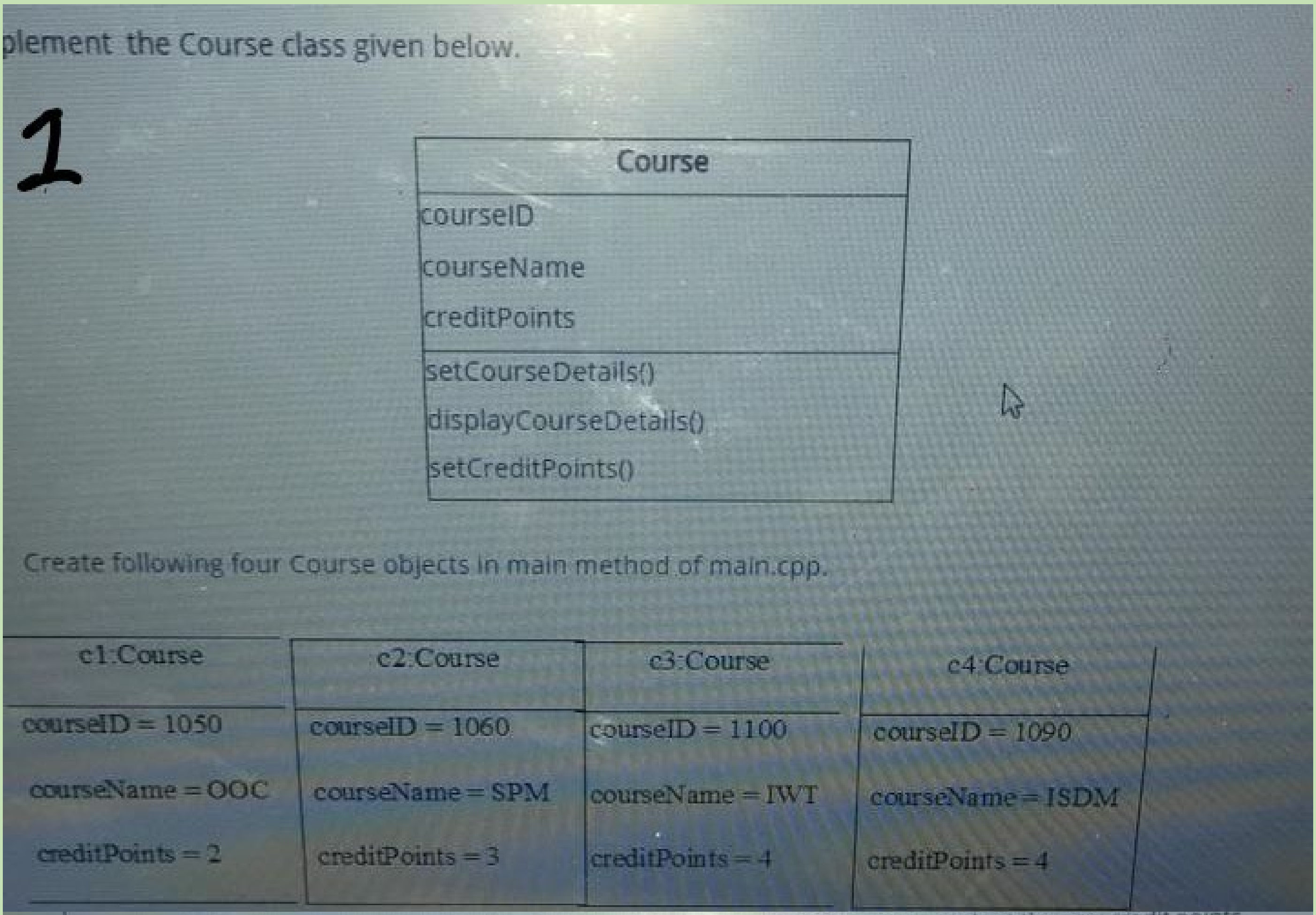
Course



plement the Course class given below.

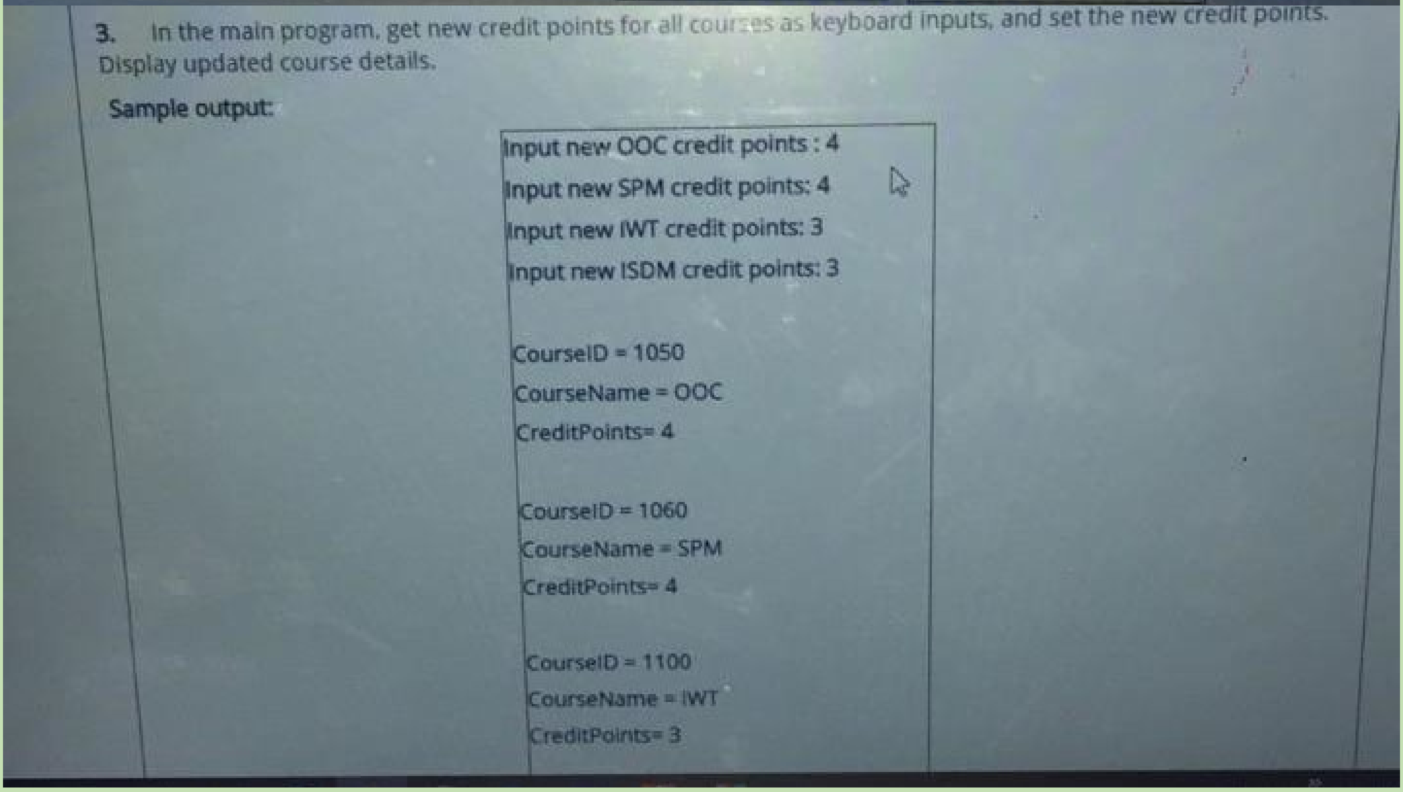
1

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



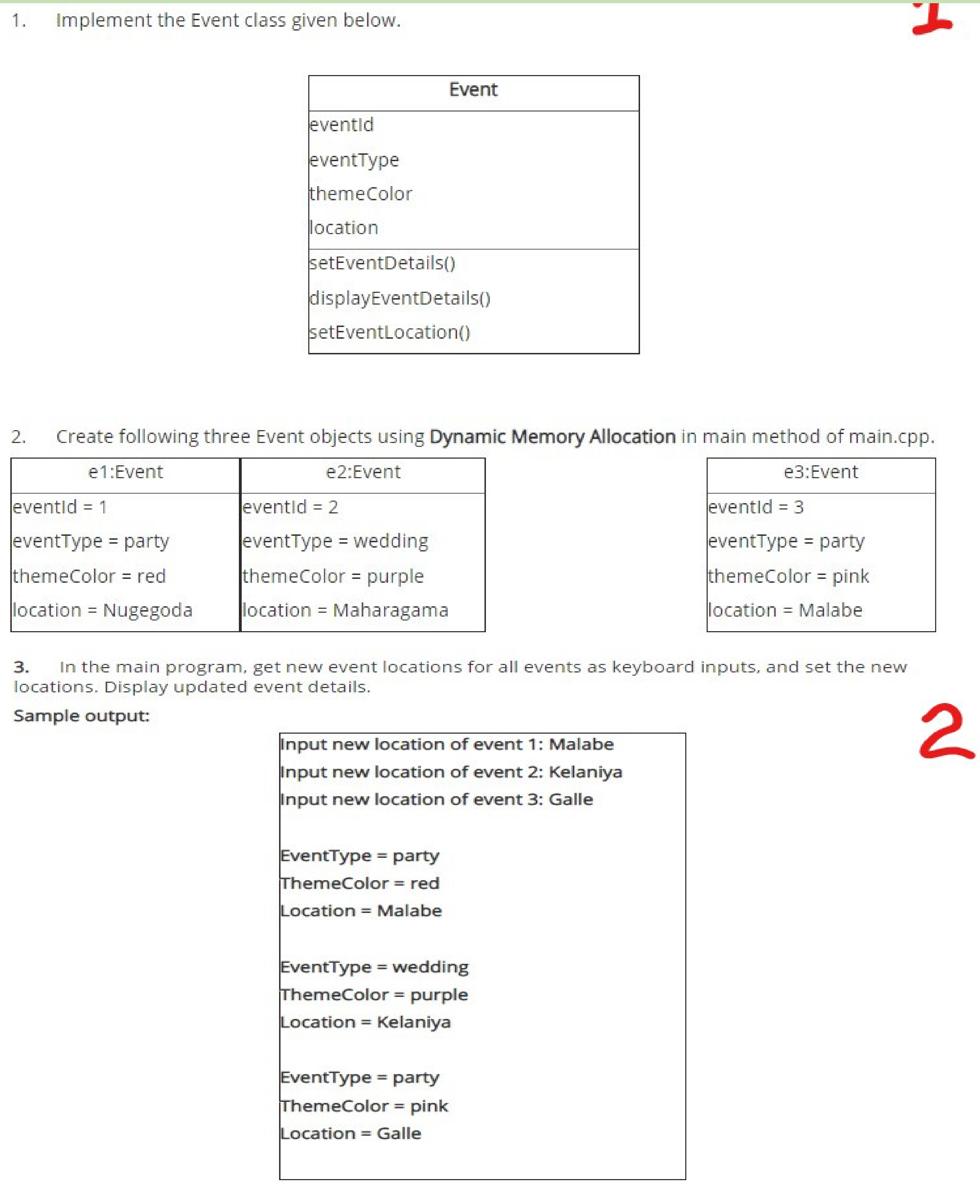
3. In the main program, get new credit points for all courses as keyboard inputs, and set the new credit points. Display updated course details.

