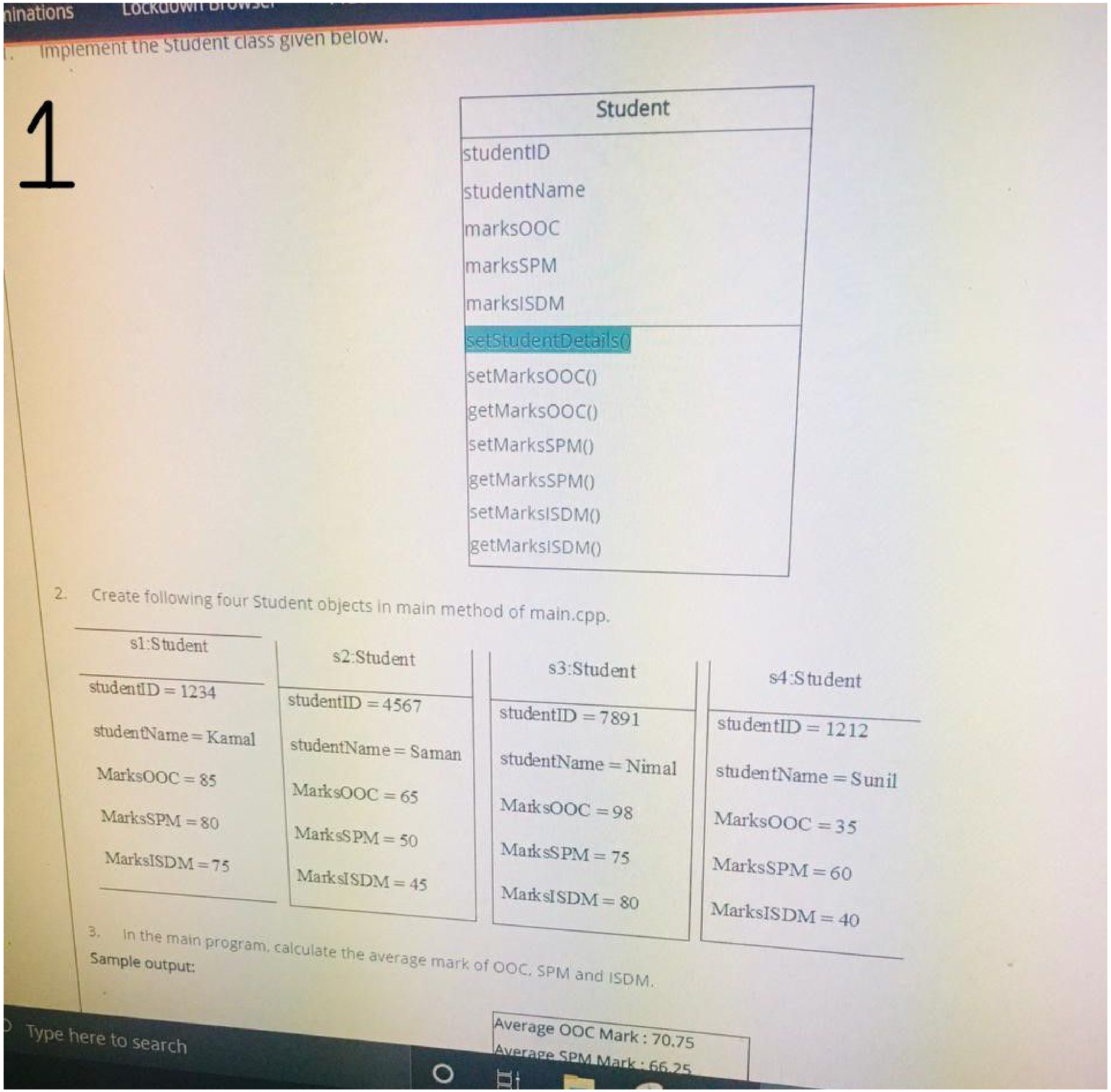
Hint : Inputted Capacity <= Lab Capacity

Sample output:

```
Insert capacity : 50
Lab 401
```

Grading Sheet

Compile correctly	1.0	
Execute correctly	2.0	
Declaring the class definition correctly	4.0	
Implementing the class methods correctly	4.0	
In client program		
▪ Creating objects correctly	3.0	
▪ Calling methods correctly	3.0	
▪ Correct calculation	3.0	



ninations   Lockdown Browser

1. Implement the Student class given below.

1

Student
studentID studentName marksOOC marksSPM marksISDM

setStudentDetails()
setMarksOOC()
getMarksOOC()
setMarksSPM()
getMarksSPM()
setMarksISDM()
getMarksISDM()

2. Create following four Student objects in main method of main.cpp.

s1:Student	s2:Student	s3:Student	s4:Student
studentID = 1234	studentID = 4567	studentID = 7891	studentID = 1212
studentName = Kamal	studentName = Saman	studentName = Nimal	studentName = Sunil
MarksOOC = 85	MarksOOC = 65	MarksOOC = 98	MarksOOC = 35
MarksSPM = 80	MarksSPM = 50	MarksSPM = 75	MarksSPM = 60
MarksISDM = 75	MarksISDM = 45	MarksISDM = 80	MarksISDM = 40

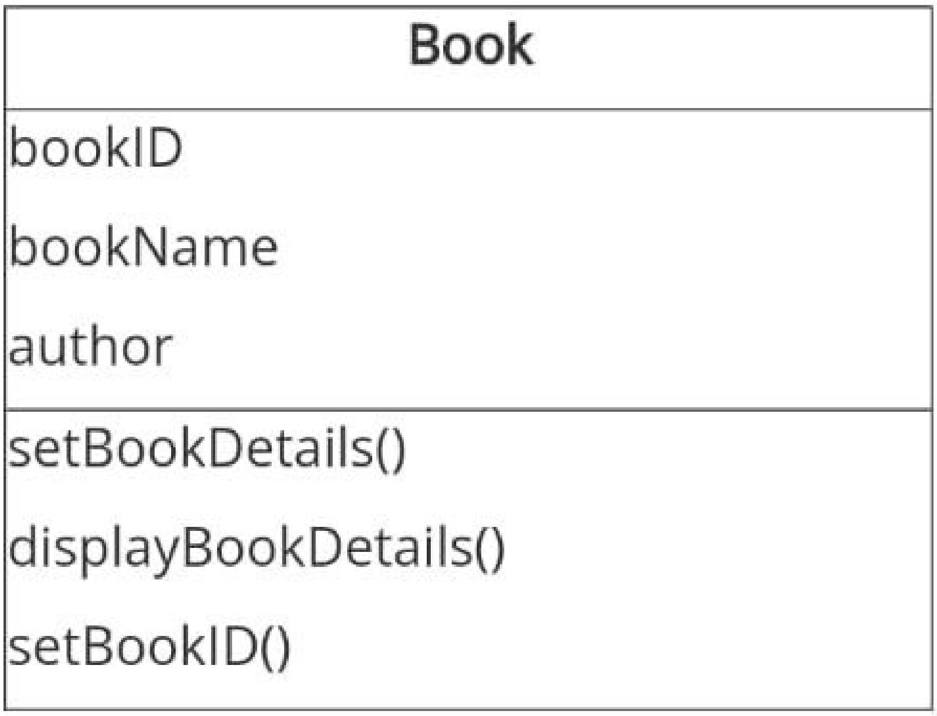
3. In the main program, calculate the average mark of OOC, SPM and ISDM.

Sample output:

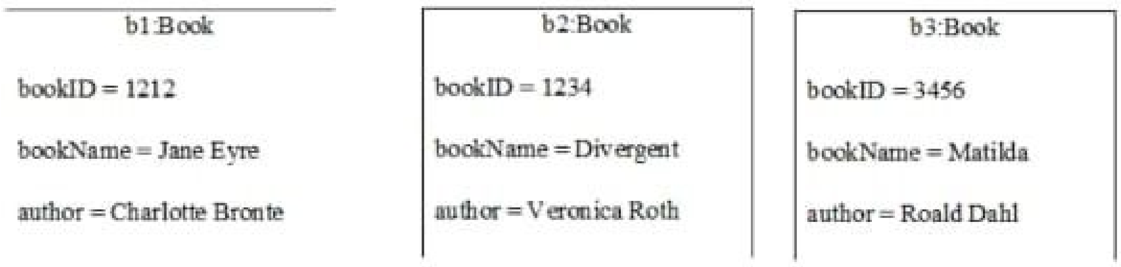
```
Average OOC Mark : 70.75
Average SPM Mark : 66.25
```

Type here to search

1. Implement the Book class given below.



2. Create following three Book objects in main method of main.cpp.



3. In the main program, get new book ids for all three books as keyboard inputs, and set the new book ids. Display updated book details

Sample output:

Sample output:

```
Input new book ID 1 : 11
Input new book ID 2 : 12
Input new book ID 3 : 13

BookID = 11
BookName = Jane Eyre
Author = Charlotte Bronte

BookID = 12
BookName = Divergent
Author = Veronica Roth

BookID = 13
BookName = Matilda
Author = Roald Dahl
```

