

1. Xây dựng lớp **Canbo** mô tả thông tin về cán bộ, bao gồm các thuộc tính: Mã số cán bộ, họ tên, đơn vị, hệ số lương, bảo hiểm. Các phương thức của lớp **Canbo** bao gồm:

- Nhập dữ liệu.
- Hiển thị dữ liệu ra màn hình.
- Tính lương của cán bộ, biết rằng lương được tính theo công thức:  
$$\text{Lương} = \text{hệ số lương} * 1350000 - \text{bảo hiểm}$$
- Phương thức trả về giá trị của thuộc tính đơn vị.

2. Xây dựng lớp **Nhanvien** mô tả cán bộ hợp đồng, lớp **Nhanvien** được kế thừa từ lớp **Canbo** và có thêm thuộc tính: ngày hợp đồng (bao gồm ngày, tháng, năm). Các phương thức:

- Nhập dữ liệu, không yêu cầu kiểm tra tính hợp lệ của dữ liệu ngày.
- Hiển thị dữ liệu ra màn hình.
- Toán tử tải bội > để so sánh hai đối tượng thuộc lớp **Nhanvien** theo ngày hợp đồng.

3. Xây dựng lớp **Quanly** (quản lý cán bộ) với các thuộc tính và phương thức thích hợp để thực hiện được các yêu cầu:

- Nhập vào mảng không quá 50 đối tượng thuộc lớp **Nhanvien**.
- Sắp xếp các đối tượng theo thứ tự tăng dần của ngày hợp đồng và hiển thị ra màn hình danh sách cán bộ đã sắp xếp, thông tin mỗi cán bộ được hiển thị trên một dòng.
- In bảng lương cán bộ đã nhập theo từng đơn vị, thông tin mỗi cán bộ được hiển thị trên một dòng.

4. Viết chương trình minh họa trên một đối tượng của lớp **Quanly**.

**Giải:**

```
//sua bai on tap 2
```

```
#include<stdio.h>
```

```
#include<iomanip>
```

```
#include<iostream>
```

```
using namespace std;
```

```
class Canbo
```

```
{
```

```
private:
```

```
    string maso, hoten, donvi;
```

```
    float heso, baohiem;
```

```
public:
```

```

void nhap()
{
    cout<<"\n Ma can bo: ";    getline(cin,maso);
    cout<<"\n Ho ten: ";        getline(cin,hoten);
    cout<<"\n Don vi: ";        getline(cin,donvi);
    cout<<"\n He so luong: ";cin>>heso;
    cout<<"\n Bao hiem: ";  cin>>baohiem;
    cin.ignore();
}
float luong()
{
    return heso*1350000 - baohiem;
}
void xuat()
{
    cout<<"\n"<<maso;
    cout<<"\t"<<hoten;
    cout<<"\t"<<heso;
    cout<<"\t"<<baohiem;
    cout<<setprecision(0)<<fixed<<"\t"<<luong();
}
string get_donvi()
{
    return donvi;
}
};

```

```

class Nhanvien:public Canbo
{
private:
    int ng,th,n;
public:
    void nhap()
    {
        Canbo::nhap();
        cout<<"\n Nhap vao ngay hop dong: ";
        cin>>ng>>th>>n;
        cin.ignore();
    }
    void xuat()
    {
        Canbo::xuat();
    }
}

```

```

        cout<<"\t"<<ng<<"/"<<th<<"/"<<n;
    }
    int operator >(Nhanvien a)          //qua tai toan tu >
    {
        if(n>a.n) return 1;
        if((n==a.n)&&(th>a.th)) return 1;
        if((n==a.n)&&(th==a.th)&&(ng>a.ng)) return 1;
        return 0;
    }
};

```

```

class Quanly

```

```

{
    Nhanvien hd[50];
    int n;
    public:
        void nhapcb(); //Cau a
        void xuatcb(); //Cau b
        void xuatcb_dv(); //Cau c
};

void Quanly::nhapcb()
{
    do
    {
        cout<<"\n Nhap so can bo hop dong: ";
        cin>>n;
    }while (n>=50);
    cin.ignore();
    for(int i=0;i<n;i++)
        hd[i].nhap();
}

void Quanly::xuatcb()          //in danh sach co sap xep theo ngay hop dong
{
    for(int i=0;i<n-1;i++)
    for(int j=i+1;j<n;j++)
    if (hd[i]>hd[j])
    {
        Nhanvien tam;
        tam = hd[i];
        hd[i]=hd[j];
        hd[j]= tam;
    }
}

```

```

    for(int i=0;i<n;i++)
        hd[i].xuat();
}

void Quanly::xuacb_dv() //in bang luong theo don vi
{
    //sap xep can bo theo don vi
    for (int i=0;i<n-1;i++)
        for (int j=i+1;j<n;j++)
            if(hd[i].get_donvi() > hd[j].get_donvi())
            {
                Nhanvien tam;
                tam = hd[i];
                hd[i] = hd[j];
                hd[j] = tam;
            }
    //in theo don vi
    for(int i=0 ; i<n ; i++)
    {
        if (i==0 || hd[i].get_donvi() != hd[i-1].get_donvi())
            cout<<"\n Don vi : "<<hd[i].get_donvi()<<endl;
        hd[i].xuat();
    }
}

int main()
{
    Quanly ob;
    ob.nhapcb();
    ob.xuacb();
    ob.xuacb_dv();

    return 0;
}

```