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
```

```
#include <iostream>

using namespace std;

class Course{
	private:
		int courseID;
		string courseName;
		int creditPoints;
	public:
		void setCourseDetails(int cid, string cname, int cpt);
		void displayCourseDetails();
		void setCreditPoints();
};

void Course::setCourseDetails(int cid, string cname, int cpt)
{
	courseID = cid;
	courseName = cname;
	creditPoints = cpt;

}

void Course::displayCourseDetails()
{
	cout << "Course ID = " << courseID << endl
		 << "CourseName = " << courseName << endl
		 << "CreditPoint = " << creditPoints << endl<<endl;
}

void Course::setCreditPoints()
{
	cout << "Input new "<< courseName <<" credit points : ";
	cin >> creditPoints;
}

int main ()
{
	Course c1,c2,c3,c4;

	c1.setCourseDetails(1050,"OOC",2);
	c2.setCourseDetails(1060,"SPM",3);
	c3.setCourseDetails(1100,"IWT",4);
	c4.setCourseDetails(1090,"ISDM",4);

	c1.setCreditPoints();
	c2.setCreditPoints();
	c3.setCreditPoints();
	c4.setCreditPoints();

	cout<<endl;

	c1.displayCourseDetails();
	c2.displayCourseDetails();
	c3.displayCourseDetails();
	c4.displayCourseDetails();


	return 0;
}
```

Event



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

eventId = 1
eventType = party
themeColor = red
location = Nugegoda

e2:Event

eventId = 2
eventType = wedding
themeColor = purple
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:

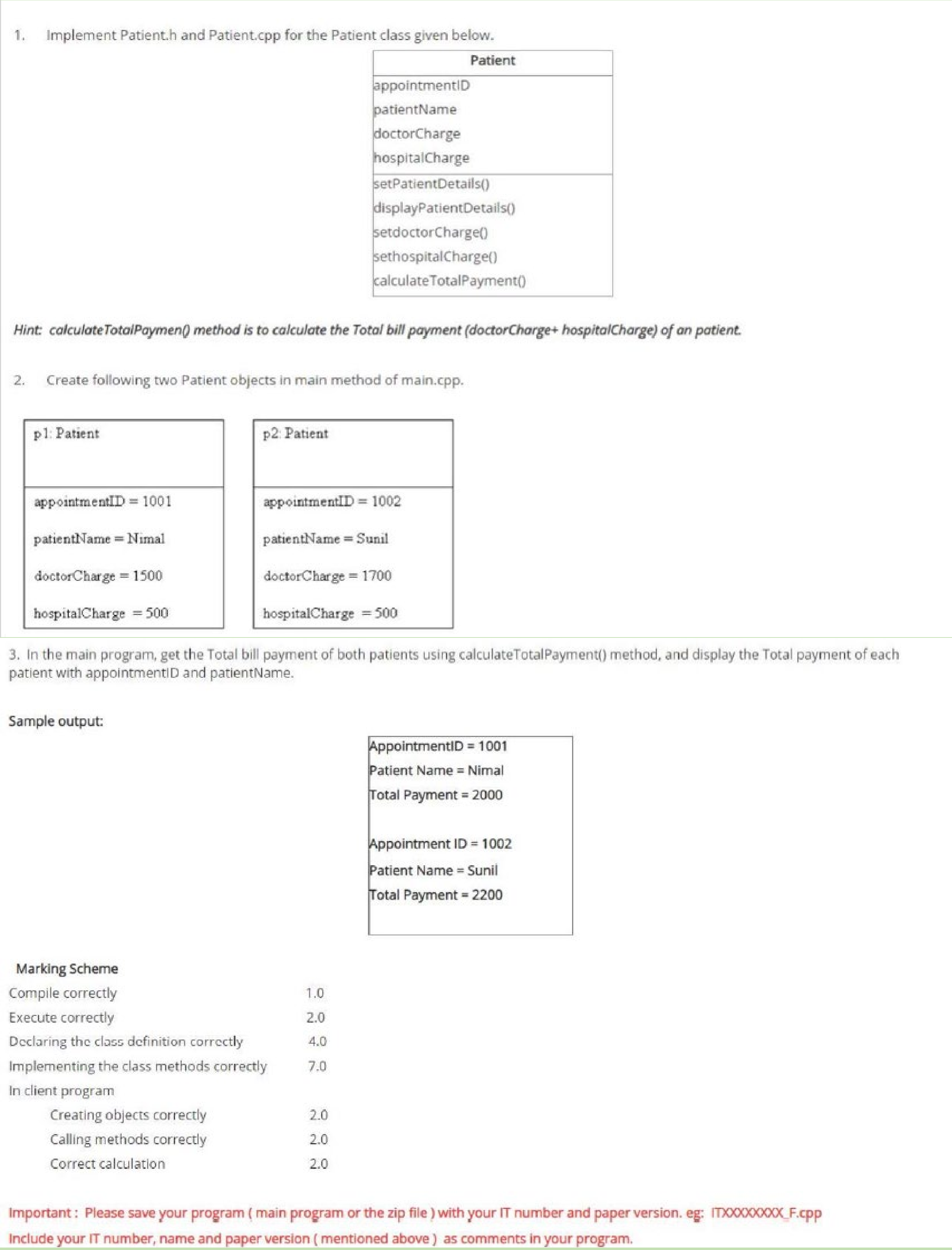
2

```
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
```

```
#include <iostream>

using namespace std;

class Event{
        private:
                int eventId;
                string eventType;
                string themeColor;
                string location;
        public:
                void setEventDetails(int eid, string etyp, string tcolr, string eloc);
                void displayEventDetails();
                void setEventLocation();
};

void Event::setEventDetails(int eid, string etyp, string tcolr, string eloc)
{
        eventId = eid;
        eventType = etyp;
        themeColor = tcolr;
        location = eloc;
}

void Event::displayEventDetails()
{
        cout << "EventType = " <<eventType<<endl
                 << "ThemeColor = " <<themeColor<<endl
                 << "Location = " <<location << endl <<endl;
}

void Event::setEventLocation()
```

```
{
	cout << "Input new location of event " << eventId <<" : ";
	cin >> location;
}

int main ()
{
	Event *e1 = new Event();
	Event *e2 = new Event();
	Event *e3 = new Event();

	e1->setEventDetails(1,"party","red","Nugegoda");
	e2->setEventDetails(2,"wedding","purple","Maharagama");
	e3->setEventDetails(3,"party","pink","Malabe");

	e1->setEventLocation();
	e2->setEventLocation();
	e3->setEventLocation();
	cout<<endl;

	e1->displayEventDetails();
	e2->displayEventDetails();
	e3->displayEventDetails();

	delete e1;
	delete e2;
	delete e3;

	return 0;
}
```

Patient



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

Patient
appointmentID patientName doctorCharge hospitalCharge

setPatientDetails()
displayPatientDetails()
setdoctorCharge()
sethospitalCharge()
calculateTotalPayment()

Hint: calculateTotalPaymen() method is to calculate the Total bill payment (doctorCharge+ hospitalCharge) of an patient.

2. Create following two Patient objects in main method of main.cpp.

p1: Patient

appointmentID = 1001
patientName = Nimal
doctorCharge = 1500
hospitalCharge = 500

p2: Patient

appointmentID = 1002
patientName = Sunil
doctorCharge = 1700
hospitalCharge = 500

3. In the main program, get the Total bill payment of both patients using calculateTotalPayment() method, and display the Total payment of each patient with appointmentID and patientName.

Sample output:

```
AppointmentID = 1001
Patient Name = Nimal
Total Payment = 2000

Appointment ID = 1002
Patient Name = Sunil
Total Payment = 2200
```

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

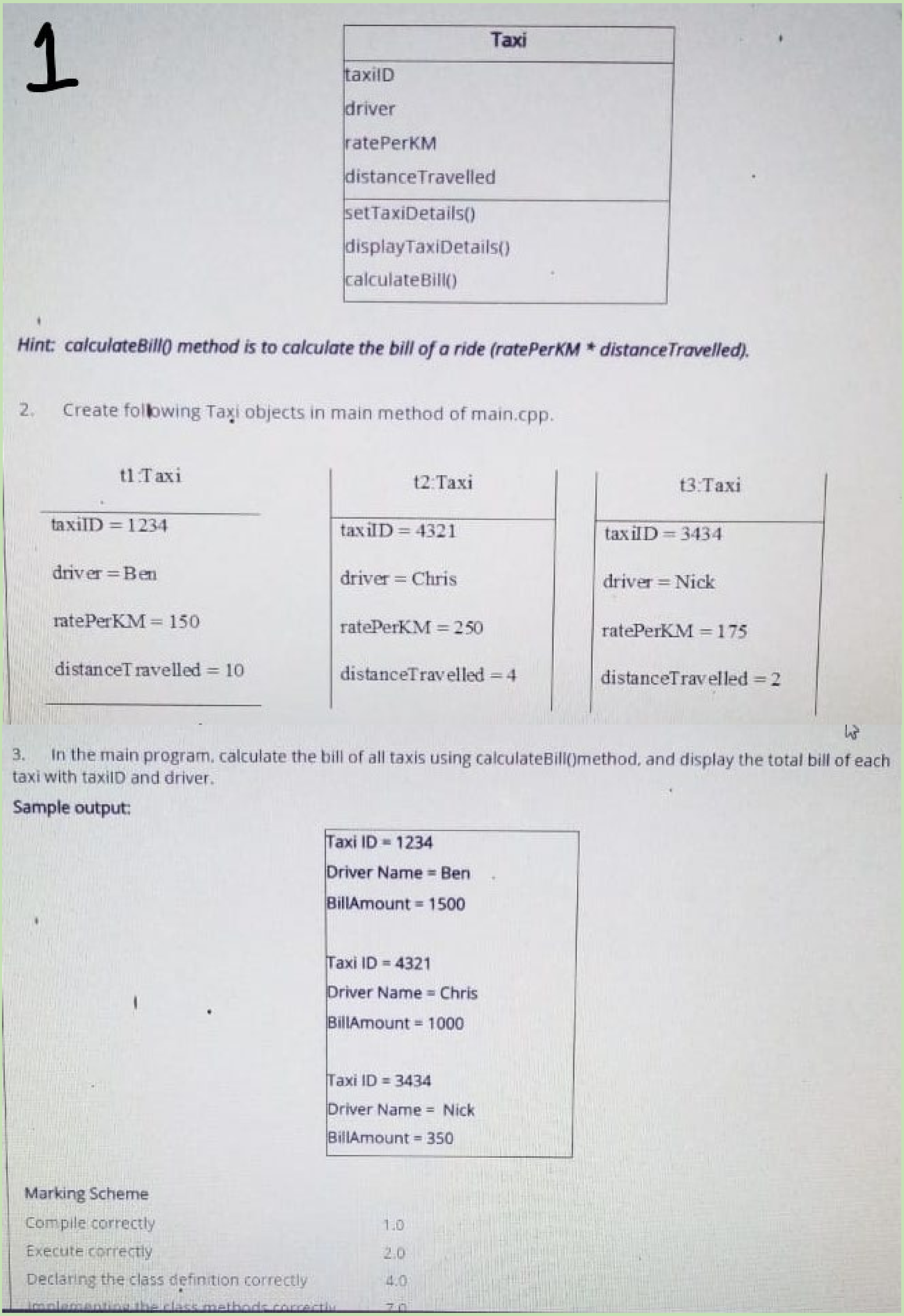
Important : Please save your program (main program or the zip file) with your IT number and paper version. eg: ITXXXXXXXX_F.cpp

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

```
#include <iostream>

using namespace std;

class Patient {
    private:
        int appoinmentID;
        string patientName;
        float doctorCharge;
        float hospitalCharge;
    public:
      void setpatientDetaile (int pId, string pname);
        void displayPatientDetails ();
        void setDoctorCharge (float dCharge);
        void setHospitalCharge (float hcharge);
        float calculateTolatPayment ();
};


void Patient::setpatientDetaile(int pID, string pname)
{
    appoinmentID = pID;
    patientName = pname;
}

void Patient::setDoctorCharge(float dCharge)
{
    doctorCharge = dCharge;
}

void Patient::setHospitalCharge(float hcharge)
{
    hospitalCharge = hcharge;
}
```

```
void Patient::displayPatientDetails()

{

    cout << "Appoinment ID = " << appoinmentID << endl

         << "Patient Name  = " << patientName  << endl;


}


float Patient::calculateTolatPayment()

{

    return doctorCharge + hospitalCharge;

}



int main ()

{

    Patient p1,p2;


    p1.setpatientDetaile(1001,"Nimal");

    p1.setDoctorCharge(1500);

    p1.setHospitalCharge(500);


    p2.setpatientDetaile(1002,"Sunil");

    p2.setDoctorCharge(1700);

    p2.setHospitalCharge(500);


    p1.displayPatientDetails();

    cout << "Total Payment = "  << p1.calculateTolatPayment() << endl << endl;


    p2.displayPatientDetails();

    cout << "Total Payment = "  << p2.calculateTolatPayment() << endl;


    return 0;

}
```

Taxi



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

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.

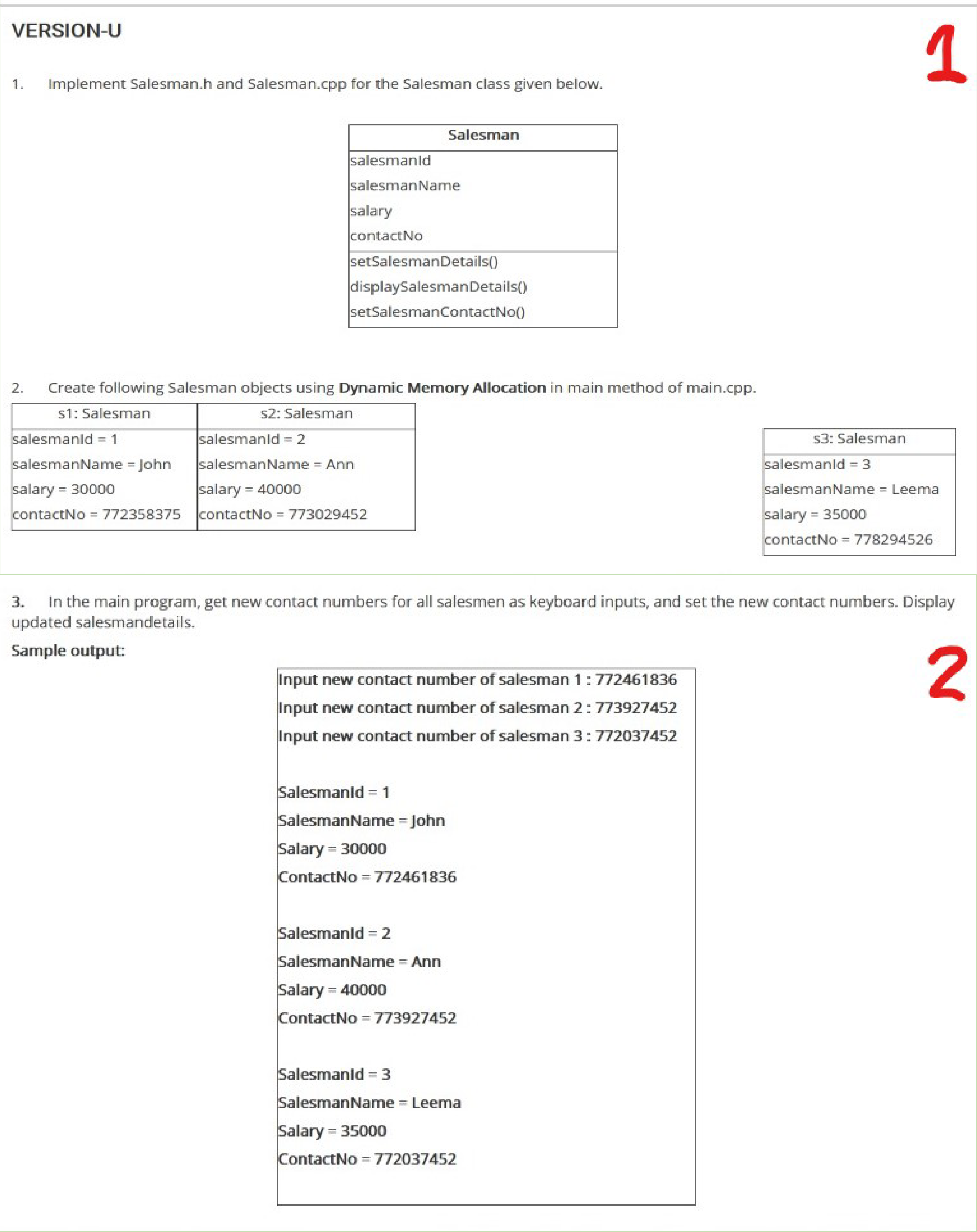
Sample output:

```
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

```
#include <iostream>

#include <iostream>


using namespace std;


class Taxi{

    private:

        int taxiID;

        string driver;

        int ratePerKM;

        int distanceTravelled;

    public:

        void setTaxiDetails(int id, string name, int rate, int dist);

        void displayTaxiDetails();

        float calculateBill();


};


void Taxi::setTaxiDetails(int id, string name, int rate, int dist)

{

    taxiID = id;

    driver = name;

    ratePerKM = rate;

    distanceTravelled = dist;

}


void Taxi::displayTaxiDetails()

{

    cout <<"Taxi ID = " << taxiID << endl

         <<"Driver Name = " << driver << endl;

}


float Taxi::calculateBill()

{
    return ratePerKM*distanceTravelled;
}


int main ()
{
    Taxi t1,t2,t3;


    t1.setTaxiDetails(1234,"Ben",150,10);
    t2.setTaxiDetails(4321,"Chris",250,4);
    t3.setTaxiDetails(3434,"Nick",175,2);


    t1.displayTaxiDetails();
    cout <<"BillAmount = " << t1.calculateBill() <<endl<<endl;
    t2.displayTaxiDetails();
    cout <<"BillAmount = " << t2.calculateBill() <<endl<<endl;
    t3.displayTaxiDetails();
    cout <<"BillAmount = " << t3.calculateBill() <<endl;


    return 0;
}
```

Salesman



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 updated salesmandetails.