Marking Scheme

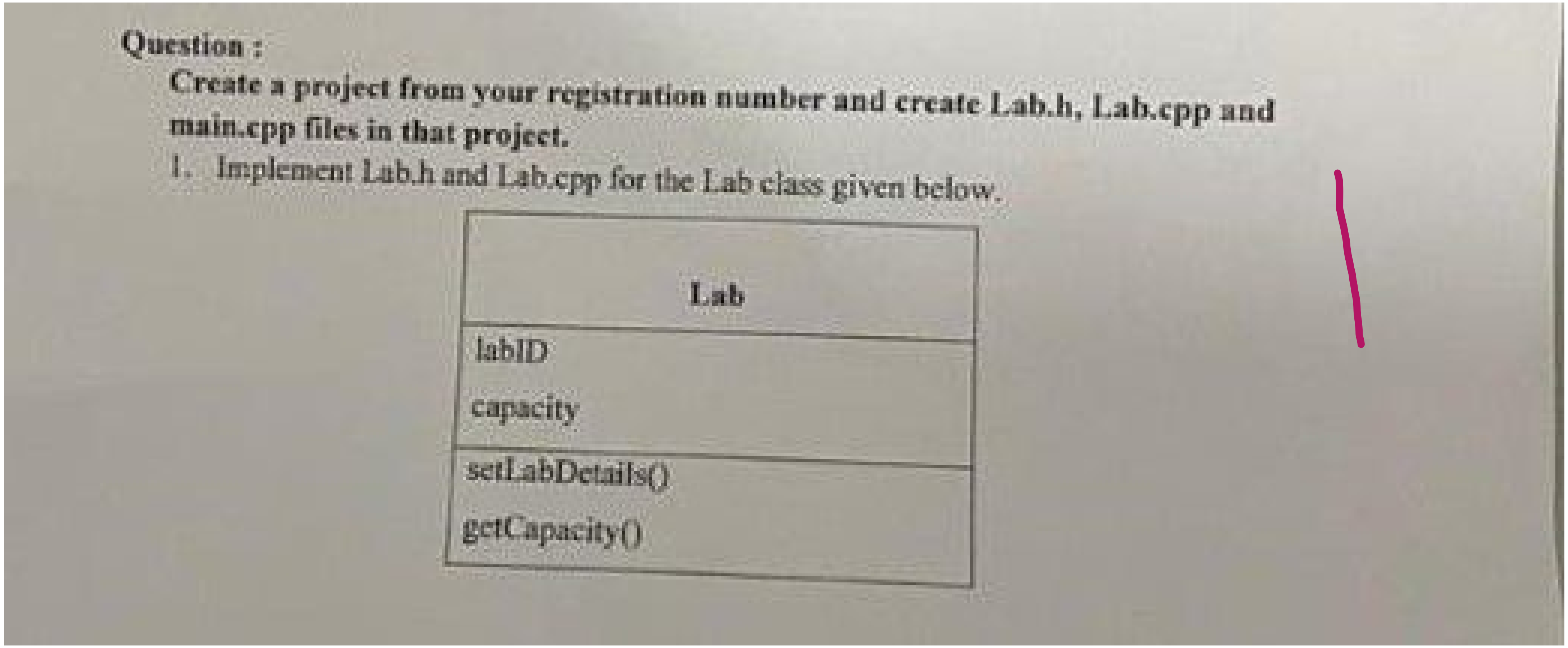
Compile correctly
1.0

Execute correctly
2.0

Declaring the class definition correctly
4.0

Implementing the class methods correctly
7.0

In client program



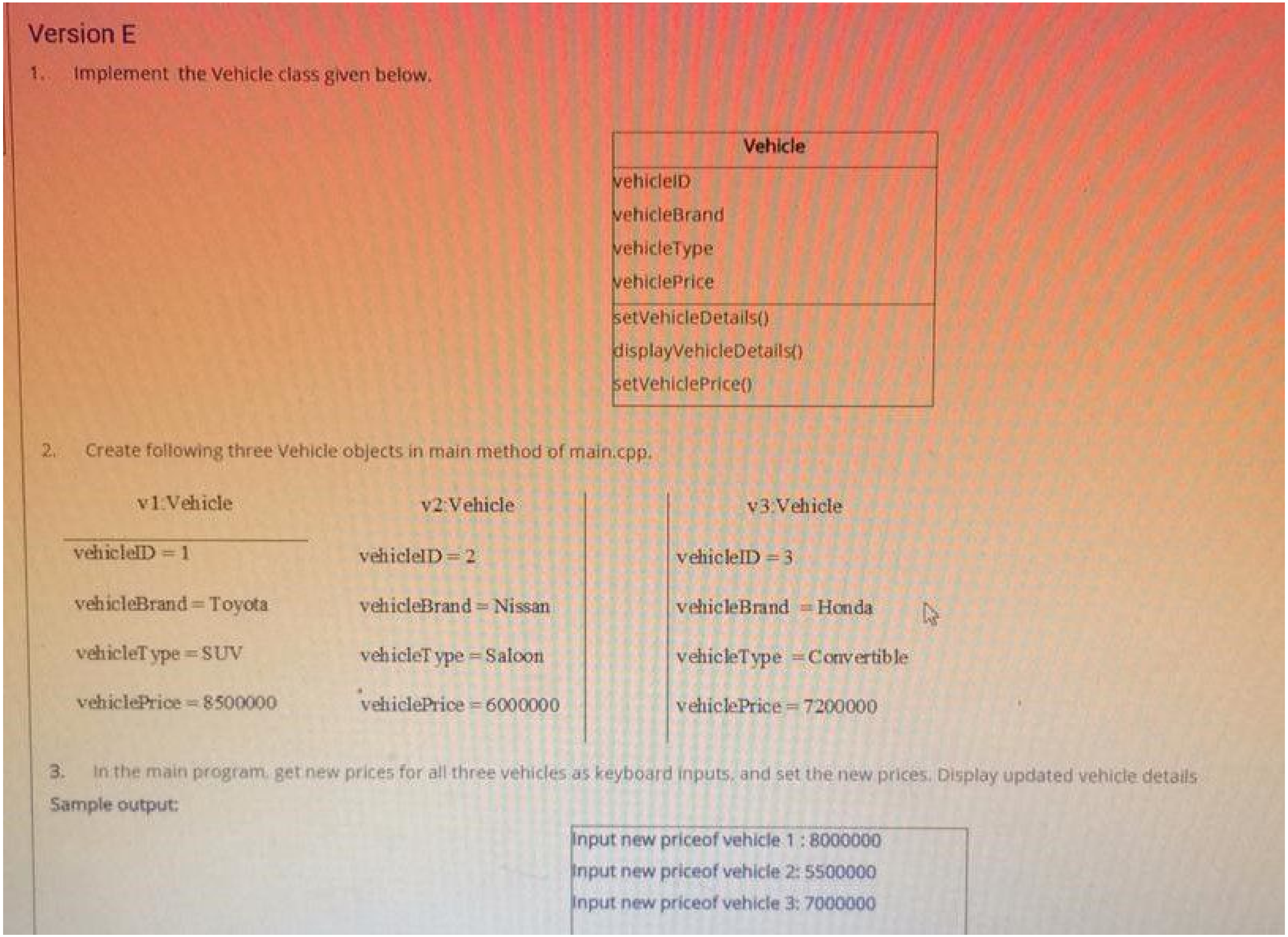
Question :

Create a project from your registration number and create Lab.h, Lab.cpp and main.cpp files in that project.

1. Implement Lab.h and Lab.cpp for the Lab class given below.

Lab
labID capacity

setLabDetails()
getCapacity()



Version E

1. Implement the Vehicle class given below.

Vehicle
vehicleID vehicleBrand vehicleType vehiclePrice

setVehicleDetails()
displayVehicleDetails()
setVehiclePrice()

2. Create following three Vehicle objects in main method of main.cpp.

v1:Vehicle	v2:Vehicle	v3:Vehicle
vehicleID = 1	vehicleID = 2	vehicleID = 3
vehicleBrand = Toyota	vehicleBrand = Nissan	vehicleBrand = Honda
vehicleType = SUV	vehicleType = Saloon	vehicleType = Convertible
vehiclePrice = 8500000	vehiclePrice = 6000000	vehiclePrice = 7200000

3. In the main program, get new prices for all three vehicles as keyboard inputs, and set the new prices. Display updated vehicle details

Sample output:

```
Input new priceof vehicle 1 : 8000000
Input new priceof vehicle 2: 5500000
Input new priceof vehicle 3: 7000000
```

Version H

1. Implement the Guest class given below.

Guest
guestID
guestName
ratePerDay
numberOfDays
setGuestDetails()
displayGuestDetails()
calculateGuestBill()

Hint: calculateGuestBill() method is to calculate the bill (ratePerDay * numberOfDays).

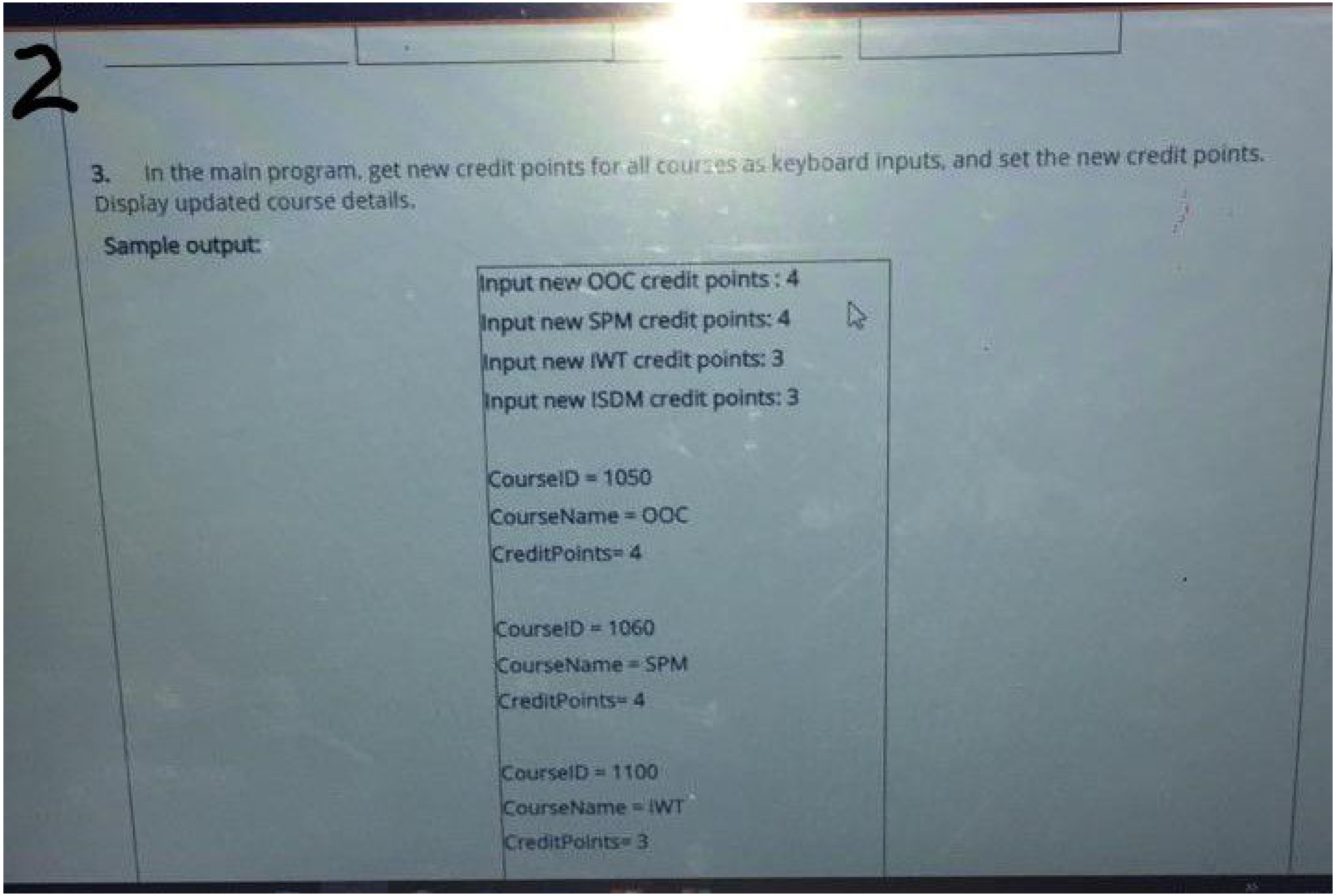
2. Create following guest objects in main method of main.cpp.

