Namespace

```
//SyMarks
#include<iostream>
using namespace std;
namespace SY{
        class SYMARKS{
        int Tcom;
        int Telec;
        int Tmaths;

        public:
                SYMARKS();
                SYMARKS(int,int,int);
                void setTcom(int);
                void setTelec(int);
                void setTmaths(int);

                int getTcom();
                int getTelec();
                int getTmaths();




};
}
//header file
#include<iostream>
using namespace std;

namespace TY{
        class TYmarks{
```

```
    int theory;
    int practical;

    public:
        TYmarks();
        TYmarks(int,int);
        void setTheory(int);
        void setPractical(int);

        int getTheory();
        int getPractical();




};
}

#include"TYmarks.h"

TY::TYmarks::TYmarks(){
    this->theory=0;
    this->practical=0;
}
TY::TYmarks::TYmarks(int t,int p){
    this->theory=t;
    this->practical=p;
}
void TY::TYmarks::setTheory(int t){
    this->theory=t;
}
```

```
void TY::TYmarks::setPractical(int p){
	this->practical=p;
}

int TY::TYmarks::getTheory(){
	return this->theory;
}
int TY::TYmarks::getPractical(){
	return this->practical;
}
#include"SYMARKS.h"

namespace SY{
	SYMARKS::SYMARKS(){
	this->Tcom=0;
	this->Telec=0;
	this->Tmaths=0;
}
SYMARKS::SYMARKS(int c,int e,int m){
	this->Tcom=c;
	this->Telec=e;
	this->Tmaths=m;
}
void SYMARKS::setTcom(int c){
	this->Tcom=c;
}
void SYMARKS::setTelec(int e){
	this->Telec=e;
}
void SYMARKS::setTmaths(int m){
```

```
                        this->Tmaths=m;
}


int SYMARKS::getTcom(){
            return this->Tcom;
}
int SYMARKS::getTelec(){
            return this->Telec;
}
int SYMARKS::getTmaths(){
            return this->Tmaths;
}


}
#include"SYMARKS.h"
#include"TYmarks.h"
using namespace TY;
using namespace SY;
//has a relationship
class Student {
            int rnum;
            char name[20];
            //SY::SYMARKS s; //as we are not defining using namespace SY;
            SYMARKS s;
            TYmarks t;
            public:
            Student();
            Student(int,const char*,SYMARKS,TYmarks);


            void setRnum(int );
            void setName(const char*);
```

```
        void setSYMARKs(SYMARKS);
        void setTYmarks(TYmarks);

        int getRnum();
        char* getName();
        SYMARKS getSYMARKS();
        TYmarks getTYmarks();

        void Result();
        void display();

};

ostream& operator<<(ostream& ,TYmarks&);
ostream& operator<<(ostream&,SYMARKS&);
#include"student.h"



Student::Student(){
        this->rnum=0;
        strcpy(this->name,"Student");
        this->s=SYMARKS(0,0,0);
        this->t=TYmarks(0,0);
}
Student::Student(int rnum,const char* name,SYMARKS sm,TYmarks tm){
        this->rnum=rnum;
        strcpy(this->name,name);
        this->s=sm;
        this->t=tm;
}
```

```
void Student::setRnum(int r ){
        this->rnum=r;
}
void Student::setName(const char *nm){
        strcpy(this->name,nm);
}
void Student::setSYMARKs(SYMARKS sm){
        this->s=sm;
}
void Student::setTYmarks(TYmarks tm){
        this->t=tm;
}

int Student::getRnum()
{
        return this->rnum;
}
char* Student::getName(){
        return this->name;
}
SYMARKS Student::getSYMARKS(){
        return this->s;
}

TYmarks Student::getTYmarks(){
        return this->t;
}

void Student::Result(){
```

```
//      SY::SYMARKS sy=this->getSYMARKS();
//      TY::TYmarks ty=this->getTYmarks();
//      int total=sy.getTcom()+ty.getPractical()+ty.getTheory();

int total=this->getSYMARKS().getTcom()+this->getTYmarks().getPractical()+this->getTYmarks().getTheory();
int avg=total/3;

if(avg>=70){
    cout<<"Stdent:"<<this->name<<"\tGRADE:"<<"A";
}else{
    if(avg>=60){
            cout<<"Stdent:"<<this->name<<"\tGRADE:"<<"B";
    }
    else{
        if(avg>=50){
                cout<<"Stdent:"<<this->name<<"\tGRADE:"<<"C";
            }else{
                if(avg>=40){
                        cout<<"Stdent:"<<this->name<<"\tGRADE:"<<"Fails";
                }
            }
        }
    }



}

void Student::display(){
    cout<<"Student roll number:"<<this->rnum<<"\n";
    cout<<"Student Nane:"<<this->name<<"\n";
```

```
            //cout<<"SYMARKS:\n";
//          cout<<this->getSYMARKS().getTcom()<<"\n";
//          cout<<this->getSYMARKS().getTelec()<<"\n";
//          cout<<this->getSYMARKS().getTmaths()<<"\n";
            cout<<this->s;
            cout<<this->t;
}



ostream& operator<<(ostream& o,SYMARKS& s ){
                              o<<"SYMarks:\n";
                              o<<"marks of computer:"<<s.getTcom()<<"\n";
                              o<<"Marks of electroincs:"<<s.getTelec()<<"\n";
                              o<<"Marks of Maths:"<<s.getTmaths()<<"\n";


                              return o;
                    }


            ostream& operator<<(ostream& o,TYmarks& t ){
                              o<<"TYMarks:\n";
                              o<<"Marks of theory:"<<t.getTheory()<<"\n";
                              o<<"Marks of Practical:"<<t.getPractical()<<"\n";


                              return o;
                    }


#include <iostream>
using namespace std;
#include"student.h"
/* run this program using the console pauser or add your own getch, system("pause") or input loop
*/
```

```
using namespace SY;
using namespace TY;

int main() {
	SY::SYMARKS st(70,80,98);
	TYmarks t(80,70);

	Student s(101,"PRachit",st,t);
	cout<<s.getRnum()<<"\n";
	cout<<s.getName()<<"\n";
	SY::SYMARKS sy=s.getSYMARKS();
	cout<<sy<<"\n";//as we are referencing it must be variable if we cout<<s.getSYMARKS()  s is returning in dummy
	TY::TYmarks ty=s.getTYmarks();
	cout<<ty<<"\n";



	s.Result();
	s.display();
	return 0;
}
```