Sample output:

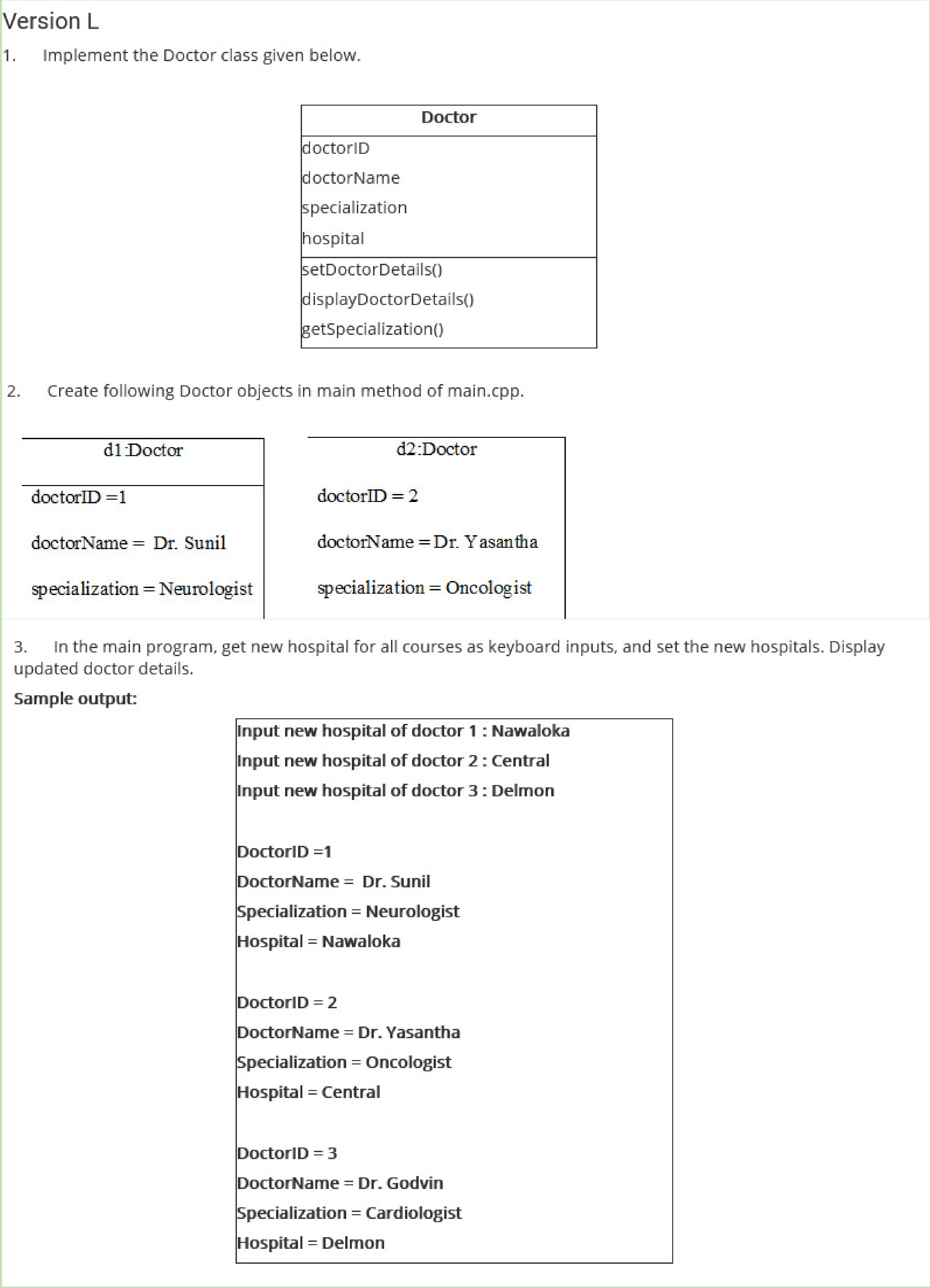
2

```
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
```

```
#include <iostream>

using namespace std;

class Salesman{
        private:
                int salesmanID;
                string salesmanName;
                float salary;
                int contactNo;
        public:
                void setSalesmanDetails(int sid, string name, float sal, int cNo);
                void displaySalesmanDetails();
                void setSalesmanContactNo();
};

void Salesman::setSalesmanDetails(int sid, string name, float sal, int cNo)
{
        salesmanID = sid;
        salesmanName = name;
        salary = sal;
        contactNo = cNo;
}

void Salesman::displaySalesmanDetails()
{
        cout << "SalesmanId = " << salesmanID << endl
                 << "SalesmanName = " << salesmanName << endl
                 << "salary = " << salary << endl
                 << "ContactNo = " << contactNo << endl<<endl;
}

void Salesman::setSalesmanContactNo()
```

```
{
        cout<<"Input new contact number of salesman "<< salesmanID << " : ";
        cin>>contactNo;
}

int main ()
{
        Salesman *s1 = new Salesman();
        Salesman *s2 = new Salesman();
        Salesman *s3 = new Salesman();

        s1->setSalesmanDetails(1,"John",30000,772358375);
        s2->setSalesmanDetails(2,"Ann",40000,773029452);
        s3->setSalesmanDetails(3,"Leema",35000,778294526);

        s1->setSalesmanContactNo();
        s2->setSalesmanContactNo();
        s3->setSalesmanContactNo();

        cout<<endl;

        s1->displaySalesmanDetails();
        s2->displaySalesmanDetails();
        s3->displaySalesmanDetails();

        delete s1,s2,s3;

        return 0;
}
```

Doctor



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

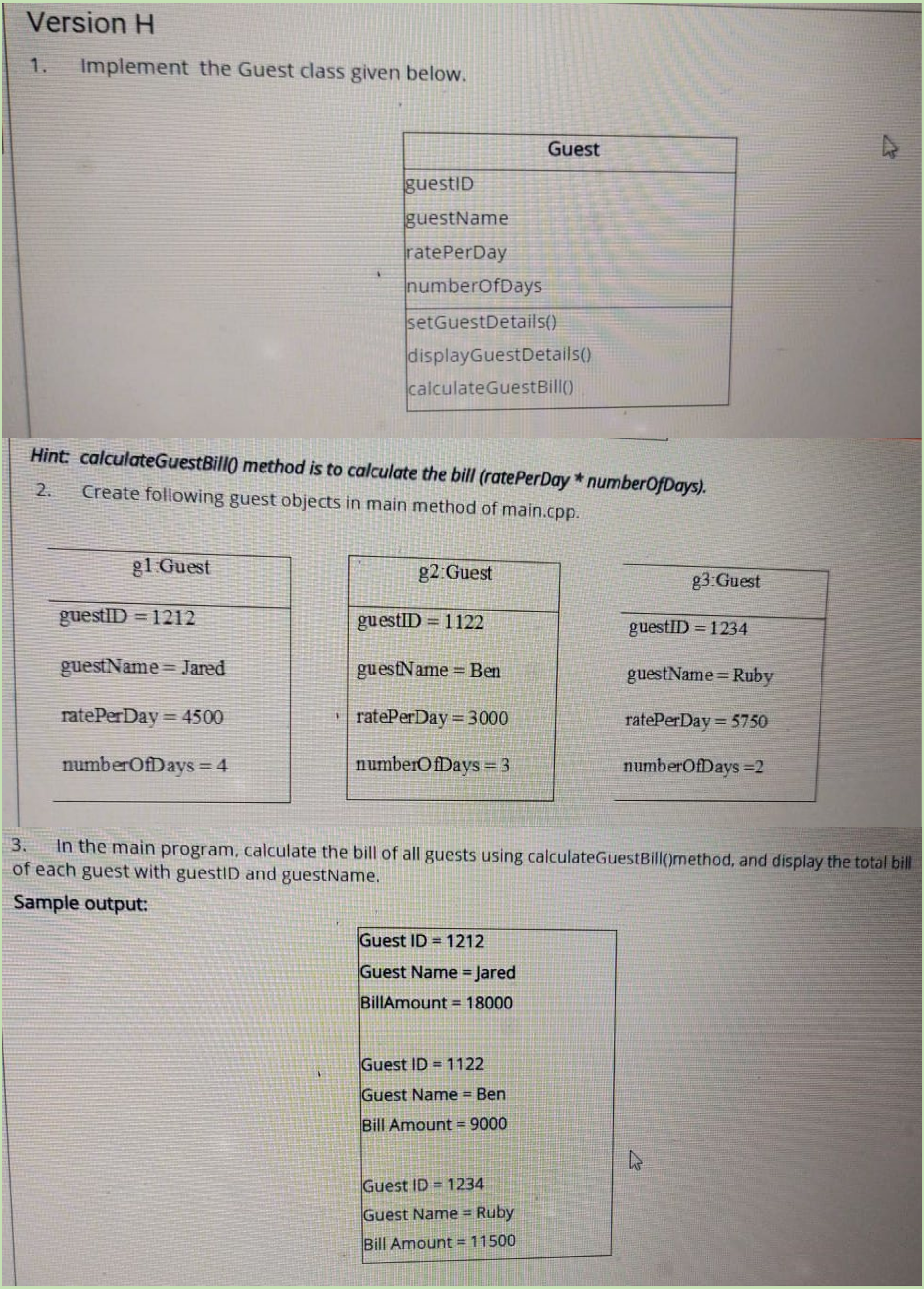
3. In the main program, get new hospital for all courses as keyboard inputs, and set the new hospitals. Display updated doctor details.