g1:Guest
guestID = 1212

g2:Guest
guestID = 1122
guestName = Ben

g3:Guest
guestID = 1234
guestName = Ruby

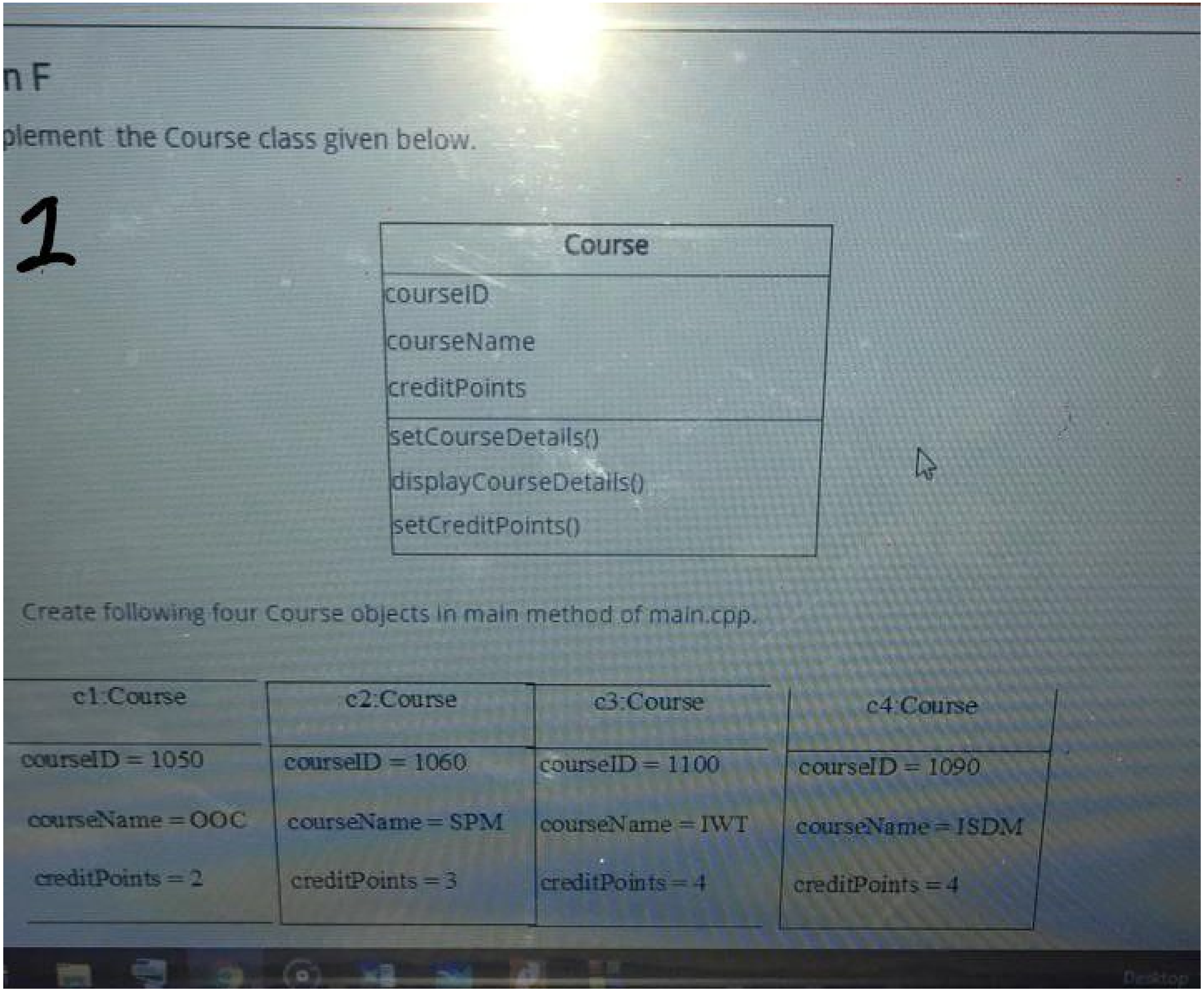
2



3. In the main program, get new credit points for all courses as keyboard inputs, and set the new credit points. Display updated course details.

Sample output:

```
Input new OOC credit points : 4
Input new SPM credit points: 4
Input new IWT credit points: 3
Input new ISDM credit points: 3

CourseID = 1050
CourseName = OOC
CreditPoints= 4

CourseID = 1060
CourseName = SPM
CreditPoints= 4

CourseID = 1100
CourseName = IWT
CreditPoints= 3
```



n F

plement the Course class given below.

1

Course
courseID
courseName
creditPoints
setCourseDetails()
displayCourseDetails()
setCreditPoints()

Create following four Course objects in main method of main.cpp.

c1:Course	c2:Course	c3:Course	c4:Course
courseID = 1050	courseID = 1060	courseID = 1100	courseID = 1090
courseName = OOC	courseName = SPM	courseName = IWT	courseName = ISDM
creditPoints = 2	creditPoints = 3	creditPoints = 4	creditPoints = 4

3. In the main program, calculate the bill of all guests using calculateGuestBill()method, and display the total bill of each guest with guestID and guestName.

Sample output:

```
Guest ID = 1212
Guest Name = Jared
BillAmount = 18000

Guest ID = 1122
Guest Name = Ben
Bill Amount = 9000

Guest ID = 1234
Guest Name = Ruby
Bill Amount = 11500
```

Marking Scheme	
Compile correctly	1.0
Execute correctly	2.0
Declaring the class definition correctly	4.0
Implementing the class methods correctly	7.0

Hint: calculateGuestBill() method is to calculate the bill (ratePerDay * numberOfDays).

2. Create following guest objects in main method of main.cpp.

g1:Guest
guestID = 1212
guestName = Jared
ratePerDay = 4500
numberOfDays = 4

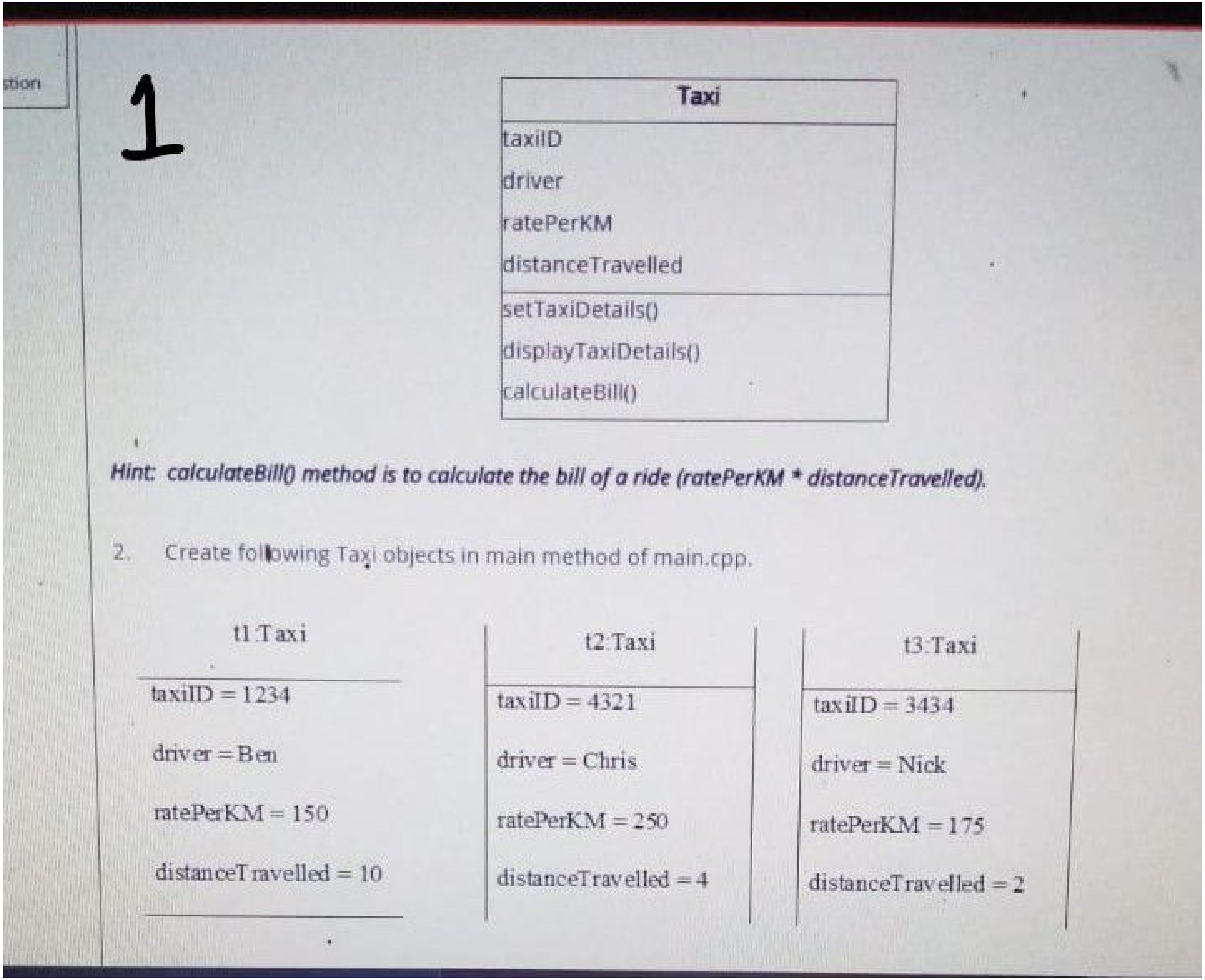
g2:Guest
guestID = 1122
guestName = Ben
ratePerDay = 3000
numberOfDays = 3

g3:Guest
guestID = 1234
guestName = Ruby
ratePerDay = 5750
numberOfDays =2

3. In the main program, calculate the bill of all guests using calculateGuestBill()method, and display the total bill of each guest with guestID and guestName.

Sample output:

```
Guest ID = 1212
Guest Name = Jared
BillAmount = 18000


Guest ID = 1122
Guest Name = Ben
```



tion

1

Taxi
taxiID
driver
ratePerKM
distanceTravelled
setTaxiDetails()
displayTaxiDetails()
calculateBill()

Hint: calculateBill() method is to calculate the bill of a ride (ratePerKM * distanceTravelled).

