

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA HỆ THỐNG THÔNG TIN



MÔN LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG
BÀI TẬP THỰC HÀNH 1

GVHD: Nguyễn Ngọc Quý

Sinh viên thực hiện: 23521172

This image shows a full page of white paper with horizontal dashed lines, typical of primary-ruled notebook paper. The lines are evenly spaced and run across the entire width of the page. There are no margins, text, or other markings present.

(Ký tên và ghi rõ họ tên)

MỤC LỤC

1. Xây dựng lớp số phức:	6
2. Xây dựng các thuộc tính:	7
3. Xây dựng các phương thức:	8
3.1. Phương thức Nhap():	8
3.2. Phương thức Xuat():	8
3.3. Phương thức Tong():	8
3.4. Phương thức Hieu():	8
3.5. Phương thức Tich():	9
3.6. Phương thức Thuong():	9
3.7. Hàm main():	10
3. Giải quyết vấn đề:	10
4. Thực hiện chạy chương trình:	10

DANH MỤC BẢNG

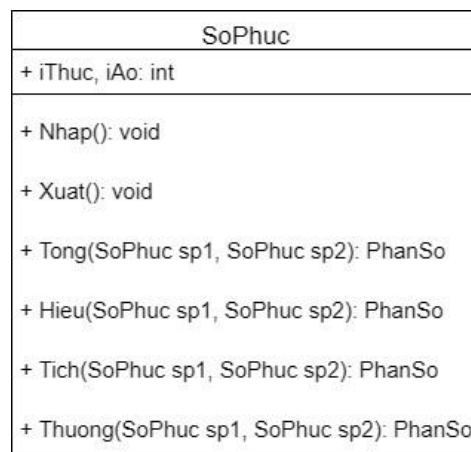
Bảng 2.1.	Thực hiện xây dựng lớp, khai báo các thuộc tính, phương thức của lớp PhanSo.	8
Bảng 3.1.	Xây dựng phương thức Nhap()	8
Bảng 3.2.	Xây dựng phương thức Xuat()	8
Bảng 3.3.	Xây dựng phương thức Tong().....	8
Bảng 3.4.	Xây dựng phương thức Hieu()	9
Bảng 3.5.	Xây dựng phương thức Tich().....	9
Bảng 3.6.	Xây dựng phương thức Thuong()	9
Bảng 3.7.	Xây dựng hàm main().....	10

DANH MỤC HÌNH ẢNH

Hình 1.1.	Class diagram của lớp SoPhuc.....	6
Hình 1.2.	Thực hiện xây dựng lớp, khai báo các thuộc tính, phương thức của lớp PhanSo.....	7
Hình 2.1.	Thực hiện xây dựng lớp, khai báo các thuộc tính, phương thức của lớp PhanSo.....	8
Hình 4.1	Màn hình in ra kết quả bài toán.....	12

NỘI DUNG BÀI LÀM**1. Xây dựng lớp số phức:**

- Thuộc tính: iThuc, iAo
- Phương thức: Nhap(), Xuat(), Tong(), Hieu(), Tich(), Thuong()
- Yêu cầu: Thực hiện xây dựng lớp, vẽ class diagram và khai báo các thuộc tính, phương thức. Viết nội dung vào các phương thức đã khai báo. Gọi các phương thức trong hàm main.
- Class diagram của lớp SoPhuc:



Hình 1.1. Class diagram của lớp SoPhuc.

IT002 – LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

- Thực hiện xây dựng lớp, vẽ class diagram và khai báo các thuộc tính, phương thức của lớp SoPhuc

```
TH > BT_TH3 > h SoPhuc > ...
1  #ifndef SOPHUC_H
2  #define SOPHUC_H
3  #pragma once
4  class SoPhuc {
5  private:
6      float iThuc;
7      float iAo;
8
9  public:
10     void Nhap();
11     void Xuat();
12     SoPhuc Tong(SoPhuc sp1, SoPhuc sp2);
13     SoPhuc Hieu(SoPhuc sp1, SoPhuc sp2);
14     SoPhuc Tich(SoPhuc sp1, SoPhuc sp2);
15     SoPhuc Thuong(SoPhuc sp1, SoPhuc sp2);
16 };
17
18 #endif
```

Hình 1.2. Thực hiện xây dựng lớp, khai báo các thuộc tính, phương thức của lớp PhanSo.