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
```

```
#include <iostream>

using namespace std;

//Class Definition
class Doctor {
  private:
    int doctorID;
    string doctorName;
    string specialization;
    string hospital;

  public:
    void setDoctorDetails(int dID, string dName, string dSpec, string dHospital);
    void displayDoctorDetails();
    string getSpecialization();
    void setHospital();
};

//Methods Implementation

void Doctor::setDoctorDetails(int dID, string dName, string dSpec, string dHospital){
    doctorID = dID;
    doctorName = dName;
    specialization = dSpec;
    hospital = dHospital;
}

void Doctor::displayDoctorDetails(){
  cout << "DoctorID = " << doctorID << endl;
  cout << "DoctorName = " << doctorName << endl;
  cout << "Specialization = " << specialization << endl;
  cout << "Hostpital = " << hospital << endl;
  cout << endl;
}

string Doctor::getSpecialization()
{
  return specialization;
}

void Doctor::setHospital()
{
  cout << "Input new hostpital of doctor " << doctorID << " : ";
  cin >> hospital;
}

int main() {
  Doctor d1,d2,d3;
  //Create Objects
  d1.setDoctorDetails(1, "Dr. Sunil", "Neurologist", "Asiri");
  d2.setDoctorDetails(2, "Dr. Yasantha", "Oncologist", "Lanka");
  d3.setDoctorDetails(3, "Mr.Godvin", "Cardiologist", "CCC");

  //Set New hospitals for doctors
  d1.setHospital();
  d2.setHospital();
  d3.setHospital();

  cout << endl;

  //Display Doctor Details
  d1.displayDoctorDetails();
  d2.displayDoctorDetails();
  d3.displayDoctorDetails();

  return 0;
}
```

Guest



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	g2:Guest	g3:Guest
guestID = 1212	guestID = 1122	guestID = 1234
guestName = Jared	guestName = Ben	guestName = Ruby
ratePerDay = 4500	ratePerDay = 3000	ratePerDay = 5750
numberOfDays = 4	numberOfDays = 3	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.

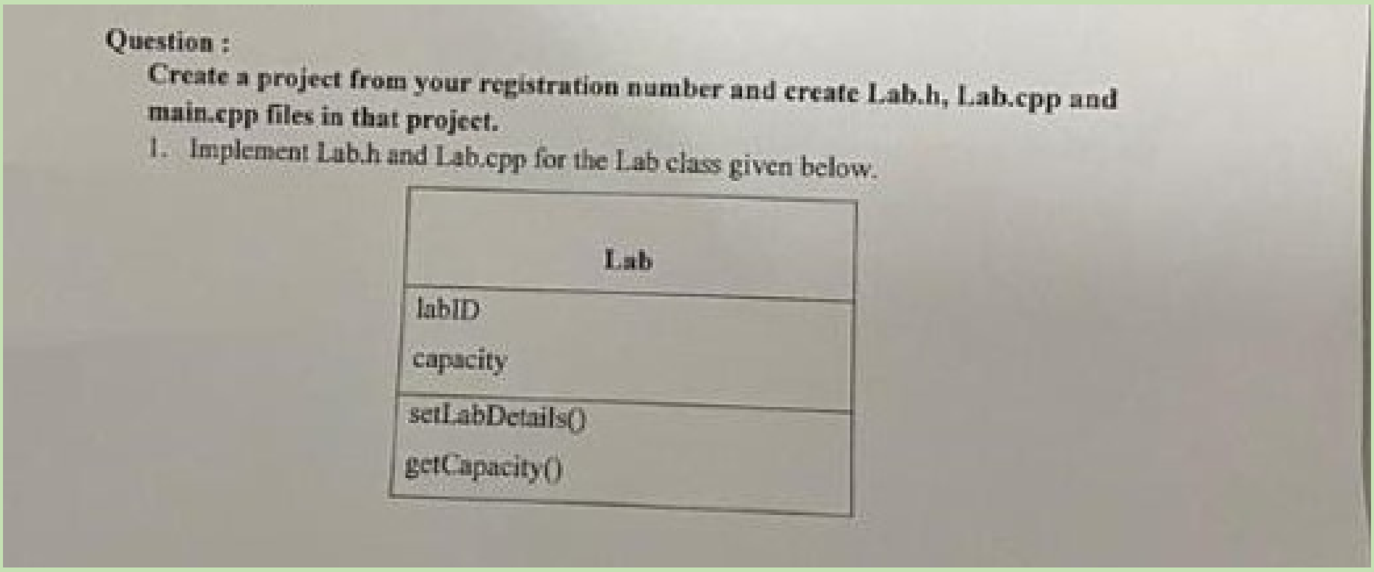
Sample output:

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

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

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

```
#include <iostream>

using namespace std;

class Guest{
        private:
                int guestID;
                string guestName;
                int ratePerDays;
                int numberOfDays;
        public:
                void setGuestDetails(int id, string name , int rate, int days);
                void displayGuestDetails();
                int calculateGuestBill();
};

void Guest::setGuestDetails(int id, string name, int rate, int days)
{
        guestID = id;
        guestName = name;
        ratePerDays = rate;
        numberOfDays = days;
}

void Guest::displayGuestDetails()
{
        cout << "Guest ID = " << guestID << endl
                 << "Guest Name = " << guestName << endl;
}

int Guest::calculateGuestBill()
{
        return ratePerDays*numberOfDays;
}
```

```cpp
int main ()
{
        Guest g1,g2,g3;

        g1.setGuestDetails(1212,"Jared",4500,4);
        g2.setGuestDetails(1122,"Ben",3000,3);
        g3.setGuestDetails(1234,"Ruby",5750,2);

        g1.displayGuestDetails();
        cout<<"BillAmount = "<<g1.calculateGuestBill()<<endl<<endl;
        g2.displayGuestDetails();
        cout<<"BillAmount = "<<g2.calculateGuestBill()<<endl<<endl;
        g3.displayGuestDetails();
        cout<<"BillAmount = "<<g3.calculateGuestBill()<<endl;

        return 0;
}
```

Lab



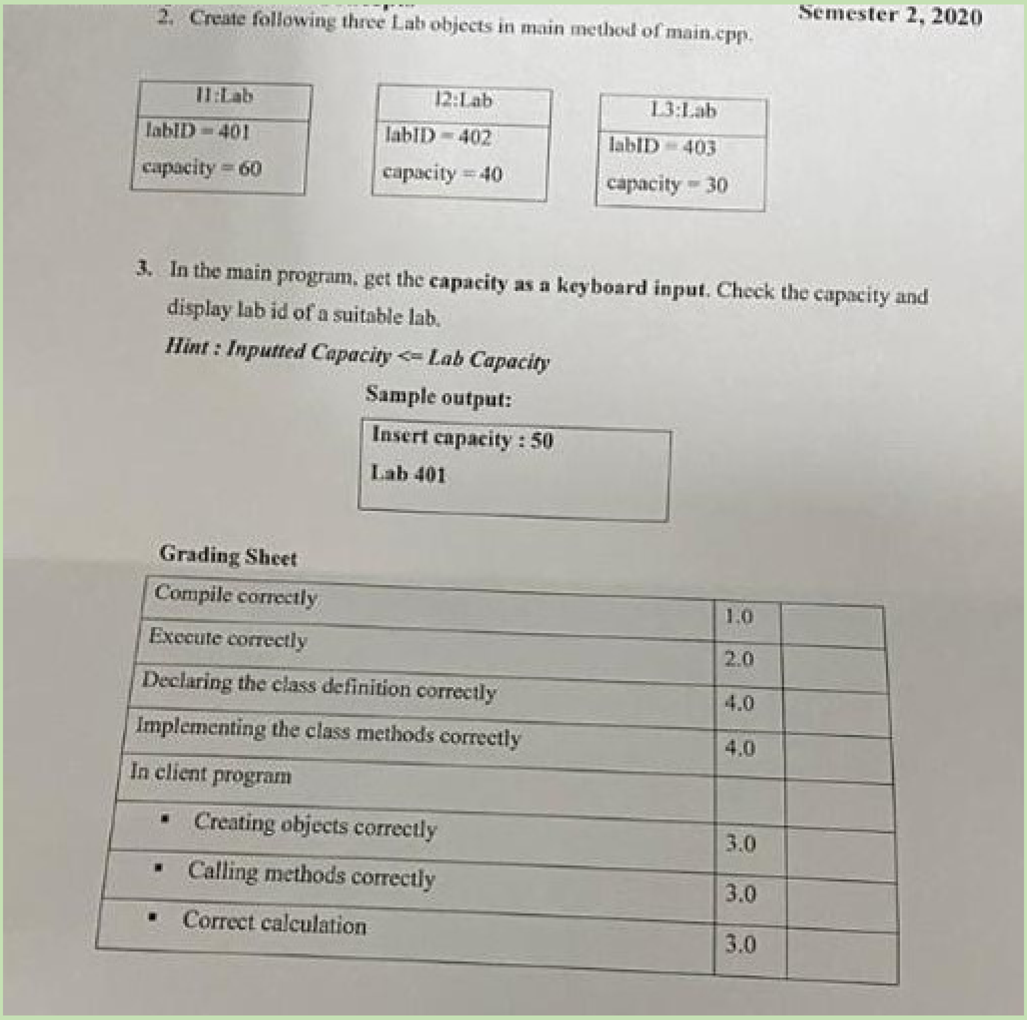
Question :

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()



Semester 2, 2020

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:

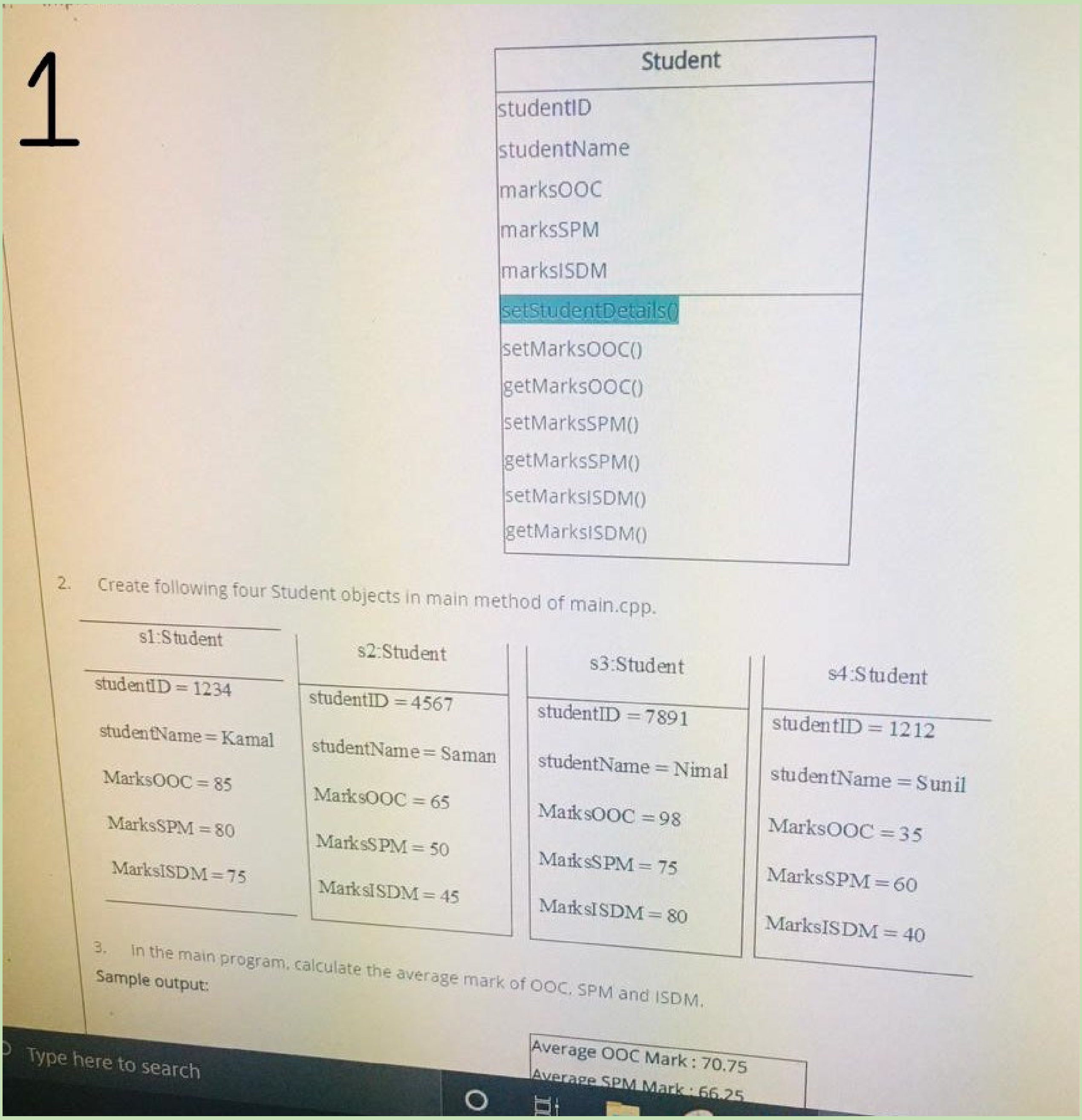
```
Insert capacity : 50
Lab 401
```

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	