2. Create following Taxi objects in main method of main.cpp.

t1:Taxi	t2:Taxi	t3:Taxi
taxiID = 1234	taxiID = 4321	taxiID = 3434
driver = Ben	driver = Chris	driver = Nick
ratePerKM = 150	ratePerKM = 250	ratePerKM = 175
distanceTravelled = 10	distanceTravelled = 4	distanceTravelled = 2

Version L

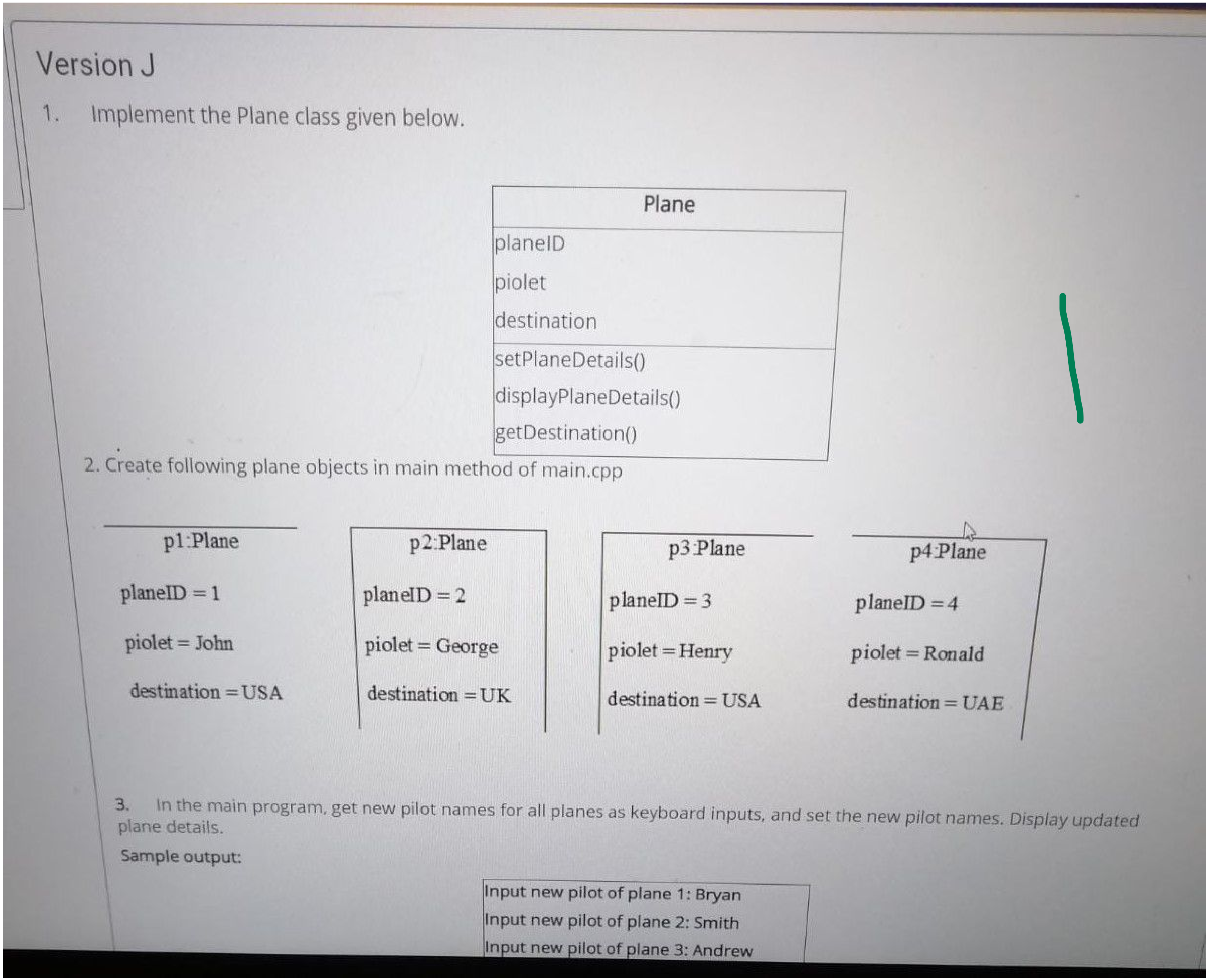
1. Implement the Doctor class given below.

Doctor
doctorID
doctorName
specialization
hospital
setDoctorDetails()
displayDoctorDetails()
getSpecialization()

2. Create following Doctor objects in main method of main.cpp.

d1:Doctor
doctorID =1
doctorName = Dr. Sunil
specialization = Neurologist

d2:Doctor
doctorID = 2
doctorName = Dr. Yasantha
specialization = Oncologist



Version J

1. Implement the Plane class given below.

Plane
planeID
piolet
destination
setPlaneDetails()
displayPlaneDetails()
getDestination()

2. Create following plane objects in main method of main.cpp

p1:Plane	p2:Plane	p3:Plane	p4:Plane
planeID = 1	planeID = 2	planeID = 3	planeID = 4
piolet = John	piolet = George	piolet = Henry	piolet = Ronald
destination = USA	destination = UK	destination = USA	destination = UAE

3. In the main program, get new pilot names for all planes as keyboard inputs, and set the new pilot names. Display updated plane details.

Sample output:

```
Input new pilot of plane 1: Bryan
Input new pilot of plane 2: Smith
Input new pilot of plane 3: Andrew
```

2

ratePerKM = 150
distanceTravelled = 10
	ratePerKM = 250
	distanceTravelled = 4
		ratePerKM = 175
		distanceTravelled = 2

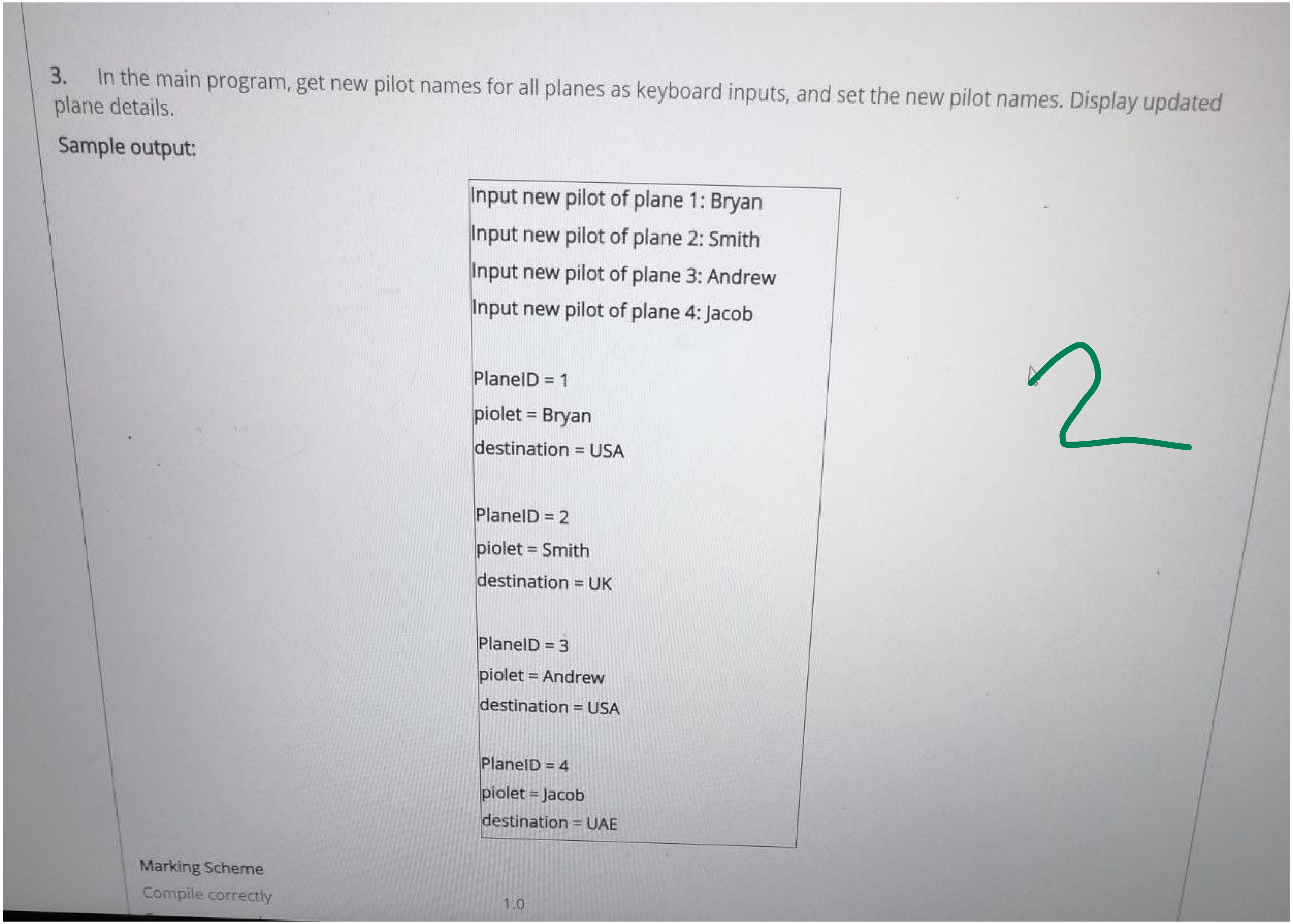
3. In the main program, calculate the bill of all taxis using calculateBill()method, and display the total bill of each taxi with taxiID and driver.

Sample output:

```
Taxi ID = 1234
Driver Name = Ben
BillAmount = 1500

Taxi ID = 4321
Driver Name = Chris
BillAmount = 1000

Taxi ID = 3434
Driver Name =  Nick
BillAmount = 350
```

Marking Scheme

Compile correctly	1.0
Execute correctly	2.0
Declaring the class definition correctly	4.0
Implementing the class methods correctly	7.0



3. In the main program, get new pilot names for all planes as keyboard inputs, and set the new pilot names. Display updated plane details.

