



TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN TP.HCM
KHOA CÔNG NGHỆ THÔNG TIN
BỘ MÔN CÔNG NGHỆ PHẦN MỀM
HỆ CHÍNH QUY
MÔN: **LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG**

HƯỚNG DẪN THỰC HÀNH TUẦN 4

NẠP CHỒNG TOÁN TỬ

✚ NGUYỄN LÊ HOÀNG DŨNG

TP.HCM, ngày 9 tháng 10 năm 2017

MỤC LỤC

1	Yêu cầu bài tập.....	3
2	Hướng dẫn, cài đặt mẫu	3
3	Bài tập	7

1 Yêu cầu bài tập

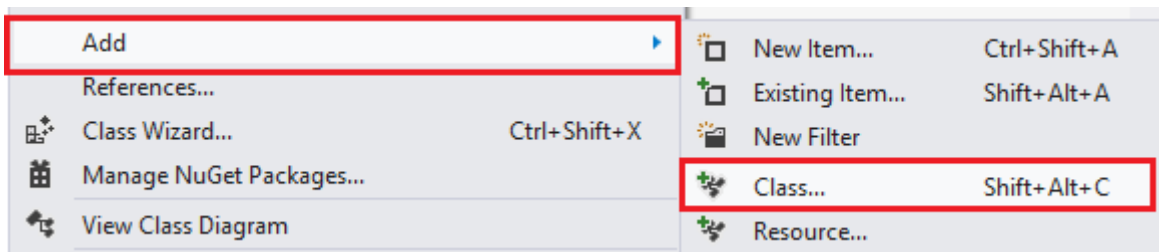
Cài đặt lớp **PHANSO** với các yêu cầu sau:

- Phương thức khởi tạo mặc định, khởi tạo có tham số, khởi tạo sao chép
- Phương thức hủy
- Phương thức getter/setter cho tất cả các thuộc tính
- Phương thức nhập, xuất
- Các toán tử $+$, $-$, $*$, $/$, $+=$, $-=$, $*=$, $/=$
- Toán tử gán $=$
- Toán tử nghịch đảo $!$
- Toán tử rút gọn \sim
- Toán tử so sánh $==$, $>$, $<$, $>=$, $<=$
- Toán tử ép kiểu *float()*
- Toán tử nhập, xuất: $>>$ và $<<$
- Các phương thức cần thiết trong hàm *main()* để sử dụng lớp **PHANSO**

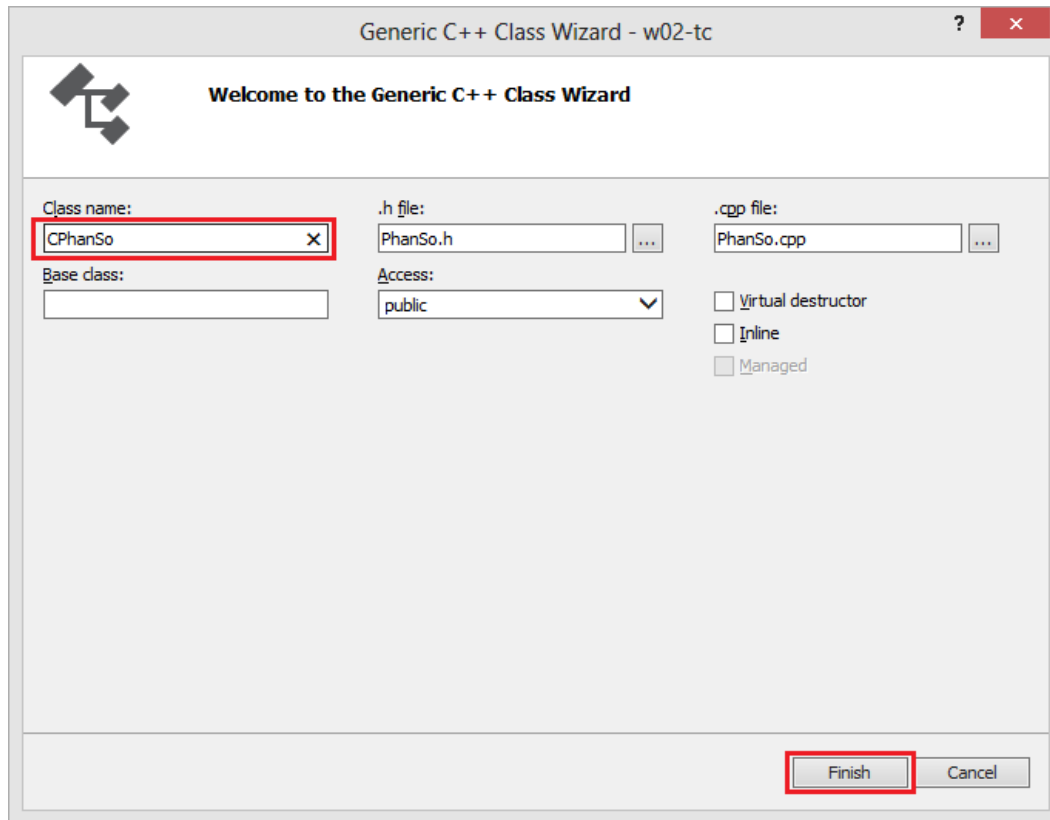
2 Hướng dẫn, cài đặt mẫu

Tạo project

- Tạo **Empty Project** và chọn phải trên project để thêm class vào project như sau:



- Đặt tên class là CphanSo



Cài đặt giao diện trong tập tin "PhanSo.h"

```
#pragma once

#include "iostream"
#include "string"
using namespace std;

class CPhanSo
{
private:
    int _tu;
    int _mau;
public:
    //Phuong thuc khoi tao
    CPhanSo(void);
    CPhanSo(int t, int m);
    CPhanSo(const CPhanSo&ps);
    //Phuong thuc huy
    ~CPhanSo(void);
    //Cac getter/setter
    int getTu();
    void setTu(int t);
    int getMau();
    void setMau(int m);
    //Phuong thuc nhap,xuat
    void Nhap();
    void Xuat();
```

```
//Cac toan tu +, -, *, /
CPhanSo operator+(const CPhanSo &ps);
//Toan tu gan bang
CPhanSo& operator=(const CPhanSo &ps);
//Toan tu so sanh
bool operator==(const CPhanSo &ps);
bool operator!=(const CPhanSo&ps);
//Toan tu ep kieu
operator float();
//Toan tu nhap,xuat >> và <<
friend ostream& operator<<(ostream &os,const CPhanSo &ps);
friend istream& operator>>(istream &is, CPhanSo &ps);
//Toan tu nghịch dao, rut gon
CPhanSo& operator!();
CPhanSo& operator~();
};
```

Cài đặt lớp trong tập tin PhanSo.cpp

```
#include "PhanSo.h"

//Cai dat cac pt khoi tao
CPhanSo::CPhanSo(void)
{
    _tu=0;
    _mau = 1;
}
CPhanSo::CPhanSo(int t, int m)
{
    _tu=t;
    _mau = m;
}
CPhanSo::CPhanSo(const CPhanSo &ps)
{
    _tu=ps._tu;
    _mau = ps._mau;
}
CPhanSo::~~CPhanSo(void)
{
    return;
}
// Cai dat cac pt getter/setter
int CPhanSo::getTu()
{
    return _tu;
}
int CPhanSo::getMau()
{
    return _mau;
}
void CPhanSo::setTu(int t)
{
    _tu=t;
}
void CPhanSo::setMau(int m)
{
    _mau=m;
}
```

```

//Cai dat pt nhap,xuat
void CPhanSo::Nhap()
{
    cout<<"Nhap tu so: ";
    cin>>_tu;
    cout<<"Nhap mau so: ";
    cin>>_mau;
}
void CPhanSo::Xuat()
{
    cout<<_tu<<"/"<<_mau<<endl;
}
//Cat dat phuong thuc +,-,*,/
CPhanSo CPhanSo::operator+(const CPhanSo&ps)
{
    CPhanSo pso;
    pso._tu=this->_tu*ps._mau+this->_mau*ps._tu;
    pso._mau=this->_mau*ps._mau;
    return pso;
}
CPhanSo& CPhanSo::operator=(const CPhanSo&ps)
{
    //...
}
bool CPhanSo::operator==(const CPhanSo&ps)
{
    float ps1 = 1.0*this->_tu/this->_mau;
    float ps2= 1.0*ps._tu/ps._mau;

    return (ps1==ps2);
}
bool CPhanSo::operator!=(const CPhanSo&ps)
{
    if(*this==ps)
        return false;

    return true;
}
//Toan tu nhap,xuat >> vaf <<
ostream& operator<<(ostream &os,const CPhanSo &ps)
{
    cout<<ps._tu<<"/"<<ps._mau<<endl;
    return os;
}
istream& operator>>(istream &is, CPhanSo &ps)
{
    cout<<"Nhap tu so: ";
    is>>ps._tu;
    cout<<"Nhap mau so: ";
    is>>ps._mau;

    return is;
}
//Toan tu ep kieu
CPhanSo::operator float()
{
    return (float)_tu/_mau;
}

```

```
//Toan tu nghich dao
CPhanSo& CPhanSo::operator!()
{
    int temp = this->_tu;
    this->_tu=this->_mau;
    this->_mau=temp;
    return (*this);
}
int UCLN (CPhanSo ps)
{
    int tu = abs(ps.getTu());
    int mau = abs(ps.getMau());
    while(tu!=mau)
    {
        if(tu>mau)
            tu-=mau;
        else
            mau-=tu;
    }
    return tu;
}
//Toan tu Rut gon
CPhanSo& CPhanSo::operator~()
{
    this->_tu=this->_tu/ UCLN(*this);
    this->_mau=this->_mau/ UCLN(*this);
    return (*this);
}
```

Chú ý: sinh viên tự cài đặt các phương thức còn lại theo yêu cầu

3 Bài tập

Xây dựng các lớp đối tượng sau:

- Lớp đơn thức (**CDonThuc**) dạng (ax^b)
 - Các phương thức khởi tạo và phương thức hủy.
 - Phương thức getter/setter cho tất cả các thuộc tính.
 - Phương thức nhập, xuất.
 - Toán tử nhập, xuất: >> và <<
 - Phương thức tính giá trị đơn thức với biến x cho trước
 - Các toán tử *, /, *=, /=
 - Toán tử gán =
 - Toán tử so sánh ==, !=
- Lớp chuỗi ký tự (**CString**) sử dụng mảng ký tự được cấp phát động để lưu trữ (có thể sử dụng các hàm trong thư viện string.h), hỗ trợ một số phương thức cơ bản như:
 - Các phương thức tạo lập và hủy có thể có.

- Lấy độ dài chuỗi.
- Lấy/Gán từ ký tự trong chuỗi.
- Các toán tử cơ bản:
 - Nhập/Xuất: >>, <<
 - Nối chuỗi: +
 - Đảo chuỗi: !
 - Chuẩn hóa chuỗi (bỏ đi các khoảng trắng thừa): ~
 - So sánh với chuỗi khác: >, >=, <, <=, ==, !=