OOP LAB #04
LAB TASK

Example 12-9

```
#include <iostream>
#include <string>
using namespace std;
class personType
{
public:
void print() const;
//Function to output the first name and last name
//in the form firstName lastName.
void setName(string first, string last);
//Function to set firstName and lastName according
//to the parameters.
//Postcondition: firstName = first; lastName = last;
string getFirstName() const;
//Function to return the first name.
//Postcondition: The value of firstName is returned.
string getLastName() const;
//Function to return the last name.
//Postcondition: The value of lastName is returned.
personType(string first = "", string last = "");
//Constructor
//Sets firstName and lastName according to the parameters.
//The default values of the parameters are null strings.
//Postcondition: firstName = first; lastName = last;
private:
string firstName; //variable to store the first name
string lastName; //variable to store the last name
};
void personType::print() const
{
	cout << firstName << " " << lastName;
}
void personType::setName(string first, string last)
{
	firstName = first;
	lastName = last;
}
string personType::getFirstName() const
{
	return firstName;
}
string personType::getLastName() const
{
	return lastName;
}
```

```
//constructor
personType::personType(string first, string last)
{
	firstName = first;
	lastName = last;
}
```

Task-1

Example 12-9 defined a class personType to store the name of a person. The member functions that we included merely print the name and set the name of a person. Redefine the class personType so that, in addition to what the existing class does, you can:

a) Set the first name only.
b) Set the last name only.
c) Store and set the middle name.
d) Check whether a given first name is the same as the first name of this person.
e) Check whether a given last name is the same as the last name of this person.

Write the definitions of the member functions to implement the operations for this class. Also, write a program to test various operations on this class.

Task-2

In this exercise, you will design various classes and write a program to computerize the billing system of a hospital.

a) Design the class doctorType, inherited from the class personType, defined in Chapter 12, with an additional data member to store a doctor's speciality. Add appropriate constructors and member functions to initialize, access, and manipulate the data members.
b) Design the class billType with data members to store a patient's ID and a patient's hospital charges, such as pharmacy charges for medicine, doctor's fee, and room charges. Add appropriate constructors and member functions to initialize and access and manipulate the data members.
c) Design the class patientType, inherited from the class personType, defined in Chapter 12, with additional data members to store a patient's ID, age, date of birth, attending physician's name, the date when the patient was admitted in the hospital, and the date when the patient was discharged from the hospital. (Use the class dateType to store the date of birth, admit date, discharge date, and the class doctorType to store the attending physician's name.)
d) Add appropriate constructors and member functions to initialize, access, and manipulate the data members.

Write a program to test your classes.