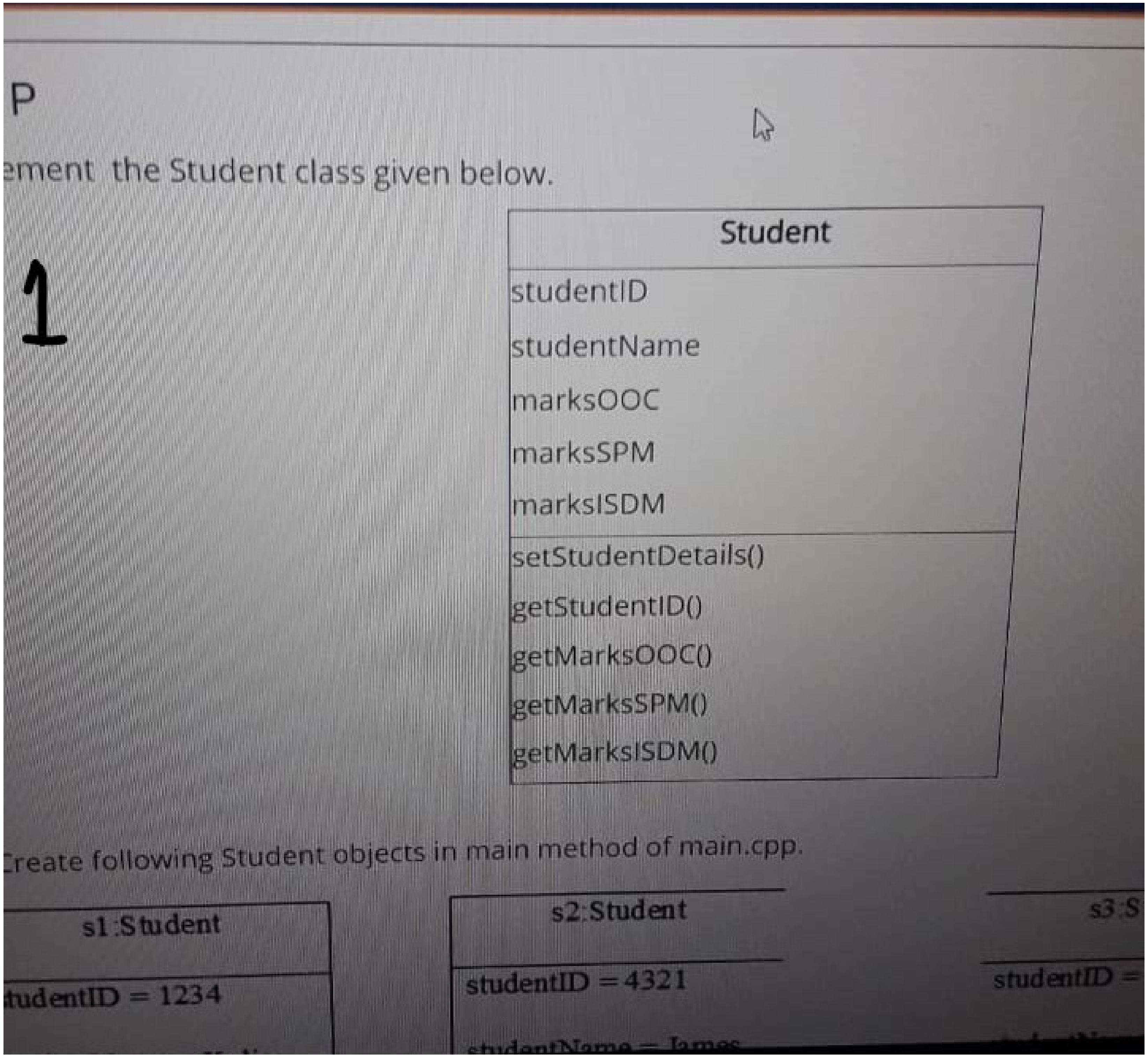
Sample output:

```
Input new pilot of plane 1: Bryan
Input new pilot of plane 2: Smith
Input new pilot of plane 3: Andrew
Input new pilot of plane 4: Jacob

PlaneID = 1
piolet = Bryan
destination = USA

PlaneID = 2
piolet = Smith
destination = UK

PlaneID = 3
piolet = Andrew
destination = USA

PlaneID = 4
piolet = Jacob
destination = UAE
```

2

Marking Scheme

Compile correctly 1.0



P

ement the Student class given below.

1

Student
studentID studentName marksOOC marksSPM marksISDM

setStudentDetails()
getStudentID()
getMarksOOC()
getMarksSPM()
getMarksISDM()

Create following Student objects in main method of main.cpp.

s1:Student
tudentID = 1234

s2:Student

studentID = 4321
studentName = James

s3:S
studentID =

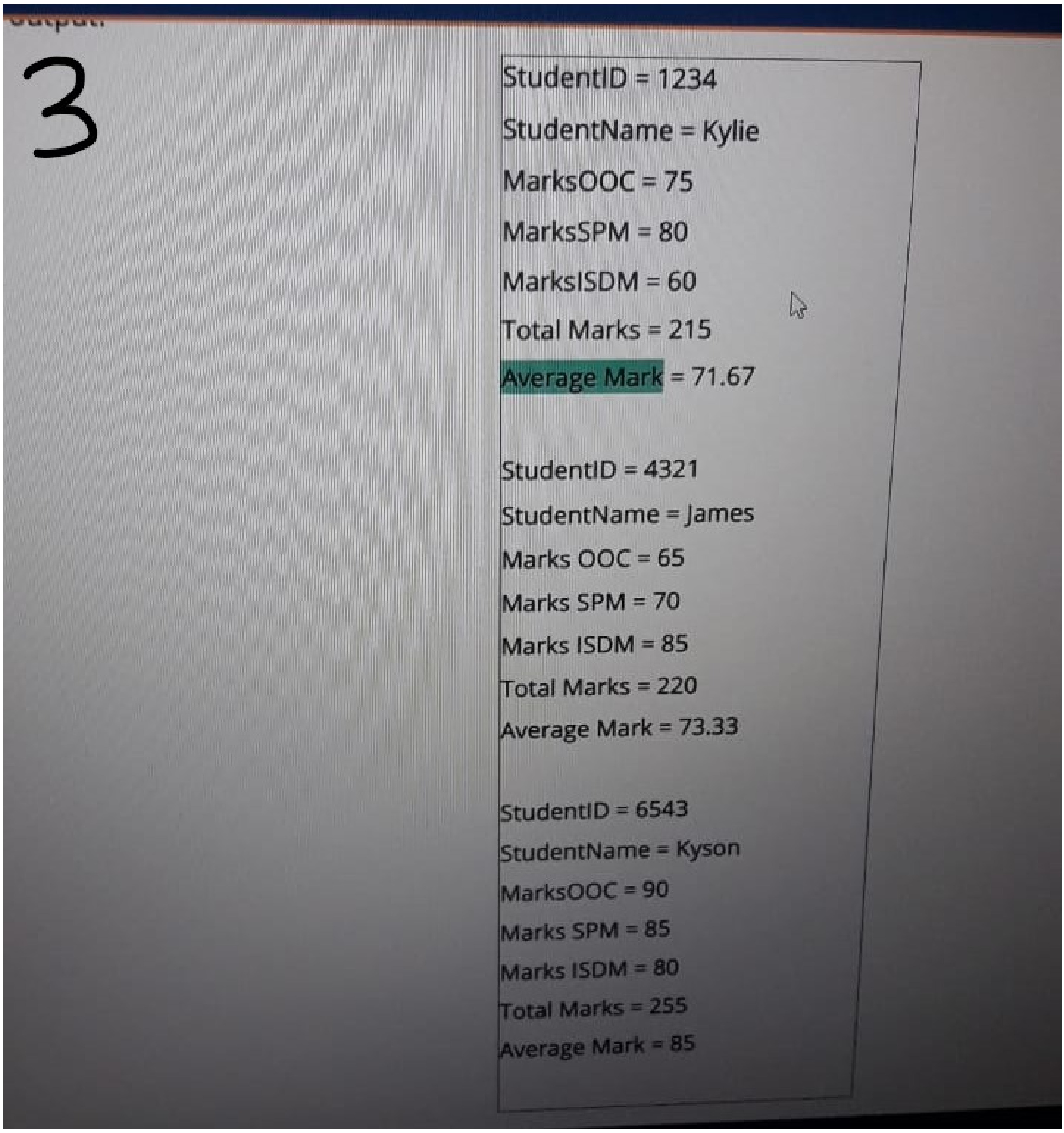
3. In the main program, get new hospital for all courses as keyboard inputs, and set the new hospitals. Display updated doctor details.

Sample output:

```
Input new hospital of doctor 1 : Nawaloka
Input new hospital of doctor 2 : Central
Input new hospital of doctor 3 : Delmon

DoctorID =1
DoctorName =  Dr. Sunil
Specialization = Neurologist
Hospital = Nawaloka

DoctorID = 2
DoctorName = Dr. Yasantha
Specialization = Oncologist
Hospital = Central

DoctorID = 3
DoctorName = Dr. Godvin
Specialization = Cardiologist
Hospital = Delmon
```

2

3

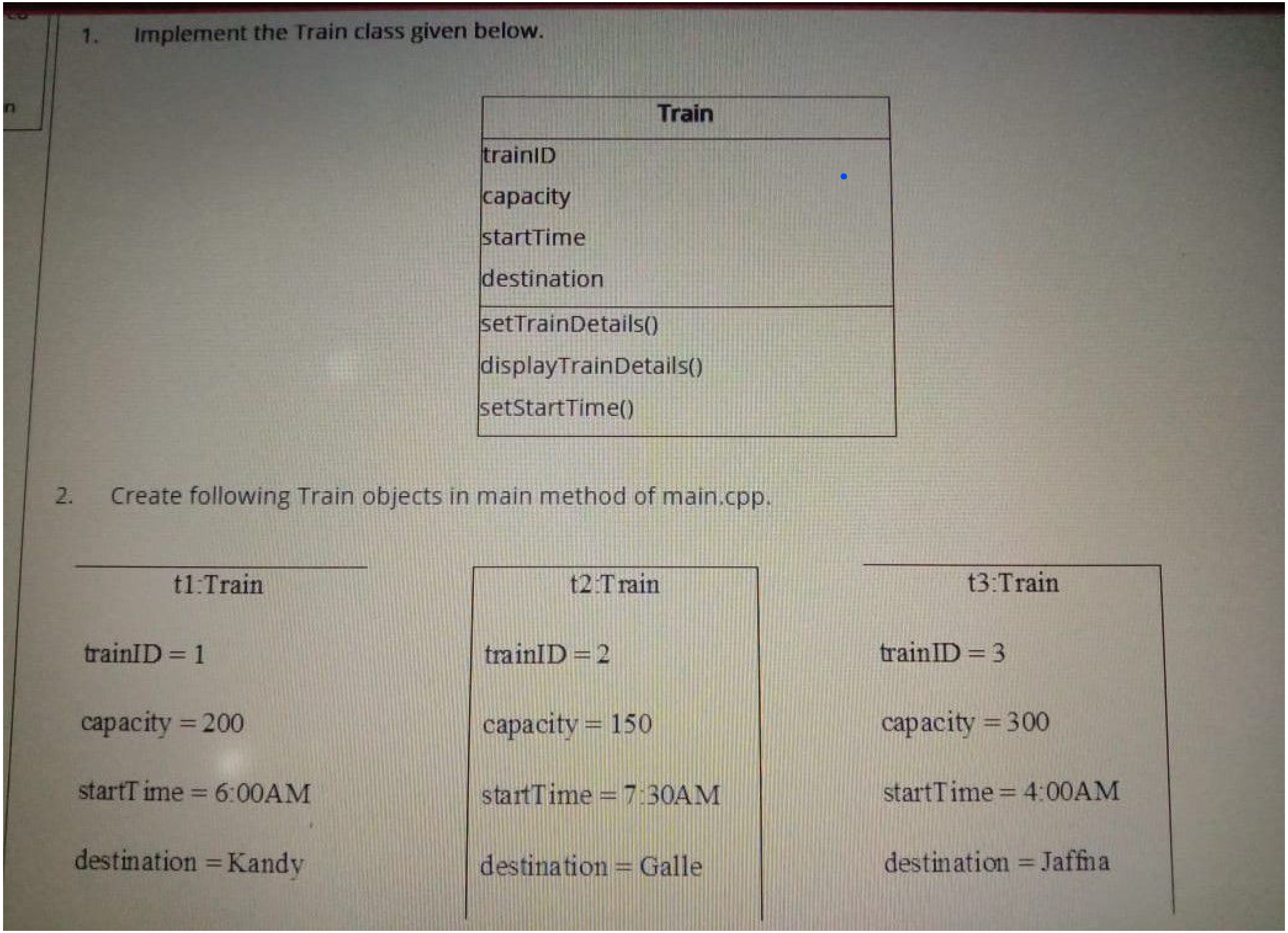


```
StudentID = 1234
StudentName = Kylie
MarksOOC = 75
MarksSPM = 80
MarksISDM = 60
Total Marks = 215
Average Mark = 71.67

StudentID = 4321
StudentName = James
Marks OOC = 65
Marks SPM = 70
Marks ISDM = 85
Total Marks = 220
Average Mark = 73.33

StudentID = 6543
StudentName = Kyson
MarksOOC = 90
Marks SPM = 85
Marks ISDM = 80
Total Marks = 255
Average Mark = 85
```

3. In the main program, get new start time for all trains as keyboard inputs, and set the new start times. Display updated train details.

