

BÀI II.

Bài 1:

1. Tạo hàm rút gọn phân số như bài trên.
2. Tạo class PhanSo gồm 2 thành phần tử mẫu là private, hàm nhập xuất cộng trừ nhân chia là public
3. Trong public khởi tạo constructor cho PhanSo, đặc mặc định tử là 0, mẫu là 1.
4. Định nghĩa cộng trừ nhân chia đều là 1 phân số mới. Làm như bình thường
5. Xuất phân số dưới dạng rút gọn.

Bài 2:

1. Tạo class tên là **con** gồm thành phần thuc, ao là private, các hàm cộng trừ nhân chia là public.
2. Trong đó, định nghĩa hàm cộng là 1 **con** tên “cong”, rồi truyền vào 1 **con** khác tên b. Khi đó, ta gọi 1 **con** trong hàm main, truy cập vào đối tượng “cong” và truyền vào số phức ta cần thực hiện phép cộng. Khi xuất ra sẽ là 1 số phức. Gán c = số phức đó r xuất.
3. Tương tự với các toán tử khác.

```
}  
con cong(con b)  
{  
    con c;  
    c.thuc = thuc + b.thuc;  
    c.ao = ao + b.ao;  
    return c;  
}  
  
c = a.cong(b);  
cout << "Tong 2 so phuc= ";
```

Bài 3:

1. Tạo class Candidate bao gồm các private theo đề bài đã cho, các thuộc tính bao gồm hàm khởi tạo, hàm xuất và hàm tính tổng điểm 3 môn toán văn anh.

```

public:
    Candidate(string ma, string name, string ntns, float toan, float van, float anh)
    {
        this -> ma = ma;
        this -> name = name;
        this -> ntns = ntns;
        this -> toan = toan;
        this -> van = van;
        this -> anh = anh;
    }
    int sum()
    {
        return toan + van + anh;
    }
    void Xuat()
    {
        cout << "Ma: " << ma << endl;
        cout << "Ten: " << name << endl;
        cout << "Ngay thang nam sinh: " << ntns << endl;
        cout << "Diem toan: " << toan << endl;
        cout << "Diem van: " << van << endl;
        cout << "Diem anh: " << anh << endl;
    }
};

```

2. Tạo thêm 1 lớp class TestCandidate chỉ có public

3. Tạo một hàm vector tĩnh tên là Nhap_ds_ts để :

- Nhập số thí sinh:

```

static vector<Candidate> Nhap_ds_ts()
{
    int n;
    cout << "Nhap vao so thi sinh: " ;
    cin >> n;
}

```

- Khởi tạo 1 vector theo dạng class Candidate có tên là danh_sach để nhập và lưu thông tin số thí sinh

```

vector<Candidate> danh_sach;
for (int i = 0; i < n; i++)
{
    cout << "Nhap vao thong tin thi sinh thu " << i+1 << ":" << endl;
    string ma;
    string name;
    string ntns;
    float toan;
    float van;
    float anh;
    cout << "Ma: ";
    cin.ignore();
    cin >> ma;
    cout << "Ten: ";
    cin.ignore();
    getline(cin, name);
    cout << "Ngay sinh: "; cin.ignore();
    getline(cin, ntns);
    cout << "Diem toan: ";
    cin >> toan;
    cout << "Diem van: ";
    cin >> van;
    cout << "Diem anh: ";
    cin >> anh;

```

```

Candidate thi_sinh(ma, name, ntns, toan, van, anh);
danh_sach.push_back(thi_sinh);

```

```

}
return danh_sach;

```

Ở đây chúng ta đẩy thông tin của thí sinh thứ i vào hàm khởi tạo có tên `thi_sinh` đã được khởi tạo ở lớp `Candidate`.

- Quay lại thí sinh thứ $i + 1$ cho đến khi i không còn thỏa điều kiện của vòng lặp `for`

```
static void display(vector<Candidate> danh_sach)
{
    cout <<endl <<"Danh sach thi sinh co tong diem hon 15: " <<endl;
    for (Candidate thi_sinh : danh_sach) {
        if (thi_sinh.sum() > 15) {
            thi_sinh.Xuat();
        }
    }
}
```

- Ta tạo một hàm tĩnh tên display để in thông tin của các thí sinh thỏa yêu cầu của đề bài là có tổng điểm 3 môn hơn 15
- Với vòng lặp for ta duyệt qua các phần tử của lớp danh_sach đã được ta nhập vào trước để đưa qua đối tượng thi_sinh dưới dạng Candidate rồi kiểm tra lại với hàm sum thỏa điều kiện của đề sau đó được ta xuất ra.