2. Xây dựng các thuộc tính:

```
TH > BT_TH3 > h SoPhuc > ...
1  #ifndef SOPHUC_H
2  #define SOPHUC_H
3  #pragma once
4  class SoPhuc {
5  private:
6      float iThuc;
7      float iAo;
8
9  public:
10     void Nhap();
11     void Xuat();
12     SoPhuc Tong(SoPhuc sp1, SoPhuc sp2);
13     SoPhuc Hieu(SoPhuc sp1, SoPhuc sp2);
14     SoPhuc Tich(SoPhuc sp1, SoPhuc sp2);
15     SoPhuc Thuong(SoPhuc sp1, SoPhuc sp2);
16 };
17
18 #endif
```

3. Xây dựng các phương thức:

3.1. Phương thức Nhap():

```
6 void SoPhuc::Nhap() {  
7     cout << "Nhap phan thuc: ";  
8     cin >> iThuc;  
9     cout << "Nhap phan ao: ";  
10    cin >> iAo;  
11 }
```

Bảng 3.1. Xây dựng phương thức Nhap()

3.2. Phương thức Xuat():

```
13 void SoPhuc::Xuat() {  
14     if (iAo<0) cout<<iThuc << "-" << iAo << "i" << endl;  
15     else cout << iThuc << " + " << iAo << "i" << endl;  
16 }
```

Bảng 3.2. Xây dựng phương thức Xuat()

3.3. Phương thức Tong():

```
18 SoPhuc SoPhuc::Tong(SoPhuc sp1, SoPhuc sp2) {  
19     SoPhuc tong;  
20     tong.iThuc = sp1.iThuc + sp2.iThuc;  
21     tong.iAo = sp1.iAo + sp2.iAo;  
22     return tong;  
23 }
```

Bảng 3.3. Xây dựng phương thức Tong()

3.4. Phương thức Hieu():

```
25 SoPhuc SoPhuc::Hieu(SoPhuc sp1, SoPhuc sp2) {  
26     SoPhuc hieu;  
27     hieu.iThuc = sp1.iThuc - sp2.iThuc;  
28     hieu.iAo = sp1.iAo - sp2.iAo;  
29     return hieu;  
30 }
```


Bảng 3.4. Xây dựng phương thức Hieu()

3.5. Phương thức Tich():

```

32  SoPhuc SoPhuc::Tich(SoPhuc sp1, SoPhuc sp2) {
33      SoPhuc tich;
34      tich.iThuc = sp1.iThuc * sp2.iThuc - sp1.iAo * sp2.iAo;
35      tich.iAo = sp1.iThuc * sp2.iAo + sp1.iAo * sp2.iThuc;
36      return tich;
37  }

```

Bảng 3.5. Xây dựng phương thức Tich()

3.6. Phương thức Thuong():

```

38  SoPhuc SoPhuc::Thuong(SoPhuc sp1, SoPhuc sp2) {
39      SoPhuc thuong;
40      float mau = sp2.iThuc * sp2.iThuc + sp2.iAo * sp2.iAo;
41      thuong.iThuc = (sp1.iThuc * sp2.iThuc + sp1.iAo * sp2.iAo) / mau;
42      thuong.iAo = (sp1.iAo * sp2.iThuc - sp1.iThuc * sp2.iAo) / mau;
43      return thuong;
44  }
45  }

```

Bảng 3.6. Xây dựng phương thức Thuong()

3.7. Hàm main():

```

3
4
5
6 int main() {
7     SoPhuc sp1, sp2, kq;
8
9     cout << "Nhap so phuc thu nhat: " << endl;
10    sp1.Nhap();
11    cout << "Nhap so phuc thu hai: " << endl;
12    sp2.Nhap();
13
14    cout << "\nCac phép toán trên số phức:" << endl;
15
16    kq = kq.Tong(sp1, sp2);
17    cout << "Tong: ";
18    kq.Xuat();
19
20    kq = kq.Hieu(sp1, sp2);
21    cout << "Hieu: ";
22    kq.Xuat();
23
24    kq = kq.Tich(sp1, sp2);
25    cout << "Tich: ";
26    kq.Xuat();
27
28    kq = kq.Thuong(sp1, sp2);
29    cout << "Thuong: ";
30    kq.Xuat();
31
32    return 0;
33 }

```

Bảng 3.7. Xây dựng hàm main().

3. Giải quyết vấn đề:

- Input (đầu vào): Nhập vào phần thực và phần ảo của số phức thứ nhất và số phức thứ 2.
- Output (đầu ra): Các phép tính tổng, hiệu, tích, thương của hai số phức.
- Hướng giải quyết: Cho 2 số phức $a+bi$ và $c+di$.

IT002 – LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

+ Với phương thức `Tong()`: cộng các phần thực và phần ảo tương ứng với nhau. Cụ thể, cho hai số phức là $(a+bi)$ và $(c+di)$, tổng của chúng sẽ là $(a+c)+(b+d)i$.