Sample output:

```
Input new start time of train 1 : 6:30AM
Input new start time of train 2 : 8:00AM
Input new start time of train 3 : 4:30AM

TrainID = 1
Capacity = 200
StartTime = 6:30AM
Destination = Kandy

TrainID = 2
Capacity = 150
StartTime = 8:00AM
Destination = Galle

TrainID = 3
Capacity = 300
Start Time = 4:30AM
Destination = Jaffna
```



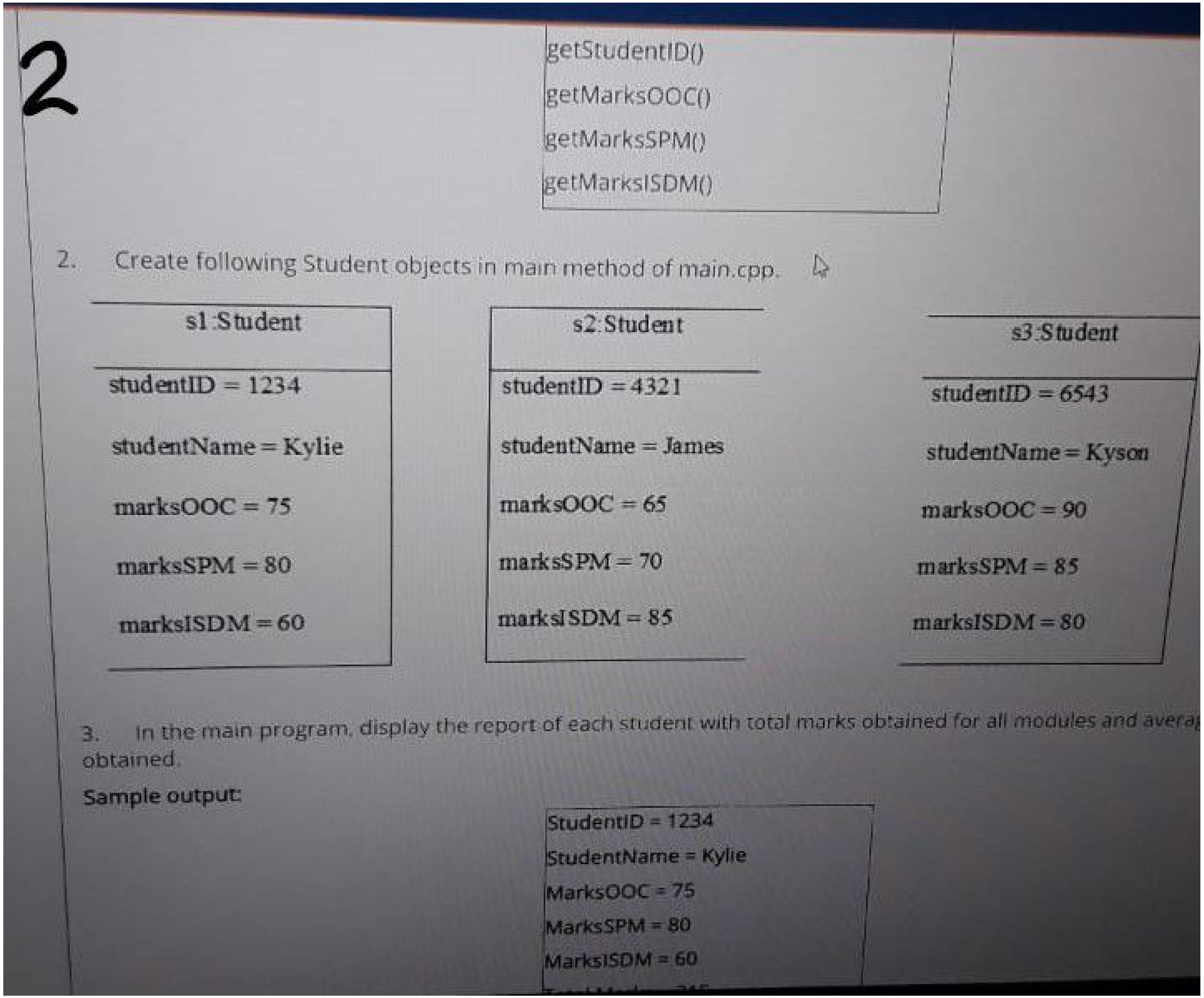
1. Implement the Train class given below.

Train
trainID capacity startTime destination

setTrainDetails()
displayTrainDetails()
setStartTime()

2. Create following Train objects in main method of main.cpp.

t1:Train	t2:Train	t3:Train
trainID = 1	trainID = 2	trainID = 3
capacity = 200	capacity = 150	capacity = 300
startTime = 6:00AM	startTime = 7:30AM	startTime = 4:00AM
destination = Kandy	destination = Galle	destination = Jaffna



2

```
getStudentID()
getMarksOOC()
getMarksSPM()
getMarksISDM()
```

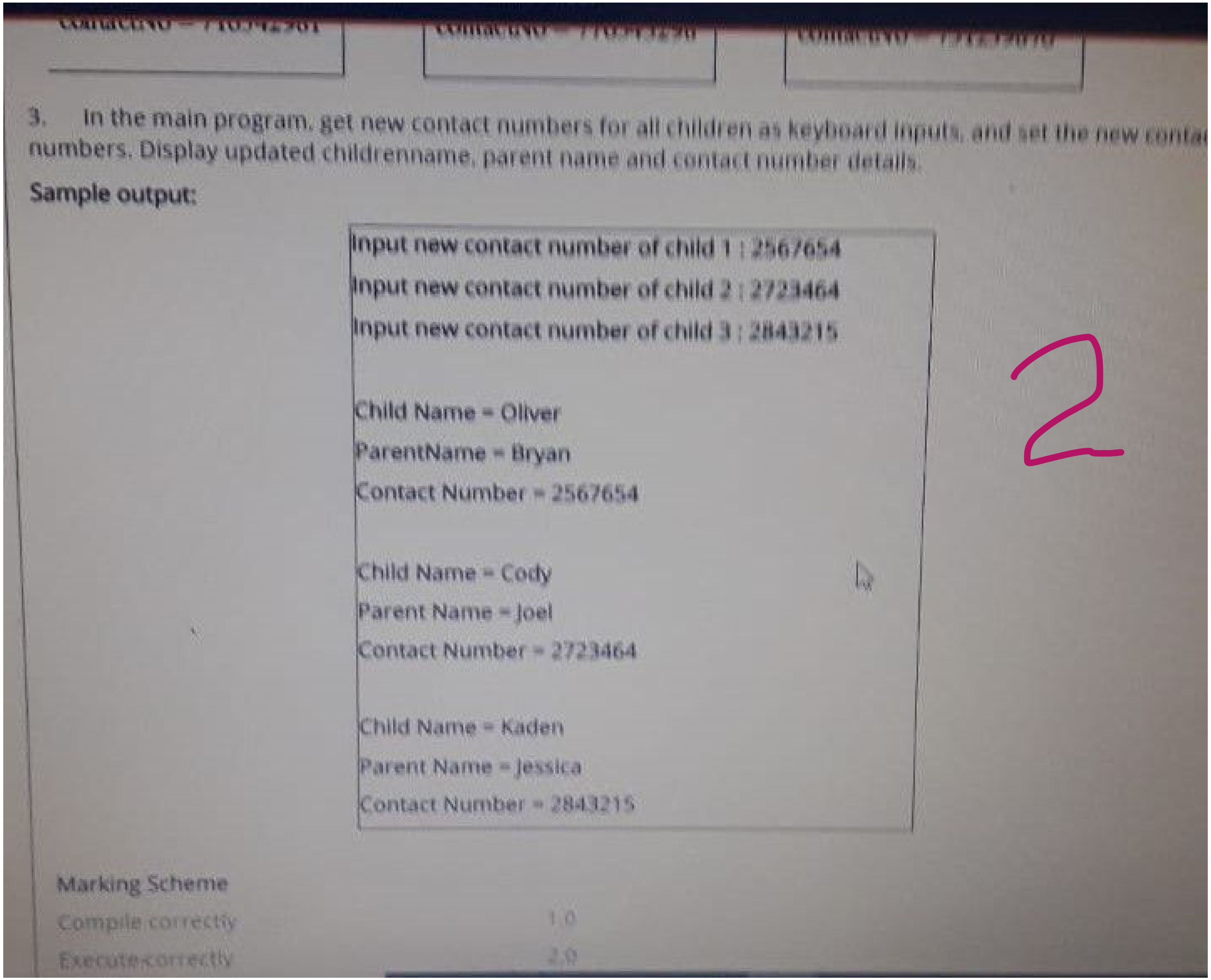
2. Create following Student objects in main method of main.cpp.

s1:Student	s2:Student	s3:Student
studentID = 1234	studentID = 4321	studentID = 6543
studentName = Kylie	studentName = James	studentName = Kyson
marksOOC = 75	marksOOC = 65	marksOOC = 90
marksSPM = 80	marksSPM = 70	marksSPM = 85
marksISDM = 60	marksISDM = 85	marksISDM = 80

3. In the main program, display the report of each student with total marks obtained for all modules and avera
obtained.

Sample output:

```
StudentID = 1234
StudentName = Kylie
MarksOOC = 75
MarksSPM = 80
MarksISDM = 60
```



3. In the main program, get new contact numbers for all children as keyboard inputs, and set the new conta numbers. Display updated childrenname, parent name and contact number details.

