

Tuần 02: Hàm dựng, Hàm hủy và Nạp chồng toán tử

I. Bài tập thực hành

Bài tập 1: Sử dụng Class Fraction đã cài đặt ở tuần trước và cài đặt thêm các phương thức, toán tử cần thiết để hàm main() bên dưới hoạt động.

```
void main()
{
    Fraction f1, f2; // 0/1
    Fraction f3(1, -7); // -1/7
    Fraction f4(f3); // Copy constructor
    Fraction f5 = f2; // Copy constructor
    Fraction f6, f7, f8; // Default constructor
    f6 = f3; // Operator =
    f7 = f1 + f5;
    f8 = f2 - f4;
    f3 = f1 * f7;
    f5 = f6 / f2; // Cannot divide by zero!
    if (f2 == f3)
        cout << "f2==f3" << endl;
    if (f3 != f1)
        cout << "f3!=f1" << endl;
    if (f2 >= f5)
        cout << "f2>=f5" << endl;
    if (f2 > f5)
        cout << "f2>f5" << endl;
    if (f5 <= f3)
        cout << "f5<=f3" << endl;
    if (f5 < f3)
        cout << "f5<f3" << endl;
    f1 = f2 + 3;
    f3 = -7 + f1;
    f5 = 7 * f3;
    f6 = f4 - 6;
    cout << f3 << endl;
    cout << f6 << endl;
    f1 += f5;
    f6 -= f7;
    f8 *= f1;
    f8 /= f2; // Cannot divide by zero!
    cout << f8++ << endl;
    cout << ++f7 << endl;
    cout << f8-- << endl;
    cout << --f7 << endl;
    f3 += Fraction(11, 2);
    int x = int(f3); // 3/2 => 1
    float f = (float)f3; // 3/2 => 1.5
    cout << x << endl;
    cout << f << endl;
}
```

Kết quả trên màn hình :

```
Cannot divide by zero!  
f2==f3  
f2>=f5  
f2>f5  
f5<=f3  
f5<f3  
-4  
-43/7  
Cannot divide by zero!  
25/-7  
1  
18/-7  
0  
1
```

Lưu ý: Khi chia cho 0 thì báo lỗi và giữ nguyên giá trị ban đầu

Tài liệu tham khảo Operator Overloading của thầy Hồ Tuấn Thanh:

https://drive.google.com/file/d/1m6axiDM5xH3YeEq1cAxdGBFZx_IejDa/view?usp=share_link

Bài tập 2: Định nghĩa và cài đặt class Stack với các yêu cầu sau:

- Thuộc tính:
 - int* arr
 - int topIndex
 - int size
- Phương thức:
 - Constructors, Destructor
 - Kiểm tra Stack có rỗng không? Kiểm tra Stack có full hay chưa?
 - Các thao tác push, pop, getTopValue
- Không bị memory leak

Bài tập 3: Định nghĩa và cài đặt class Queue với các yêu cầu sau:

- Thuộc tính:
 - int* arr
 - int frontIndex, rearIndex
 - int size
- Phương thức:
 - Constructors, Destructor

- Kiểm tra Queue có rỗng không? Kiểm tra Queue có full hay chưa?
- Các thao tác enqueue, dequeue, getFrontValue
- Không bị memory leak

Bài tập 4: Cho class Student như sau

```
class Student
{
private:
    char * studentId;
    double gpa;
    char * fullname;
    char * address;

};
```

Hãy cài đặt thêm các phương thức:

- Default constructor và Full parameter constructor
- Copy constructor
- Destructor
- Toán tử <<, >>

Yêu cầu khác:

- Các Class cần được tách ra 2 phần:
 - Phần định nghĩa thuộc tính và phương thức ở file .h
 - Phần cài đặt nằm ở file .cpp
- Hàm main nằm ở file Main.cpp, chứa các luồng demo các yêu cầu của từng bài tập như mẫu bên dưới

```
//File Main.cpp nằm Source File. Chứa hàm main

void main()
{
    //Bài 1
    ...

    //Bài 2
    ...

    //Bài 3
    ...
}
```

Các trường hợp không tuân thủ sẽ bị trừ từ 5%-20% số điểm tùy theo mức độ.

II. Nộp bài

Trên lớp: Hoàn thành ở lớp: ít nhất 1 bài tập

Về nhà: Hoàn thành cả 4 bài tập.

Tổ chức thư mục theo cấu trúc sau:

- **Source code:** Thư mục chứa mã nguồn. Cần xóa các tập tin trung gian của quá trình biên dịch cho nhẹ bớt (Build > Clean solution và xóa đi thư mục ẩn .vs)
- **Release:** Thư mục chứa tập tin thực thi .exe biên dịch ra từ mã nguồn.
- **Other:** Thư mục chứa các tài liệu nộp thêm ví dụ ảnh sơ đồ lớp.

Nén tất cả ở dạng <MSSV>_Week<X>_<Y>.zip/rar.

- **MSSV:** mã số sinh viên
- **X:** mã tuần
- **Y:** số bài đã làm
- Ví dụ: **21127001_Week02_4.zip**