

## แบบฝึกหัดที่ 22

---

1. จงติดตามการทำงานของโปรแกรมว่าได้ผลลัพธ์อะไรบ้าง

```
#include<iostream>
using namespace std;

class A {
    int a;
    public:
        static int b;
        A() { a=1; b++; cout<<"A"<<endl; }
        A(int x) { a=x+1; b++; }
        A(A &y) { a=y.b; cout<<"Wow"<<endl;}
        A f(A y) { a+=y.b; b++; return *this; }
        ~A() { cout<<a<<endl; }
        void set(int i) { a=i; }
        int get() { return a; }
};

int A::b;

class B {
    A x;
    public:
        B() { x.set(5); };
        A f() { A y(2); y.set(x.get()); return y; }
};

void main() {
    A a;
    A c(3);
    B b;
    b.f().f(c);
    cout<<A::b<<endl;
}
```

## 2. โปรแกรมข้างล่างนี้มีที่ผิดตรงไหนบ้าง

```
#include<iostream>
using namespace std;
class A
{
    int x;
protected:
    int a;
public:
    A() { x=2; a=3; }
    void f() { x=x+a; }
};
class B: virtual private A
{
    int x;
protected:
    int b;
public:
    B() { x=1; b=a; }
    void g() { x=a+b; }
};
class C: virtual protected A
{
    int x;
protected:
    int c;
public:
    C() { x=a; c=b; }
    void h() { c=a++; }
    void hh() { f(); }
};
class D: public B, public C
{
    int x;
protected:
    int d;
public:
    D() { x=b+c; d=a; }
    void k() { g(); }

    void m() { f(); }
    void n() { hh(); }
};
class F: protected B
{
    int x;
public:
    F() { x=a; }
    void p() { x=b; }
    void q() { f(); }
    void r() { g(); }
};
class G: private F
{
    int x;
public:
    G() { x=a; }
    void s() { g(); }
};
void main()
{
    F f;
    f.r();
    f.g();
    f.f();
    D d;
    d.k();
    d.h();
    d.g();
    d.f();
    B b;
    b.g();
    b.f();
    C c;
    c.f();
    G g;
    g.g();
}
```

### 3. จงติดตามการทำงานของโปรแกรมว่าได้ผลลัพธ์อะไรบ้าง

```
#include<iostream>
using namespace std;
class A {
    public:
    A() { cout<<"A"<<endl; }
    virtual ~A() {cout<<"a"<<endl;}
    int f() { return 11; }
    virtual int g() { return 1; }
};
class B: virtual public A {
    public:
    B() { cout<<"B"<<endl; }
    ~B() { cout<<"b"<<endl; }
    int f() { return 22; }
    int g() { return 2; }
};
class C: virtual public A {
    static int c;
    public:
    C() { c=0; cout<<"C"<<endl; }
    ~C() { cout<<c<<endl; }
    int f() { return 33; }
    int g() { return ++c; }
};
int C::c;
class D: public B {
    int d;
    public:
    D() { d=0; cout<<"D"<<endl; }
    ~D() { cout<<"d"<<endl; }
    int f() { return 44; }
    int g() { return ++d; }
};
class E: public D {
    public:
    E() { cout<<"E"<<endl; }
    ~E() { cout<<"e"<<endl; }
    int f() { return 55; }
    int g() { return 5; }
};
```

```

class F: public B, public C {
public:
    F() { cout<<"F"<<endl; }
    ~F() { cout<<"f"<<endl; }
    int f() { return 66; }
    int g() { return 6; }
};

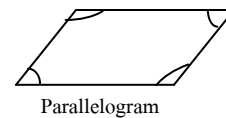
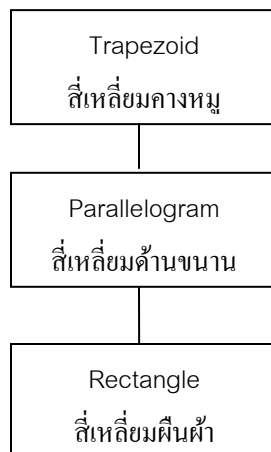
void main() {
    A *a = new F;
    A *b = new E;
    if (a->f() == b->f())
        cout<<((a->g()==b->g())? "pretty" : "lovely")<<endl;
    else
        cout<<"good"<<endl;

    C c;
    c.g();
    cout<<c.g()<<endl;

    delete a;
    delete b;
}