Sample output:

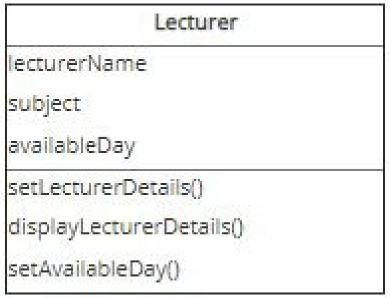
```
Input new contact number of child 1 : 2567654
Input new contact number of child 2 : 2723464
Input new contact number of child 3 : 2843215

Child Name = Oliver
ParentName = Bryan
Contact Number = 2567654

Child Name = Cody
Parent Name = Joel
Contact Number = 2723464

Child Name = Kaden
Parent Name = Jessica
Contact Number = 2843215
```

2

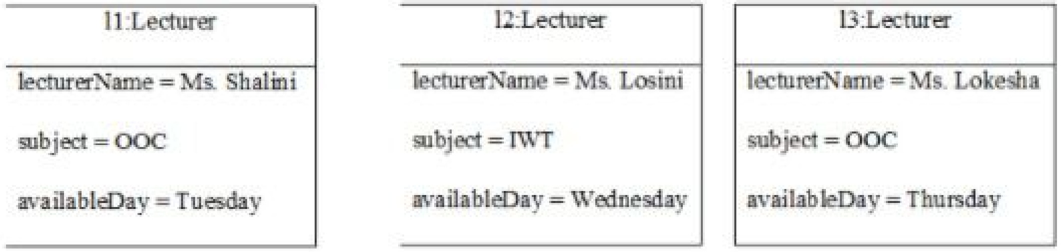
Marking Scheme

Compile correctly	1.0
Execute correctly	2.0

1. Implement the Lecturer class given below.



2. Create following lecturer objects in main method of main.cpp.



3. In the main program, get new available day for all lecturers as keyboard inputs, and set the new available days. Display updated lecturer details.

Sample output:

```
Input available day of Ms. Shalini : Friday
Input available day of Ms. Losini : Monday
Input available day of Ms. Lokesha: Tuesday

LecturerName = Ms. Shalini
Subject = OOC
AvailableDay = Friday

LecturerName = Ms. Losini
Subject = IWT
AvailableDay = Monday


LecturerName = Ms. Lokesha
Subject = OOC
AvailableDay = Tuesday
```

Version T

1. Implement the Medicine class given below.

Medicine
medicineID medicineName sickness dose

setMedicineDetails()
displayMedicineDetails()
setDose()

2. Create following Medicine objects in main method of main.cpp.

m1:Medicine	m2:Medicine	m3:Medicine
medicineID = 1	medicineID = 2	medicineID = 3

re to search

3. In the main program, get new event locations for all events as keyboard inputs, and set the new locations. Display updated event details.

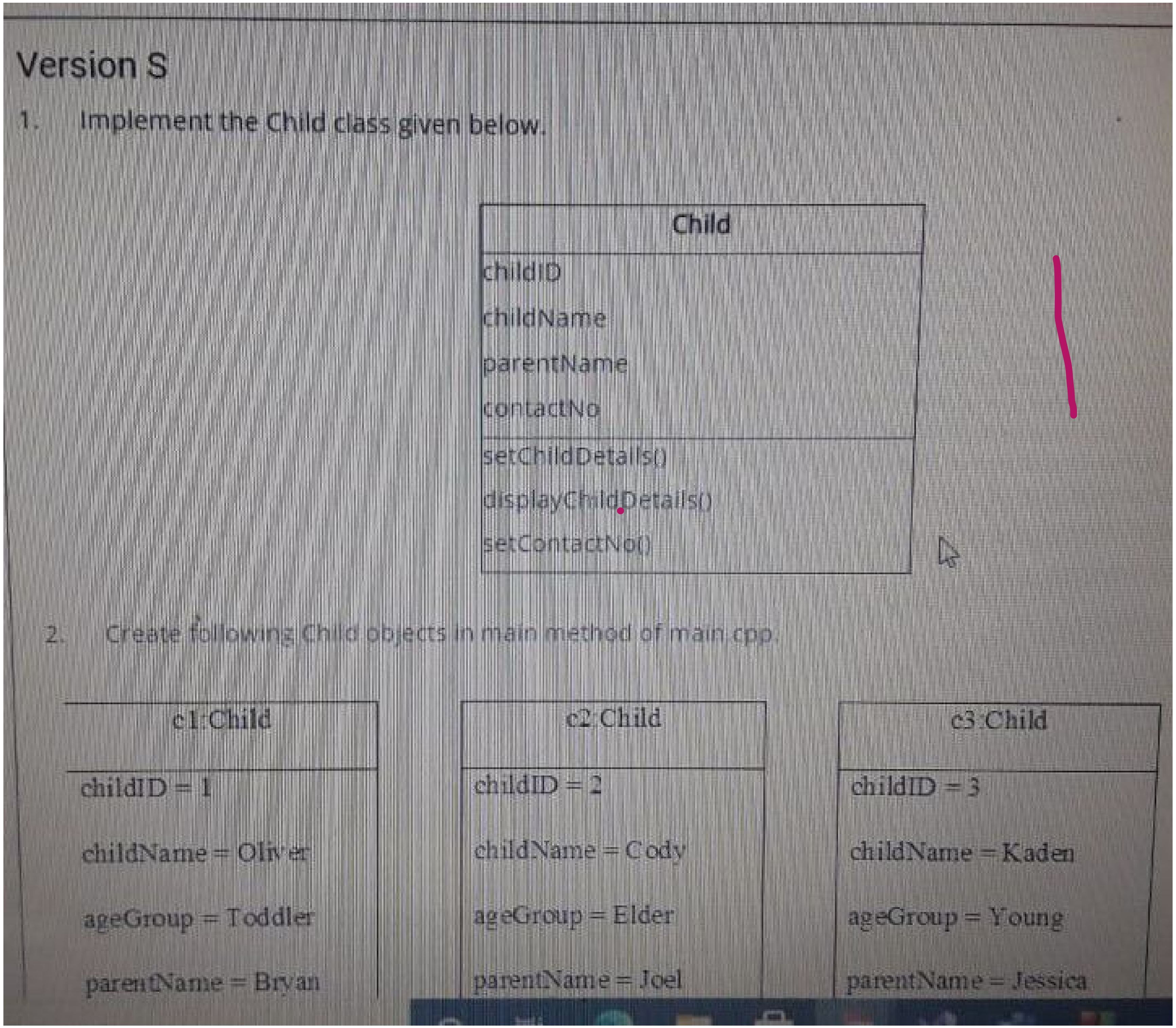
2

Sample output:

```
Input new location of event 1: Malabe
Input new location of event 2: Kelaniya
Input new location of event 3: Galle

EventType = party
ThemeColor = red
Location = Malabe

EventType = wedding
ThemeColor = purple
Location = Kelaniya

EventType = party
ThemeColor = pink
Location = Galle
```

Marking Scheme

Compile correctly	1.0
Execute correctly	2.0
Declaring the class definition correctly	4.0
Implementing the class methods correctly	7.0
In client program	
Creating objects correctly	3.0
Calling methods correctly	3.0



Version S

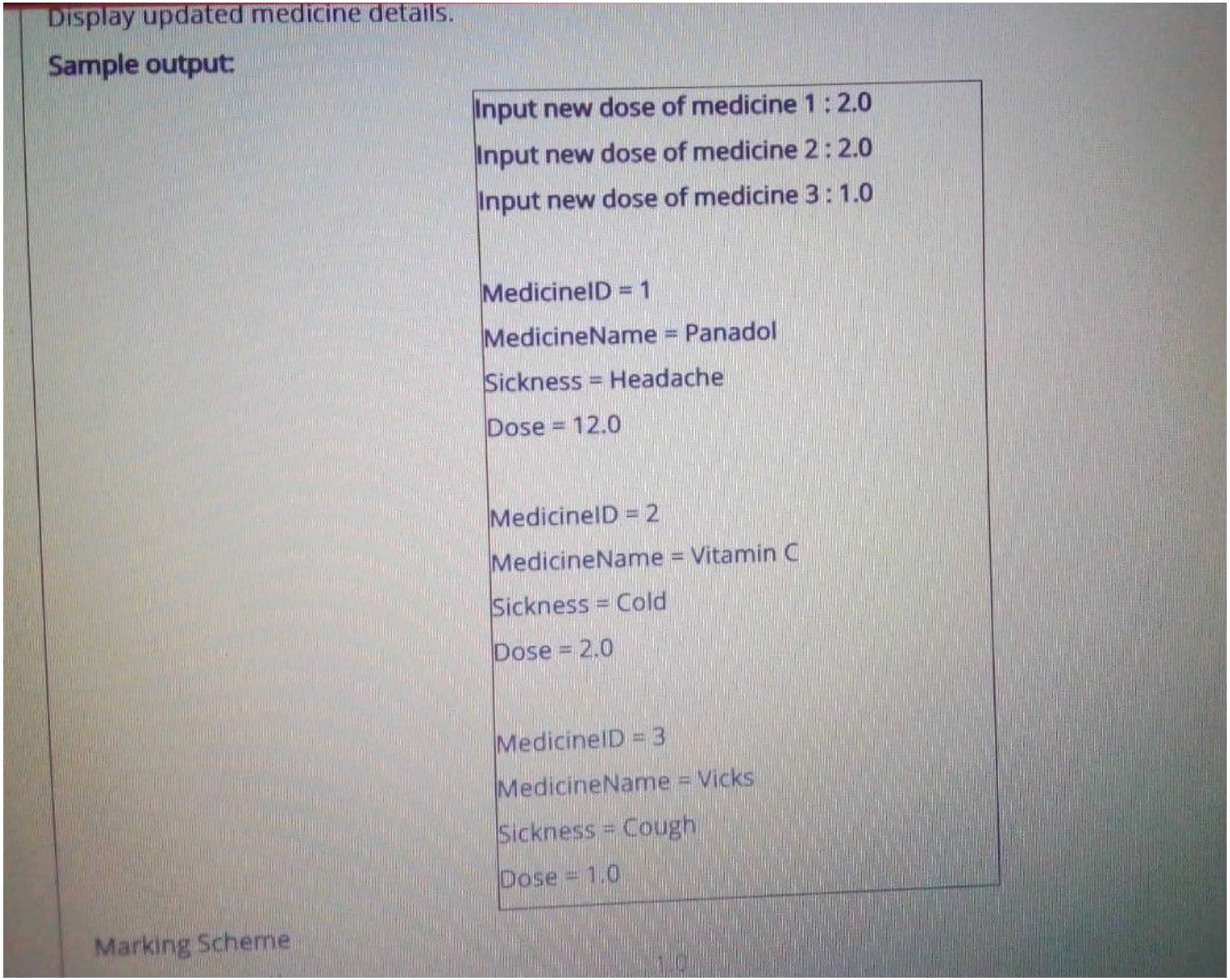
1. Implement the Child class given below.

