**Bài 1.**

```cpp
#include <iostream>
#include <string.h>
using namespace std;
class phong
{
        char loaiphong[2];
        int  songay;
public:
        phong();
        phong(char lp[2],int sn);
        void xuat();
        int get_sn();  //lay so ngay
        virtual float tien();    //ham ao, se duoc dinh nghia lai trong cac lop con
};

phong::phong()
{}

phong::phong(char lp[2],int sn)
{
        strcpy(loaiphong,lp);  //SAI: loaiphong = lp
        songay=sn;
}

int phong::get_sn()
{        return songay; }
```

```cpp
void phong::xuat()
{       cout<<"Loai phong: "<<loaiphong<<", so ngay thue: "<<songay<<endl; }


float phong::tien()
  { return 0;}


class phongA : public phong
{
   float tiendv;
public:
       phongA();
       phongA(char lp[2],int sn, float dvu);
       float tien();
};


phongA::phongA()
{}


phongA::phongA(char lp[2],int sn, float dvu):phong(lp,sn)
{

 tiendv = dvu;
}


float phongA::tien()
{
       if(get_sn()<5)
               return 80*get_sn()+tiendv;
```

```
        else
                return (80*get_sn()+tiendv)*0.9;
}


class phongB : public phong
{
public:
        phongB();
        phongB(char lp[2],int sn);
        float tien();
};


phongB::phongB()
{}


phongB::phongB(char lp[2],int sn):phong(lp,sn)
 {}


float phongB:: tien()
{
        if(get_sn()<5)
                return 60*get_sn();
        else
                return 60*get_sn()*0.9;
}

class phongC : public phong
```

```cpp
{
public:
        phongC();
        phongC(char lp[2],int sn);
        float tien();
};


phongC::phongC(void)
{}
phongC::phongC(char lp[2],int sn):phong(lp,sn)
{}


float phongC::tien()
{       return 40*get_sn();}


int main()
{
        phong *DS[100];
        int n,tl,sngay;
        cout<<"nhap so luong cac phong khach da thue: "; cin>>n;
        for (int i=0; i<n; i++)
        {
                cout<<"Loai phong (1-A,2-B,3-C): "; cin>>tl;
                cout<<"So ngay thue: "; cin>>sngay;  //thong tin chung
                switch (tl)
                {
                case 1:
                        {       float dv;cout<<"tien dich vu: "; cin>>dv;
```

```cpp
                        phongA *p=new phongA("A",sngay,dv);

                        DS[i]=p;

                        break;

                }

        case 2:

                {       phongB *p=new phongB("B",sngay);

                        DS[i]=p;

                        break;

                }

        case 3:

                {       phongC *p=new phongC("C",sngay);

                        DS[i]=p;

                        break;

                }

        }

    }


    float s = 0;

        for (int i=0; i<n; i++)

                {

                        s = s + DS[i]->tien();

                        DS[i]->xuat();

                }

        cout<<"\n Tong so tien khach thue phong la: "<<s;

        return 0;

}
```

**Bài 2.**

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

```cpp
#include <iomanip>
#include <string>
#include <stdio.h>
using namespace std;

class xe
{
        string hoten;
        string loaixe;
        int gio;
public:
        xe();
        xe(string ht, string lx, int g);
        virtual void xuat();        //ham ao
        virtual float tienthue();  //ham ao
        int get_gio();
        string get_loaixe();
};

xe::xe()
{}

xe::xe(string ht, string lx, int g)
{
        hoten = ht;
        loaixe = lx;
        gio = g;
}
```

```cpp
void xe::xuat()
{
        cout<<hoten<<setw(10)<<loaixe<<setw(10)<<gio<<setw(10);
}


float xe::tienthue()
{
  return 0;
}


int xe::get_gio()
{
   return gio;
}


string xe::get_loaixe()
{
        return loaixe;
}


class xedap : public xe
{
 public:
        xedap();
        xedap(string ht, string lx, int g);
        void xuat();
        float tienthue();
```

```cpp
};

xedap::xedap(void)
{}

xedap::        xedap(string ht, string lx, int g):xe(ht, lx, g)
{}

void xedap::xuat()
{
        xe::xuat();
}

float xedap::tienthue()
{
        if(get_gio()<=1)
                return float(30000);
        else
                return (float)30000 + (get_gio()-1)*20000;
}

class xemay: public xe
{
        int bienso;
public:
        xemay();
        xemay(string ht, string lx, int g, int bs);
        void xuat();
```

```cpp
        float tienthue();
};


xemay::xemay()

{}


xemay::xemay(string ht, string lx, int g, int bs):xe(ht, lx,g)

{

   bienso=bs;

}


float xemay::tienthue()

{

        if(get_loaixe()=="100")

                return float(15000)*get_gio();

        else

                return float(20000)*get_gio();

}


void xemay::xuat()

{

   xe::xuat();

        cout<<setw(10)<<bienso<<setw(10);

}


int main()

{

        xe *DS[100];
```

```cpp
int n,tl,g;
string ht, lx;
cout<<"\n Nhap tong so xe cho thue: "; cin>>n;
for (int i=0; i<n; i++)
{
        cout<<"1-xe dap,2-xe may: "; cin>>tl;
        cout<<"ho ten: "; fflush(stdin); getline(cin,ht);
        cout<<"So gio thue: "; cin>>g;
        cout<<"Loai xe: "; fflush(stdin); getline(cin,lx);


        switch(tl)
        {
        case 1:
                {       xedap *p=new xedap(ht,lx,g);
                        DS[i]=p;
                        break;
                }
        case 2:
                {       int bs;
                        cout<<"bien so: "; cin>>bs;
                        xemay *p=new xemay(ht,lx,g,bs);
                        DS[i]=p;
                        break;
                }
        }
}
cout<<"\n Danh sach cac xe da thue: "<<endl;
for (int i=0; i<n; i++)
```

```
                    {

                        DS[i]->xuat();

            cout<<setw(10)<<DS[i]->tienthue()<<"\n";;

                    }

            return 0;

}
```

**Bài 3.**

**//Hướng dẫn câu 1**

```cpp
#include <iostream>

#include <string>

#include <iomanip>

using namespace std;


class Date {

    private:

        int d;

        int m;

        int y;

    public:

        Date() { };

        Date(int d, int m, int y)

                {

            this -> d = d;

            this -> m = m;

            this -> y = y;

        }

        bool operator > (Date ob);

};
```

```cpp
bool Date::operator > (Date ob)
{
    if (y > ob.y)
        return true;
    if (y == ob.y and m > ob.m)
        return true;
    if (y == ob.y and m == ob.m and d > ob.d)
        return true;
    return false;
}

class Person {
    private:
        string ho;
        string ten;
        Date ngaysinh;
    public:
        Person() { };
        Person(string h, string t, Date ns)
                {
            ho = h;
            ten = t;
            ngaysinh = ns;
        }
        void set_ho(string h) { ho = h; }
        string get_ho() { return ho; }
        void set_ten(string t) { ten = t; }
```

```cpp
        string get_ten() { return ten; }
        void set_ngaysinh(Date ns) { ngaysinh = ns; }
        Date get_ngaysinh()  { return ngaysinh; }
};


class Student : public Person {
    private:
        string ma_sv;
        string sdt;
        float dtb;
    public:
        Student() {};
        Student(string h, string t, Date ns, string ma, string dt, float tb)
        : Person {h, t, ns}
                {
            ma_sv = ma;
            sdt = dt;
            dtb = tb;
        }
        void set_masv(string ms) { ma_sv = ms; }
        string get_masv() const { return ma_sv; }
        void set_sdt(string _phone) { this -> sdt = _phone; }
        string get_sdt() const { return sdt; }
        void set_dtb(float tb) { this -> dtb = tb; }
        float gettb() { return dtb; }


        bool operator > (Student ob);
        bool operator >= (Student ob);
```

```cpp
        bool operator < (Student ob);

        void set_Student();

        void print_Student();

};


bool Student::operator > (Student ob) {

    return (get_ngaysinh() > ob.get_ngaysinh());

}


bool Student::operator >= (Student ob) {

    return (this -> gettb() >= ob.gettb());

}


bool Student::operator < (Student ob) {

    if (this -> get_ten() < ob.get_ten())

        return true;

    else if (this -> get_ten() == ob.get_ten()) {

        if (this -> get_ho() < ob.get_ho())

            return true;

     return false;

    }

    return false;

}
//Sinh viên tự giải tiếp câu 2
```