+ Với phương thức `Hieu()`: trừ phần thực và phần ảo tương ứng với nhau. Cụ thể, cho hai số phức là $(a+bi)$ và $(c+di)$, tổng của chúng sẽ là $(a-c)+(b-d)i$.

+ Với phương thức `Tich()`:

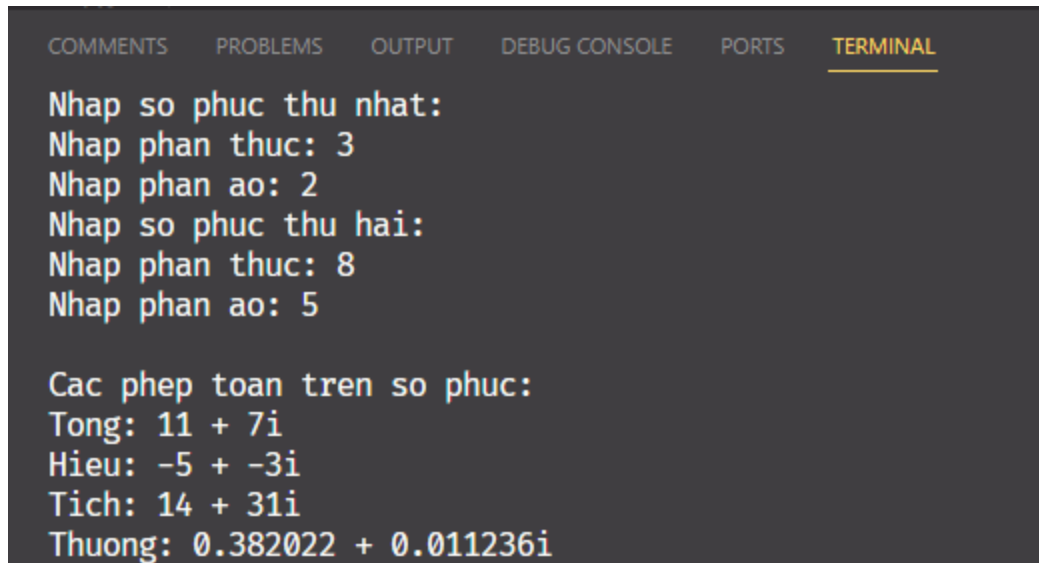
- Phần thực của số phức kết quả sẽ là tích của hai phần thực ban đầu trừ tích của phần ảo ban đầu: $(a \times c) - (b \times d)$.
- Phần ảo của số phức kết quả sẽ là tổng của tích của phần thực của số phức đầu tiên với phần ảo của số phức thứ hai, và tích của phần ảo của số phức đầu tiên với phần thực của số phức thứ hai: $(a \times d) + (b \times c)$.

+ Với phương thức `Thuong()`:

- Tính **mẫu số** của phép chia, tức là bình phương của phần thực của số phức thứ hai cộng với bình phương của phần ảo của số phức thứ hai: $(c^2 + d^2)$. Nếu mẫu số này bằng 0, phép chia không xác định (vì không thể chia cho 0).
- Tính tử số của phép chia và chia cho mẫu số:
 - **Phần thực** của tử số sẽ là tích của phần thực của số phức đầu tiên với phần thực của số phức thứ hai, cộng với tích của phần ảo của số phức đầu tiên với phần ảo của số phức thứ hai: $(a \times c) + (b \times d)$.
 - **Phần ảo** của tử số sẽ là tích của phần ảo của số phức đầu tiên với phần thực của số phức thứ hai, trừ tích của phần thực của số phức đầu tiên với phần ảo của số phức thứ hai: $(b \times c) - (a \times d)$.
- **Hàm main**: Gọi các phương thức của class `SoPhuc` để thực hiện phép tính và in ra kết quả trên màn hình.

4. Thực hiện chạy chương trình:

- Thực hiện các phép tính giữa 2 số phức $(3+2i)$ và $(8+5i)$



```
COMMENTS  PROBLEMS  OUTPUT  DEBUG CONSOLE  PORTS  TERMINAL

Nhap so phuc thu nhat:
Nhap phan thuc: 3
Nhap phan ao: 2
Nhap so phuc thu hai:
Nhap phan thuc: 8
Nhap phan ao: 5

Cac phép toán trên số phức:
Tong: 11 + 7i
Hieu: -5 + -3i
Tich: 14 + 31i
Thuong: 0.382022 + 0.011236i
```

Hình 4.1 Màn hình in ra kết quả bài toán