Child
childID childName parentName contactNo

setChildDetails()
displayChildDetails()
setContactNo()

2. Create following Child objects in main method of main.cpp.

c1:Child	c2:Child	c3:Child
childID = 1	childID = 2	childID = 3
childName = Oliver	childName = Cody	childName = Kaden
ageGroup = Toddler	ageGroup = Elder	ageGroup = Young
parentName = Bryan	parentName = Joel	parentName = Jessica

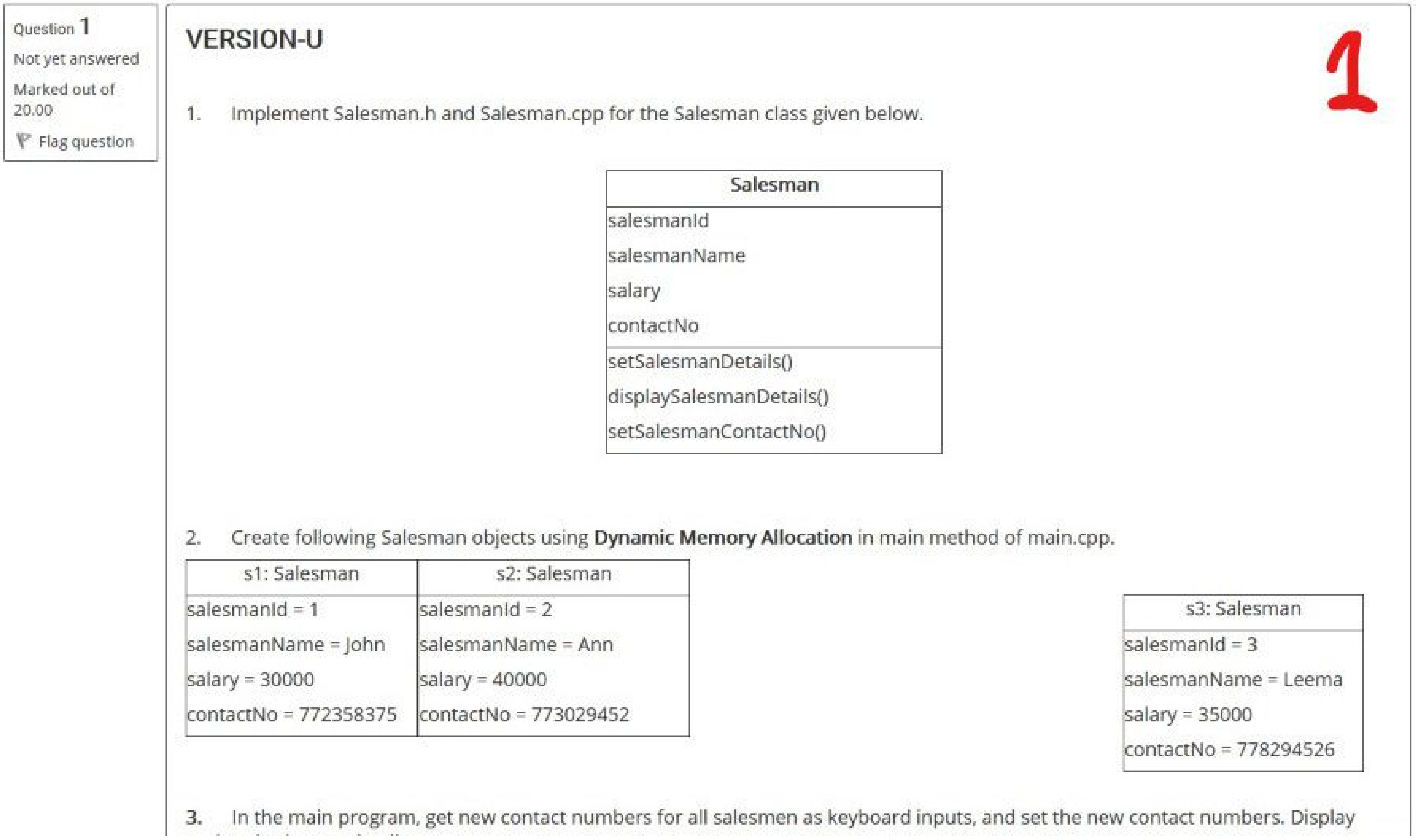


Display updated medicine details.

Sample output:

```
Input new dose of medicine 1 : 2.0
Input new dose of medicine 2 : 2.0
Input new dose of medicine 3 : 1.0

MedicineID = 1
MedicineName = Panadol
Sickness = Headache
Dose = 12.0

MedicineID = 2
MedicineName = Vitamin C
Sickness = Cold
Dose = 2.0

MedicineID = 3
MedicineName = Vicks
Sickness = Cough
Dose = 1.0
```

Marking Scheme



Question **1**

Not yet answered

Marked out of 20.00

Flag question

VERSION-U

1

1. Implement Salesman.h and Salesman.cpp for the Salesman class given below.

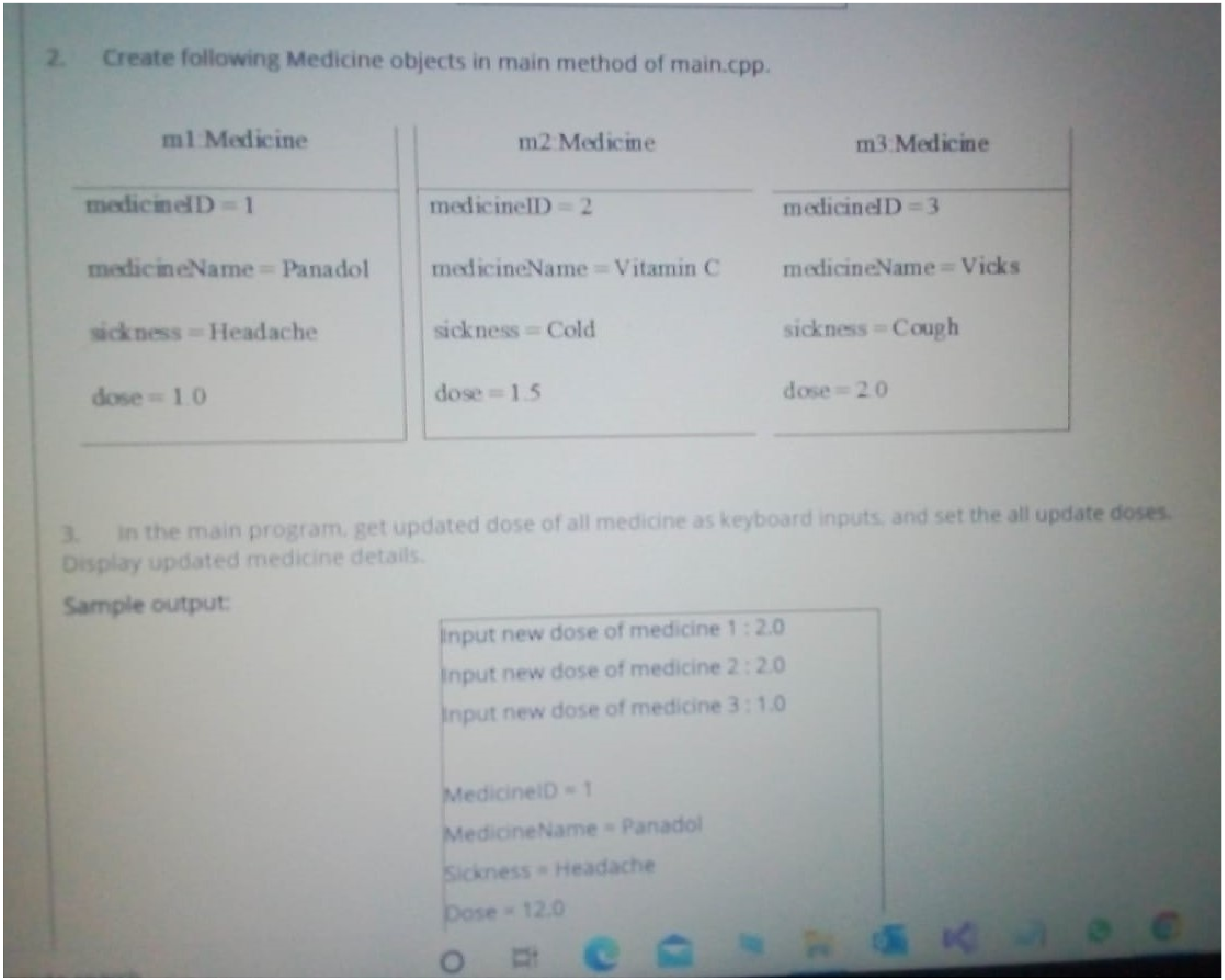
Salesman
salesmanId
salesmanName
salary
contactNo
setSalesmanDetails()
displaySalesmanDetails()
setSalesmanContactNo()

2. Create following Salesman objects using **Dynamic Memory Allocation** in main method of main.cpp.

s1: Salesman	s2: Salesman
salesmanId = 1	salesmanId = 2
salesmanName = John	salesmanName = Ann
salary = 30000	salary = 40000
contactNo = 772358375	contactNo = 773029452

s3: Salesman
salesmanId = 3
salesmanName = Leema
salary = 35000
contactNo = 778294526

3. In the main program, get new contact numbers for all salesmen as keyboard inputs, and set the new contact numbers. Display



2. Create following Medicine objects in main method of main.cpp.

m1:Medicine	m2:Medicine	m3:Medicine
medicineID = 1	medicineID = 2	medicineID = 3
medicineName = Panadol	medicineName = Vitamin C	medicineName = Vicks
sickness = Headache	sickness = Cold	sickness = Cough
dose = 1.0	dose = 1.5	dose = 2.0

3. In the main program, get updated dose of all medicine as keyboard inputs, and set the all update doses. Display updated medicine details.

Sample output:

```
Input new dose of medicine 1 : 2.0
Input new dose of medicine 2 : 2.0
Input new dose of medicine 3 : 1.0


MedicineID = 1
MedicineName = Panadol
Sickness = Headache
Dose = 12.0
```

3. In the main program, get new contact numbers for all salesmen as keyboard inputs, and set the new contact numbers. Display updated salesmandetails.

2

Sample output:

```
Input new contact number of salesman 1 : 772461836
Input new contact number of salesman 2 : 773927452
Input new contact number of salesman 3 : 772037452

SalesmanId = 1
SalesmanName = John
Salary = 30000
ContactNo = 772461836

SalesmanId = 2
SalesmanName = Ann
Salary = 40000
ContactNo = 773927452

SalesmanId = 3
SalesmanName = Leema
Salary = 35000
ContactNo = 772037452
```

VERSION-V

1

1. Implement the Event class given below.

Event
eventId
eventType
themeColor
location
setEventDetails()
displayEventDetails()
setEventLocation()

2. Create following three Event objects using **Dynamic Memory Allocation** in main method of main.cpp.

e1:Event	e2:Event
eventId = 1	eventId = 2
eventType = party	eventType = wedding
themeColor = red	themeColor = purple
location = Nugegoda	location = Maharagama

e3:Event
eventId = 3
eventType = party
themeColor = pink
location = Malabe

3. In the main program, get new event locations for all events as keyboard inputs, and set the new locations. Display updated event details.

Sample output: