Topic : Health Insurance Management System

Group no : MLB_03.02_07

Campus : Malabe

Submission Date :

We declare that this is our own work and this Assignment does not incorporate without acknowledgment any material previously submitted by anyone else in SLIIT or any other university/Institute. And we declare that each one of us equally contributed to the completion of this Assignment.

IT21237904	Gunawardana U.G.C.D	it21237904@my.sliit.lk	0771504890
IT21242236	Jayakody J.A.D.H.S	it21242236@my.sliit.lk	0763694142
IT21240492	Gunarathna A.M.K.D	it21240492@my.sliit.lk	0742512760
IT21229848	Weerasinghe H	it21229848@my.sliit.lk	0710874212
IT21245060	Weerakoon H.P.S.P	it21245060@my.sliit.lk	0764114869

	Student ID	Name	Contribution
1	IT21237904	Gunawardana U.G.C.D	Description and noun verb analysis Classes : -policy -individual -group -family -seniorCitizen
2	IT21242236	Jayakody J.A.D.H.S	Class : payment
3	IT21240492	Gunarathna A.M.K.D	Class : subscription
4	IT21229848	Weerasinghe H	Class : client
5	IT21245060	Weerakoon H.P.S.P	Class : claim

Object Oriented Concepts – IT1050
Assignment 2

Description

Wellness Health Insurance is a famous health insurance company situated in Colombo. It has an online management system to manage activities for all users of the system.

A registered client can view or search the insurance policies available and make a subscription to a policy by providing the required details such as monthly income.

A subscription has its id, start date and end date.

Policy consists of individual, group, family, and senior citizen types.

The manager can view, add policies under each category, update and delete policies if needed.

The client must make a yearly payment for the subscribed insurance policy. During an emergency the client can make a claim from the insurance by providing the medical bill.

Nouns – Verbs

Nouns

- Client - Class
- Policy – Class
- Subscription – Class
- Monthly income – attribute of subscription
- Id, start date, end date – attributes of subscription
- Individual - Class
- Group - Class
- Family - Class
- Senior citizen - Class
- Manger – actor of the system
- Payment - Class
- Claim - Class
- Medical bill – Attribute of claim

CRC Cards

Client	
Responsibilities	**Collaborations**
Register details	
Subscribe to policy	policy
Make payment	payment
Make claim	claim

Policy	
Responsibilities	**Collaborations**
Display policy	
Search policy	
Store policy details	

Subscription	
Responsibilities	**Collaborations**
Display subscription	
Store subscription details	

Individual	
Responsibilities	**Collaborations**
Store details	
Display details	

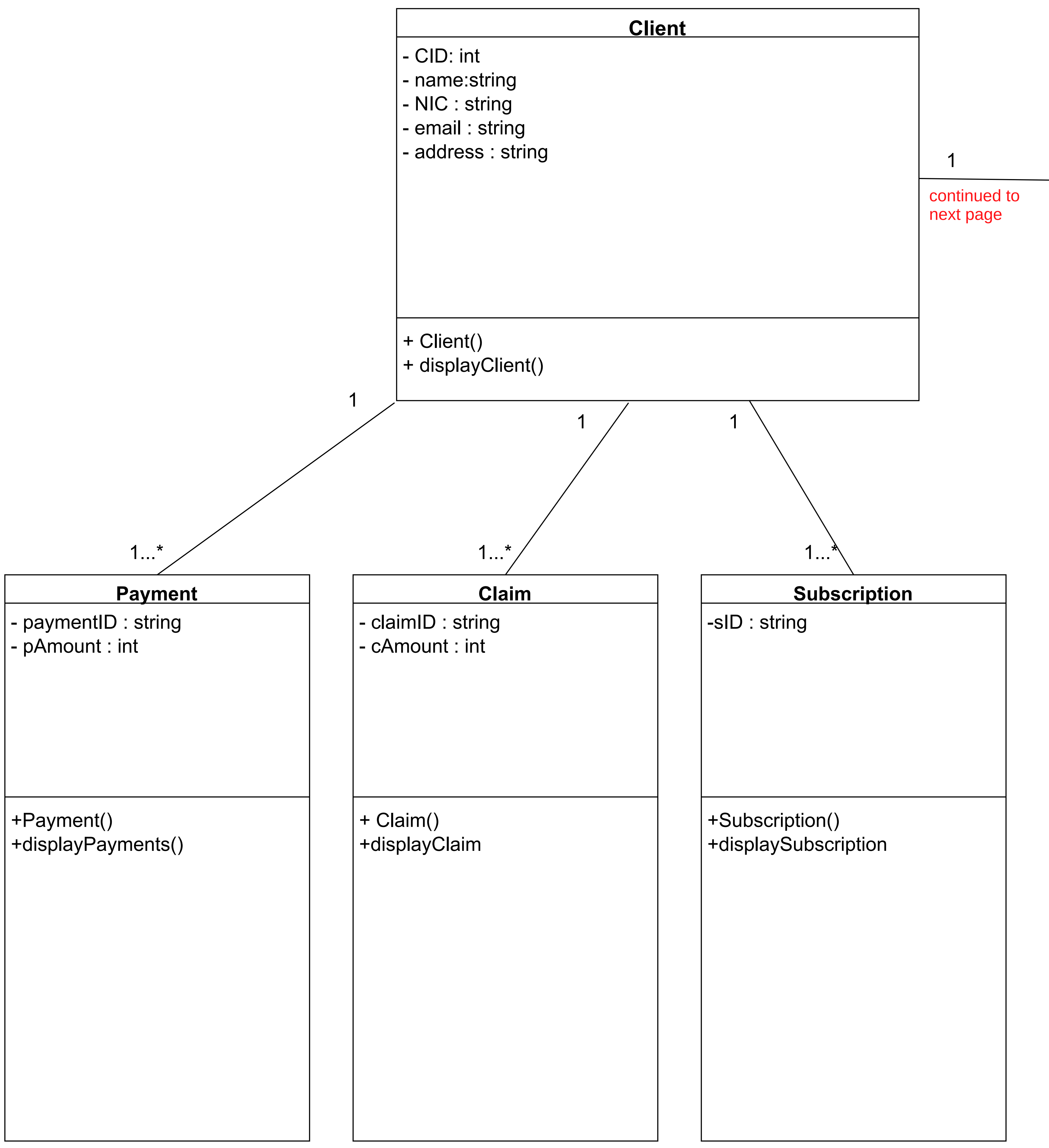
Group	
Responsibilities	**Collaborations**
Store details	
Display details	

Family	
Responsibilities	**Collaborations**
Store details	
Display details	

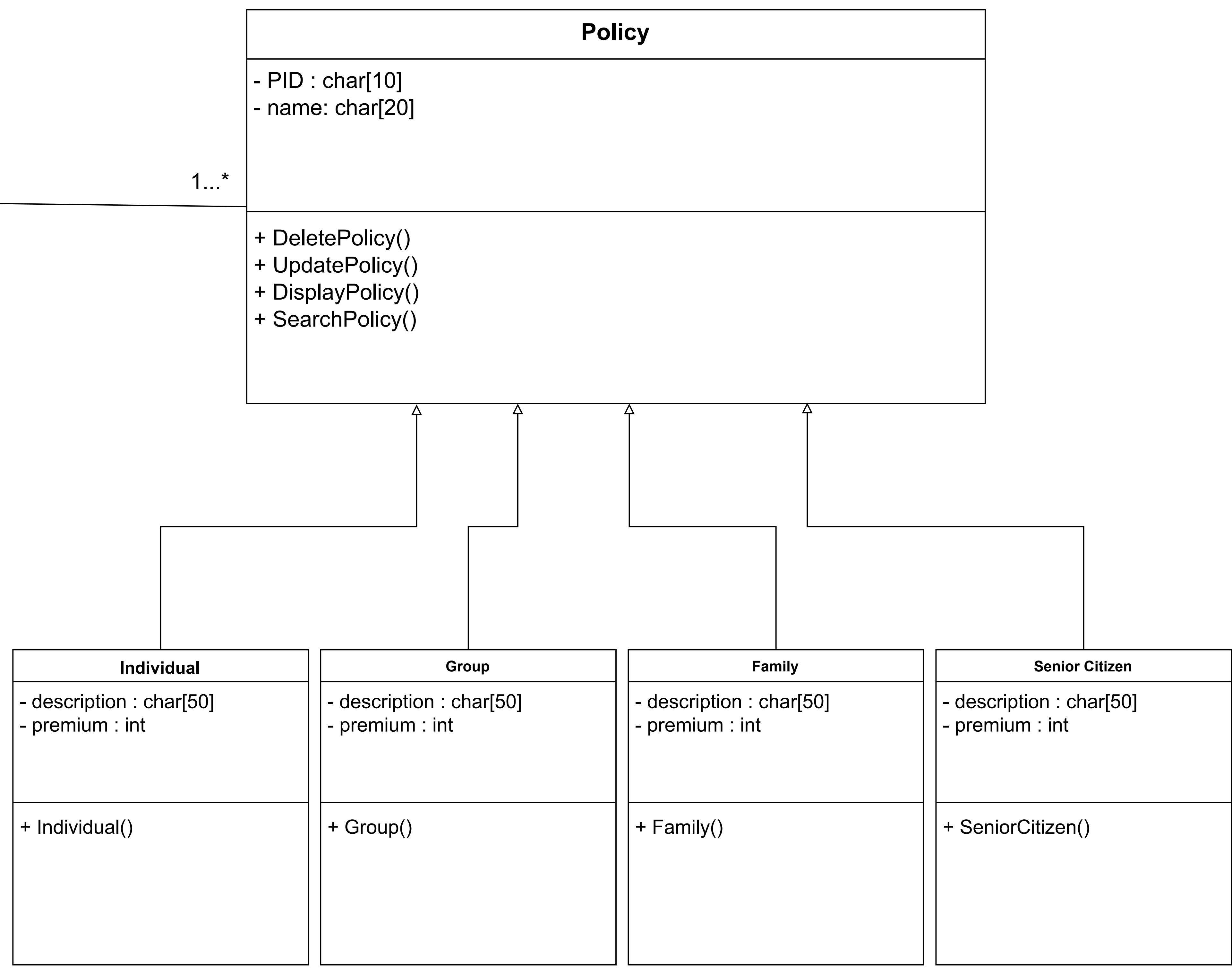
Senior citizen	
Responsibilities	**Collaborations**
Store details	
Display details	

Payment	
Responsibilities	**Collaborations**
Store details	client

Claim	
Responsibilities	**Collaborations**
Store details	client



continued to next page


1...*
Policy
- PID : char[10]
- name: char[20]
+ DeletePolicy()
+ UpdatePolicy()
+ DisplayPolicy()
+ SearchPolicy()
Individual
- description : char[50]
- premium : int
+ Individual()
Group
- description : char[50]
- premium : int
+ Group()
Family
- description : char[50]
- premium : int
+ Family()
Senior Citizen
- description : char[50]
- premium : int
+ SeniorCitizen()

Contents

policy.h

```
//uni directional association with client
#include <iostream>
#include <string>
using namespace std;

class Policy {
        protected :
          char PID[10];
          char pname[20];



        public:
   Policy();
          void DeletePolicy();
          void UpdatePolicy();
          void  DisplayPolicy();
          void  SearchPolicy();
};
```

individual.h

```
#include <iostream>

#include <cstring>

using namespace std;


class Individual : public Policy {  //derived class from policy

        protected :

          char description[50];

          int premium;

        public:

          Individual(char pid[], char name[], char desc[], int prem) { //overloading constructor

            strcpy(PID, pid);

            strcpy(pname, name);

            strcpy(description, desc);

            premium = prem;

          }

                void displayDetails() {  //method to display details

                        cout << PID << endl;

                        cout << pname << endl;

                        cout << description << endl;

                        cout << premium << endl;

                }

        };
```

group.h

```
#include <iostream>

#include <cstring>

using namespace std;


class Group : public Policy {  //derived class from policy

        protected :

          char description[50];

          int premium;

        public:

          Group(char pid[], char name[], char desc[], int prem) { //overloading constructor

            strcpy(PID, pid);

            strcpy(pname, name);

            strcpy(description, desc);

            premium = prem;

          }

                void displayDetails() {  //method to display details

                        cout << PID << endl;

                        cout << pname << endl;

                        cout << description << endl;

                        cout << premium << endl;

                }

        };
```

family.h

```
#include <iostream>

#include <cstring>

using namespace std;


class Family : public Policy {  // derived class from policy

        protected :

          char description[50];

          int premium;

        public:

          Family(char pid[], char name[], char desc[], int prem) { //overloading constructor

            strcpy(PID, pid);

            strcpy(pname, name);

            strcpy(description, desc);

            premium = prem;

          }

                void displayDetails() {  //method to display details

                        cout << PID << endl;

                        cout << pname << endl;

                        cout << description << endl;

                        cout << premium << endl;

                }

        };
```

seniorCitizen

```
#include <iostream>

#include <cstring>

using namespace std;


class SeniorCitizen : public Policy {  //derived class from policy

        protected :

          char description[50];

          int premium;

        public:

          SeniorCitizen(char pid[], char name[], char desc[], int prem) { //overloading
constructor

            strcpy(PID, pid);

            strcpy(pname, name);

            strcpy(description, desc);

            premium = prem;

          }

                void displayDetails() {  //method to display details

                        cout << PID << endl;

                        cout << pname << endl;

                        cout << description << endl;

                        cout << premium << endl;

                }

        };
```

payment.h

```
//bi directional association with client
#include <iostream>
#include <string>
using namespace std;

class Payment
{
   private:
            string paymentID;
            Client *cli;

   public:
            Payment(string          payID,  Client *pcli){
       paymentID = payID;
       cli = pcli;
       cli->addPayment(this);
    }

            void displayPayments(){
      cout << " Payment ID " <<paymentID << endl;
    }
};
```

subscription.h

```
//bi directional association with client

#include <iostream>

#include <string>

using namespace std;


class Subscription

{

   private:

            string sID;

            Client *cli;


   public:

            Subscription(string     subID,  Client *pcli){

       sID = subID;

       cli = pcli;

    }


            void displaySubscription();

    }

};
```

client.h

```
#include <iostream>

#include <string>

#define SIZE 10

using namespace std;


class Client

{

  private:

    int CID;

    string name;

    string NIC;

    string email;

    string address;

    Payment *payment[SIZE];

    int noOfPayments;


    Policy *p;//an object of policy is created as atrribute


  public:

    Client();

    Client(int cCID,string cname,string cNIC,string cEmail, string caddress, Policy *policy){

      CID = cCID;

      name = cname;

      NIC = cNIC;

      email = cEmail;

      address = caddress;

      noOfPayments = 0;

      p = policy;
    }
    void addPayment(Payment *P){
      if(noOfPayments < SIZE)
        order[noOfPayments] = 0;
      noOfOrders++;
    }
    void displayClient(){
      cout << "Customer CID =" << CID << endl;
      cout << "Customer name =" << name << endl;
      cout << "Customer NIC =" << NIC << endl;
      cout << "Customer Email =" << email << endl;
      cout << "Customer Address = " << address << endl;

      for(int i = 0; i < noOfPayments; i++)
        Payment[i] -> displayPayments();
    }

};
```

claim.h

```
#include <cstring>
#include <iostream>
using namespace std;

clas claim{
  private:
    string claimID;
    float  amount;
    client * c2;




public :
  claim() //default constructor
  claim(string  cID,float amo){

    claimID=cID;
    amount = amo;



  }
  cliam :: void setClaimDetails()



      claimID=cID;
    amount = amo;
}
claim:: void displayclaim()
{
  cout << "claim ID :"<<claimID<<endl <<cout<<"Amount :"    <<amount
}
```

main.cpp

```
#include <cstring>

#include <iostream>

using namespace std;


clas claim{

  private:

    string claimID;

    float  amount;

    client * c2;






public :

  claim() //default constructor

  claim(string  cID,float amo){


    claimID=cID;

    amount = amo;





  }

  cliam :: void setClaimDetails()






      claimID=cID;

    amount = amo;
}
claim:: void displayclaim()
{
  cout << "claim ID :"<<claimID<<endl <<cout<<"Amount :"    <<amount
}
```