```

4. จงสร้างคลาส Point, คลาส Trapezoid, คลาส Parallelogram, และคลาส Rectangle โดยทั้ง 4 คลาสมีความสัมพันธ์กันดังนี้



ด้านตรงข้ามขนานกันและยาวเท่ากัน  
มุมตรงข้ามเท่ากัน

คลาส Point ประกอบด้วย โคออร์ดิเนต x และ y

คลาส Trapezoid ประกอบด้วย w1, w2 เป็นความยาวด้านทั้งสอง(ตามรูป)  
p1 เป็นอ็อบเจกต์ของคลาสPoint ซึ่งเป็นจุดอยู่ที่มุม a  
p2 เป็นอ็อบเจกต์ของคลาสPoint ซึ่งเป็นจุดอยู่ที่มุม b

คลาส Parallelogram สืบทอดมาจาก Trapezoid โดยที่ ด้าน w1=w2

คลาส Rectangle สืบทอดมาจาก Parallelogram โดยที่ทุกมุมมีค่าเท่ากับ 90

ข้อมูลทั้งหมดให้เก็บไว้ในส่วนของ private ส่วนฟังก์ชันให้เก็บไว้ที่ public

คลาส Point

1. จงสร้างฟังก์ชันในการหาระยะทางระหว่างจุดสองจุดโดยใช้หลักการ function overloading
2. ให้สร้างข้อมูลและฟังก์ชันตามความเหมาะสมเพื่อใช้ตอบคำถามต่างๆได้

คลาสสี่เหลี่ยมทั้ง 3 คลาส

- จงสร้างฟังก์ชัน set ในการกำหนดค่าให้กับข้อมูลทั้งหมดให้ถูกต้องโดยใช้หลักการ function overiding โดยที่ Trapezoid ให้รับค่า w1 w2 p1 p2 ส่วน Parallelogram ให้รับค่า w1 p1 p2 ส่วนคลาส Rectangle ให้รับค่า w1 p1 และความสูง h
- จงสร้างดีฟอลต์คอนสตรัคเตอร์ คอนสตรัคเตอร์โอเวอร์โหลดดิง ดีสตรัคเตอร์ และก๊อปปี้คอนสตรัคเตอร์ โดยกำหนดค่าตามเงื่อนไขที่ระบุไว้ข้างต้น
- ฟังก์ชัน get ในการรีเทิร์นค่าของข้อมูลแต่ละตัว โดยต้องสอดคล้องกับเงื่อนไขที่ระบุไว้ข้างต้น (subclass สามารถใช้ get ของ superclass ได้)
- จงสร้างฟังก์ชันในการหาพื้นที่ของสี่เหลี่ยม โดยใช้หลักการ function overiding

$$\text{Trapezoid Area} = \frac{w1+w2}{2} \times h$$

ส่วน Parallelogram และ Rectangle มีพ.ท. = w1 x h

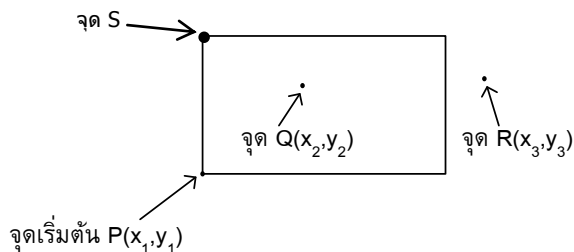
- จงสร้างฟังก์ชันในการหาความยาวของเส้นรอบรูปสี่เหลี่ยม (subclass สามารถเรียกใช้ฟังก์ชันจาก superclass ได้)
- operator << ในการแสดงค่าที่รับมาจากฟังก์ชัน set ของแต่ละคลาสออกทางจอภาพ โดยใช้หลักการ dynamic binding