```
#include <iostream>

using namespace std;

class Lab{
        private:
                int labID;
                int capacity;
        public:
                void setLabDetails(int id, int capt);
                int getCapacity();
};

void Lab::setLabDetails(int id, int capt)
{
        labID = id;
        capacity = capt;
}

int Lab::getCapacity()
{
        return capacity;
}

int main ()
{
        Lab l1,l2,l3;

        l1.setLabDetails(401,60);
        l2.setLabDetails(402,40);
        l3.setLabDetails(403,30);

        int ncapt;
        cout <<"Insert Capacity : ";
        cin >> ncapt;

        if (ncapt <= l3.getCapacity())
        {
                cout << "Lab 403"<<endl;
        }

        else if (ncapt <= l2.getCapacity())
        {
                cout << "Lab 402"<<endl;
        }

        else if (ncapt <= l1.getCapacity())
        {
                cout << "Lab 401"<<endl;
        }

        return 0;
}
```

Student



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.

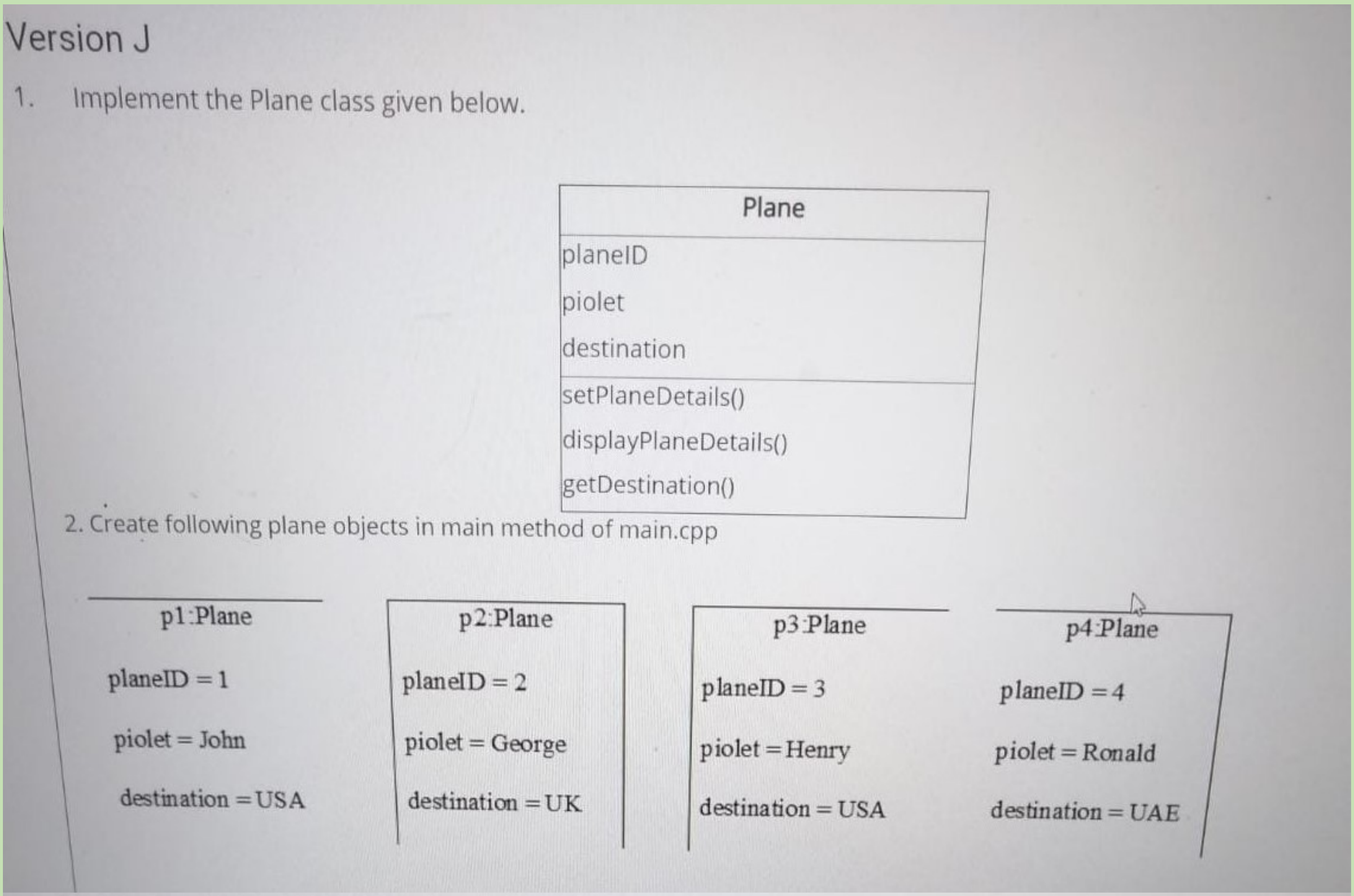
Sample output:

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

```
#include <iostream>

using namespace std;

class Student{
        private:
                int studentID;
                string studentName;
                int marksOOC;
                int marksSPM;
                int marksISDM;
        public:
                void setStudentDetails(int sid, string sname);
                void setMarksOOC(int ooc);
                int getMarksOOC();
                void setMarksSPM(int spm);
                int getMarksSPM();
                void setMarksISDM(int isdm);
                int getMarksISDM();
};

void Student::setStudentDetails(int sid, string sname)
{
        studentID = sid;
        studentName = sname;
}

void Student::setMarksOOC(int ooc)
{
        marksOOC = ooc;
}

int Student::getMarksOOC()
{
	return marksOOC;
}

void Student::setMarksSPM(int spm)
{
	marksSPM = spm;
}

int Student::getMarksSPM()
{
	return marksSPM;
}

void Student::setMarksISDM(int isdm)
{
	marksISDM = isdm;
}

int Student::getMarksISDM()
{
	return marksISDM;
}

int main()
{
	Student s1,s2,s3,s4;

	s1.setStudentDetails(1234,"Kamal");
	s1.setMarksOOC(85);
	s1.setMarksSPM(80);
	s1.setMarksISDM(75);

	s2.setStudentDetails(4567,"Saman");
	s2.setMarksOOC(65);
```

```
        s2.setMarksSPM(50);
        s2.setMarksISDM(45);

        s3.setStudentDetails(7891,"Nimal");
        s3.setMarksOOC(98);
        s3.setMarksSPM(75);
        s3.setMarksISDM(80);

        s4.setStudentDetails(1212,"Sunil");
        s4.setMarksOOC(35);
        s4.setMarksSPM(60);
        s4.setMarksISDM(40);

        float avgOOC,avgSPM,avgISDM;

        avgOOC = (s1.getMarksOOC()+s2.getMarksOOC()+s3.getMarksOOC()+s4.getMarksOOC())/4.0;
        avgSPM = (s1.getMarksSPM()+s2.getMarksSPM()+s3.getMarksSPM()+s4.getMarksSPM())/4.0;
        avgISDM = (s1.getMarksISDM()+s2.getMarksISDM()+s3.getMarksISDM()+s4.getMarksISDM())/4.0;

        cout << "Average OOC mark : " << avgOOC << endl;
        cout << "Average SPM mark : " << avgSPM << endl;
        cout << "Average ISDM mark : " << avgISDM << endl;

        return 0;
}
```

Plane



Version J

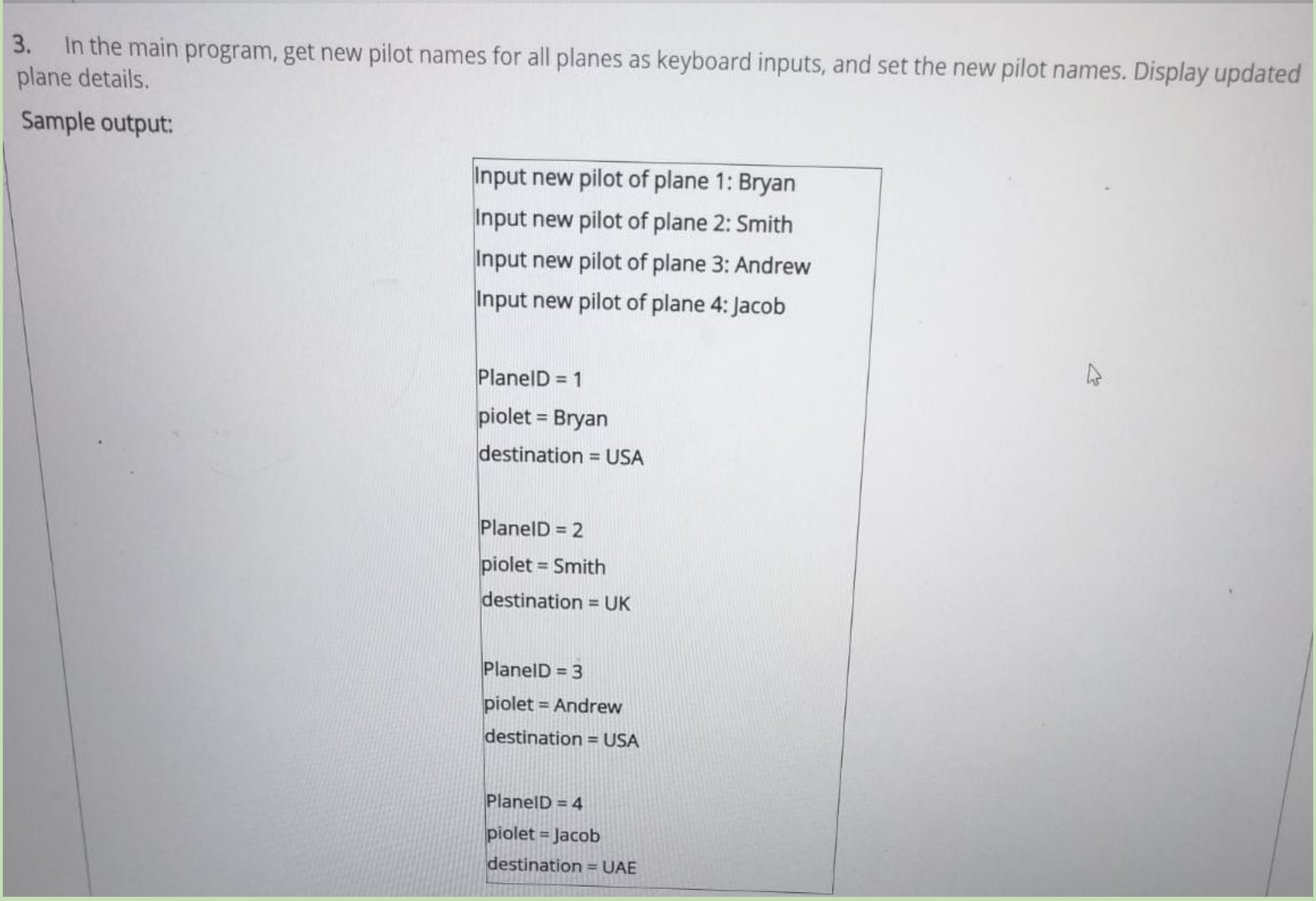
1. Implement the Plane class given below.

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



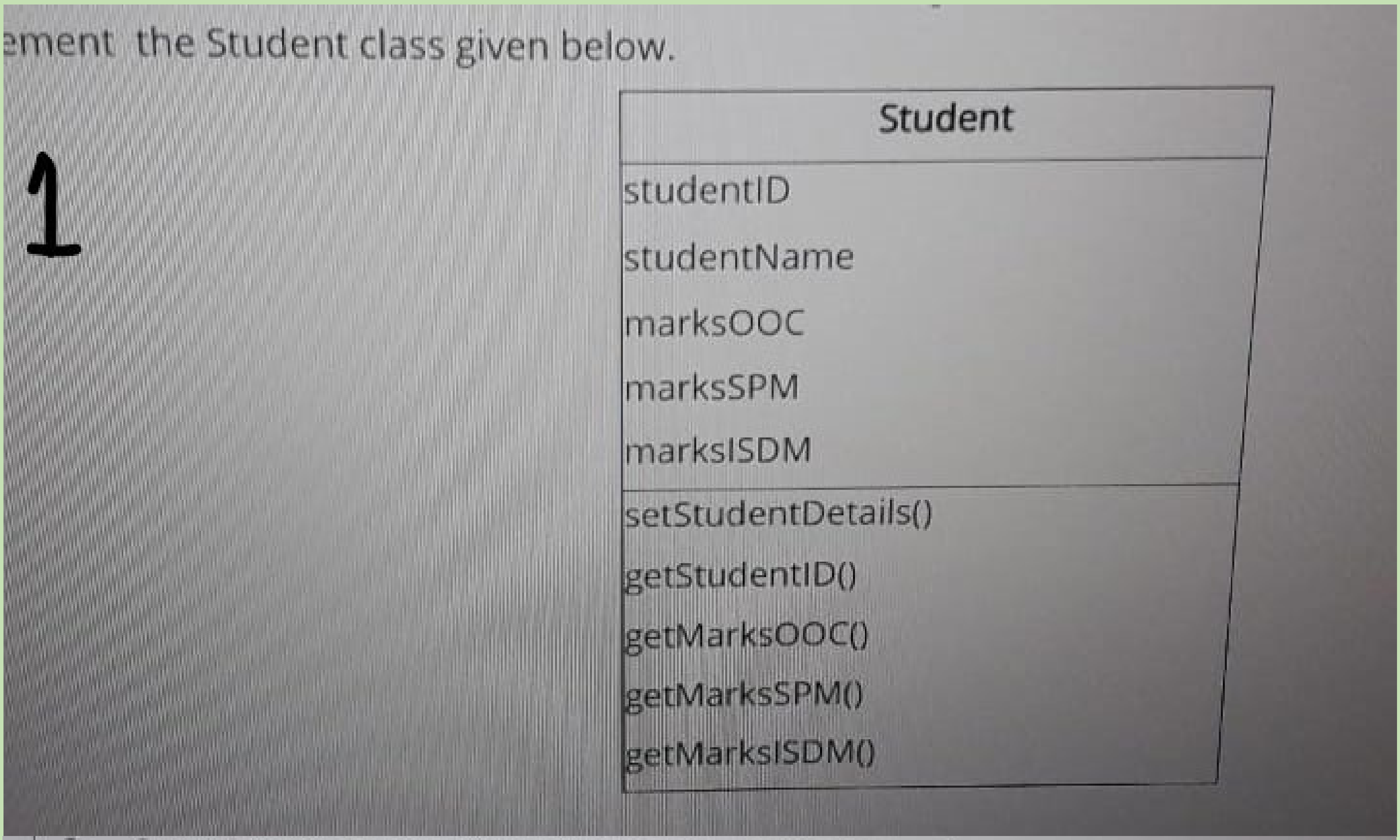
3. In the main program, get new pilot names for all planes as keyboard inputs, and set the new pilot names. Display updated plane details.

Sample output:

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

```
#include <iostream>

using namespace std;

class Plane {
        private:
                int planeID;
                string piolet;
                string destination;
        public:
                void setPlaneDetails(int pID, string pname, string pdest);
                void displayPlaneDetails();
                void getPioletName();
};

void Plane::setPlaneDetails(int pID, string pname, string pdest)
{
        planeID = pID;
        piolet = pname;
        destination = pdest;
}

void Plane::displayPlaneDetails()
{
        cout << "Plane ID = " << planeID << endl
                 << "Piolet = " << piolet << endl
                 << "Destination = " << destination <<endl<<endl;
}

void Plane::getPioletName()
{
        cout << "Input new pilot of plane " << planeID <<" : ";
        cin >> piolet;
}


int main ()

{

        Plane p1,p2,p3,p4;


        p1.setPlaneDetails(1,"John","USA");

        p2.setPlaneDetails(2,"George","UK");

        p3.setPlaneDetails(3,"Henry","USA");

        p4.setPlaneDetails(4,"Ronald","UAE");


        p1.getPioletName();

        p2.getPioletName();

        p3.getPioletName();

        p4.getPioletName();


        cout<<endl;


        p1.displayPlaneDetails();

        p2.displayPlaneDetails();

        p3.displayPlaneDetails();

        p4.displayPlaneDetails();


        return 0;