จงสร้างฟังก์ชันในการนับจำนวนอ็อบเจกต์ของคลาส Rectangle กรณีที่ไม่มีอ็อบเจกต์ของคลาสนี้อยู่เลยก็ต้องสามารถบอกได้ด้วย

จงสร้างเฟรนด์ฟังก์ชันในการตรวจสอบว่าจุดใดๆ ซึ่งเป็น อ็อบเจกต์ของคลาส Point อยู่ในสี่เหลี่ยมซึ่งเป็นอ็อบเจกต์ของคลาส Rectangle หรือไม่ (เช่น จุด Q อยู่ในสี่เหลี่ยม และจุด R ไม่อยู่ในสี่เหลี่ยม)

และถ้าจุดนั้นเป็นจุดที่อยู่ในสี่เหลี่ยม (เช่นจุด Q) ให้สร้างอ็อบเจกต์ของ Rectangle ขึ้นมาใหม่โดยให้จุด Q เป็นจุด p1 และกำหนดให้ความยาวมีค่าเท่ากับระยะทางจากจุด Q ไปยังจุด P และกำหนดให้ความสูงมีค่าเท่ากับระยะทางจากจุด Q ไปยังจุด S และทำการรีเทิร์น อ็อบเจกต์ใหม่ที่มีข้อมูลครบถ้วนออกจากฟังก์ชัน

แต่ถ้าจุดนั้นไม่อยู่ในสี่เหลี่ยม (เช่นจุด R) ให้แทนที่จุด p1 ด้วยจุด R พร้อมทั้งแก้ไขข้อมูลที่เกี่ยวข้องให้ถูกต้อง และรีเทิร์นอ็อบเจกต์เดิมที่มีการย้ายจุดเริ่มต้นไปยังจุด R ออกจากฟังก์ชัน



จงสร้าง operator - ในการลบ Rectangle ซึ่งจะได้ Rectangle ใหม่ที่มีความยาวด้าน w1 เท่ากับผลต่างของความยาวด้าน w1 ของทั้ง 2 Rectangle และมีความสูงเท่ากับผลต่างของความสูงของทั้ง 2 Rectangle ถ้าผลต่างเป็น 0 ให้กำหนดค่าเป็น 1 และให้กำหนดค่า p1 มีค่าเท่ากับจุดกึ่งกลางของ p1 ของทั้ง 2 Rectangle และให้รีเทิร์น Rectangle ออกจากฟังก์ชัน โดยมีการกำหนดค่าทุกค่าอย่างถูกต้องด้วย

ใน main

1. จงสร้าง array ของ object ของคลาส Rectangle ให้มีขนาดเท่ากับ 4 และมีการกำหนดค่าเริ่มต้นโดยใช้ฟังก์ชัน set และให้แสดงค่าที่อยู่ใน array ทั้งหมดโดยใช้ฟังก์ชัน get
2. จากข้อ 1 จงสร้าง object ให้ไปเรียกใช้ฟังก์ชันในการหาพื้นที่สี่เหลี่ยมโดยใช้หลักการ dynamic binding
3. จงหาผลรวมของพื้นที่ของสี่เหลี่ยมที่เก็บอยู่ใน array ในข้อ 1 และแสดงผลรวมออกทางจอภาพ
4. จงสร้าง array ของ object ของคลาส Parallelogram ให้มีขนาดเท่ากับ 3 และมีการกำหนดค่าเริ่มต้นให้เรียบร้อย จากนั้นให้เปรียบเทียบค่าความยาวรอบรูปของ Parallelogram แต่ละรูปว่ารูปใดมีความยาวรอบรูปมากที่สุด จากนั้นให้แสดงผลลัพธ์ว่าเป็น object ใดและมีความยาวเท่าไร

(หมายเหตุ: ทุกๆ object จะต้องถูกสร้างและถูกทำลายได้อย่างถูกต้อง และน.ศ. สามารถสร้าง attribute และ function เพิ่มเติมได้ตามความเหมาะสม)