}
```

Student



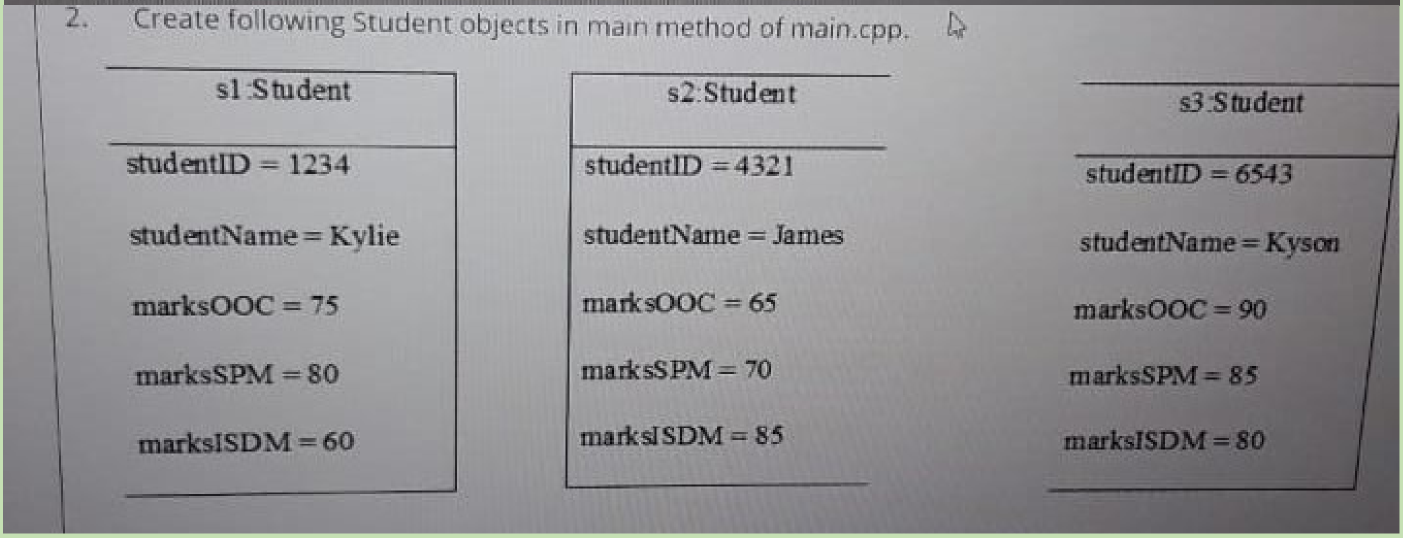
ement the Student class given below.

1

Student

studentID
studentName
marksOOC
marksSPM
marksISDM

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



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

s1:Student

studentID = 1234
studentName = Kylie
marksOOC = 75
marksSPM = 80
marksISDM = 60

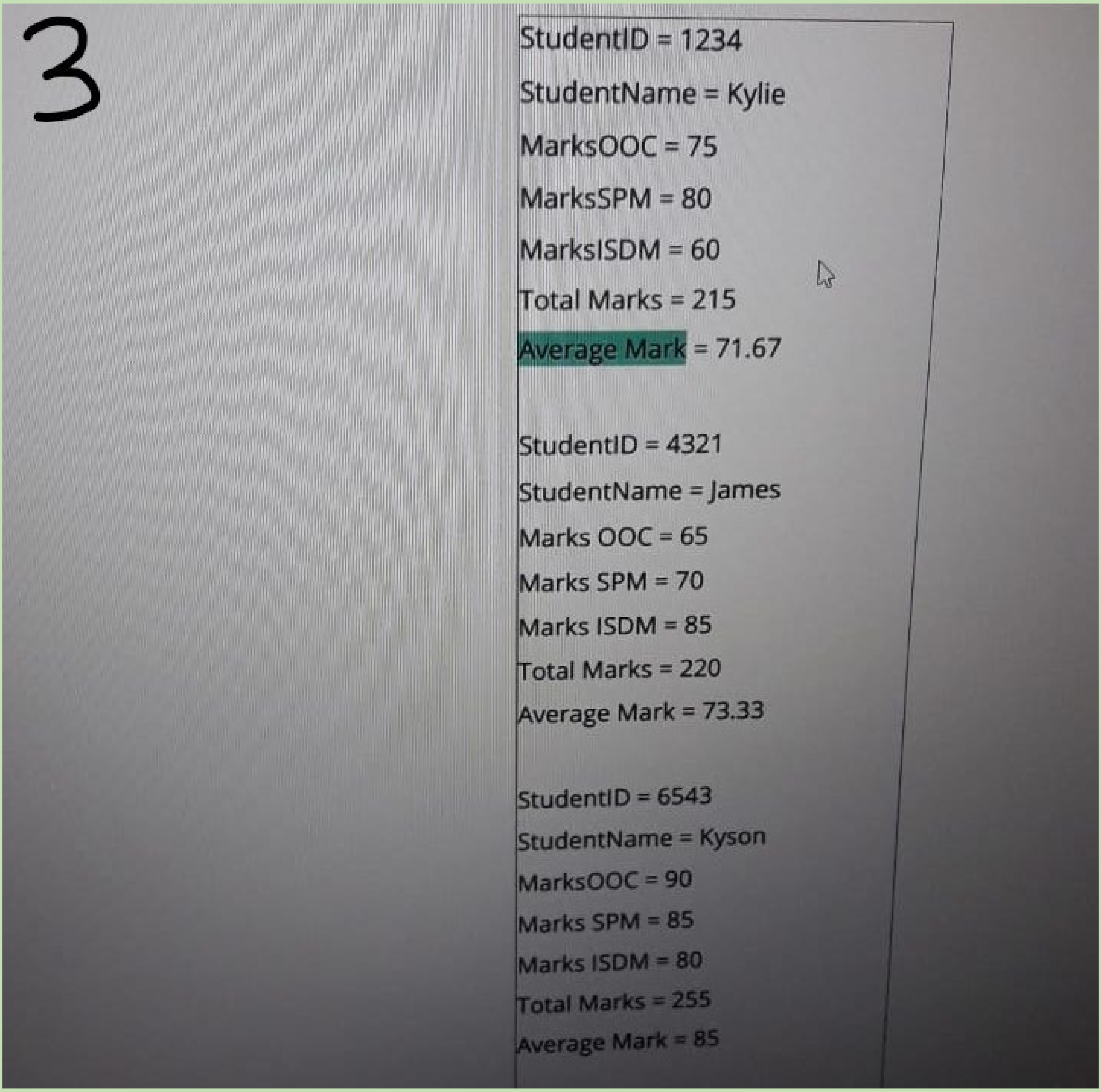
s2:Student

studentID = 4321
studentName = James
marksOOC = 65
marksSPM = 70
marksISDM = 85

s3:Student

studentID = 6543
studentName = Kyson
marksOOC = 90
marksSPM = 85
marksISDM = 80

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
```

```cpp
#include <iostream>
#include <iomanip>

using namespace std;

class Student{
	private:
		int studentID;
		string studentName;
		int marksOOC;
		int marksSPM;
		int marksISDM;
	public:
		void setStudentDetails(int sid, string sname, int ooc, int spm, int isdm);
		int getStudentID();
		int getMarksOOC();
		int getmarksSPM();
		int getmarksISDM();
		string getstudentName();
};

void Student::setStudentDetails(int sid, string sname, int ooc, int spm, int isdm)
{
	studentID = sid;
	studentName = sname;
	marksOOC = ooc;
	marksSPM = spm;
	marksISDM = isdm;
}

int Student::getStudentID()
{
	return studentID;
```

```
}

int Student::getMarksOOC()
{
        return marksOOC;
}

int Student::getmarksSPM()
{
        return marksSPM;
}

int Student::getmarksISDM()
{
        return marksISDM;
}

string Student::getstudentName()
{
        return studentName;
}

int main ()
{
        Student s1,s2,s3;

        s1.setStudentDetails(1234,"kylie",75,80,60);
        s2.setStudentDetails(4321,"james",65,70,85);
        s3.setStudentDetails(6543,"kyson",90,85,80);

        int s1TM, s2TM, s3TM;

        s1TM = s1.getmarksISDM()+s1.getMarksOOC()+s1.getmarksSPM();
        s2TM = s2.getmarksISDM()+s2.getMarksOOC()+s2.getmarksSPM();
```

```
    s3TM = s3.getmarksISDM()+s3.getMarksOOC()+s3.getmarksSPM();

    cout << "Student ID = " << s1.getStudentID() << endl;
    cout << "StudentName = "<< s1.getstudentName() <<endl;
    cout << "Marks OOC = " << s1.getMarksOOC() << endl;
    cout << "Marks SPM = " << s1.getmarksSPM() << endl;
    cout << "Marks ISDM = " << s1.getmarksISDM()<< endl;
    cout << "Total Marks = " << s1TM << endl;
    cout << "Average Mark = " << setiosflags(ios::fixed) <<setprecision(2) << s1TM/3.0 << endl << endl;

    cout << "Student ID = " << s2.getStudentID() << endl;
    cout << "StudentName = "<< s2.getstudentName() <<endl;
    cout << "Marks OOC = " << s2.getMarksOOC() << endl;
    cout << "Marks SPM = " << s2.getmarksSPM() << endl;
    cout << "Marks ISDM = " << s2.getmarksISDM()<< endl;
    cout << "Total Marks = " << s2TM << endl;
    cout << "Average Mark = " <<setprecision(2) << s2TM/3.0 << endl << endl;

    cout << "Student ID = " << s3.getStudentID() << endl;
    cout << "StudentName = "<< s3.getstudentName() <<endl;
    cout << "Marks OOC = " << s3.getMarksOOC() << endl;
    cout << "Marks SPM = " << s3.getmarksSPM() << endl;
    cout << "Marks ISDM = " << s3.getmarksISDM()<< endl;
    cout << "Total Marks = " << s3TM << endl;
    cout << "Average Mark = " <<setprecision(2) << s3TM/3.0 << endl;
    return 0;
}
```

Best of luck…