

แบบฝึกหัดที่ 2

lab - 2

2.1 จงสร้างคลาส Point ซึ่งประกอบด้วยจุดโดยอธิบาย (x, y) เป็นพิกนิยม และชื่อของจุด (name) เช่น จุดชื่อ "mypoint" มีค่า x=2 และ y=3 เป็นต้น และจงสร้างตัวฟอล์กอนสตัตวัตเตอร์ ค่อนสตัตวัตเตอร์ไว้เรียบร้อยแล้วดัง ตัวอย่างนี้
ฟังก์ชัน set ในการกำหนดค่า x, y, name ฟังก์ชัน getX ในการ return ค่า x ฟังก์ชัน getY ในการ return ค่า y ฟังก์ชัน show ในการแสดงค่า x, y และ name ออกทางจอภาพ

static Func

✓ 2.2 จากข้อ 2.1 จงสร้างฟังก์ชันในการนับจำนวนอีบเจกต์ของคลาส Point ในกรณีที่ไม่มีอีบเจกต์ถูกสร้างขึ้นและยกตัวอย่างสามารถบอกได้ว่ามีจำนวนทั้งหมดเป็นศูนย์

```
#include<iostream>
#include<string.h>
using namespace std;

class Point {
private:
    static int num; → นับ obj
    double x,y;
    string name1;
    char name2[20];
public:
    //destructor
    Point(){
        num++;
        set(0.0, 0.0, "noname");
    }

    Point(double x1 , double y1){
        num++;
        set(x1, y1, "noname");
    }

    Point(char * m){
        num++;
        set(0.0, 0.0, m);
    }

    Point(double x1, double y1, char *m){
        num++;
        set(x1, y1, m);
    }

    ~Point(){ → destructor
        num--;
        cout<<"\n----- count Obj = "<<Point::count();
    }

    //set all
    void set(double x1, double y1, char *m){ } set 1111
    { } 9 คุณ構造器
        x = x1;
        y = y1;
        strcpy(name2, m);
    }

    //set
    void setX(double x1){
        x=x1;
    }
    void setY(double y1){
        y=y1;
    }
    void setName1(string n){
        name1=n;
    }
    void setName2(char * m){
        strcpy(name2,m);
    }

    //get
    double getX(){
        return x;
    }
    double getY(){
        return y;
    }
    string getName1(){
        return name1;
    }
    char * getName2(){
        return name2;
    }

    //show
    void show();
};

//static นับเลข
static int count(){
    return num;
}

}; → ฟังก์ชันนี้จะต้องใช้ static ไม่ใช่ member
```

```
void Point::show(){
    cout<<"\n----- get -----";
    cout<<"name : "<<name2<<endl;
    cout<<"X = "<<x<<endl;
    cout<<"Y = "<<y<<endl;
}

main(){
    cout<<"\n----- count Obj = "<<endl;
    cout<<"\n----- count Obj = "<<Point::count()<<endl;
    Point a;

    cout<<"\n----- count Obj = "<<endl;
    cout<<"\n----- count Obj = "<<Point::count()<<endl;
    Point b;
    b.show();

    cout<<"\n----- count Obj = "<<endl;
    cout<<"\n----- count Obj = "<<Point::count()<<endl;
    Point c(2,3);
    c.show();

    cout<<"\n----- count Obj = "<<endl;
    cout<<"\n----- count Obj = "<<Point::count()<<endl;
    Point d("Kingkong");
    d.show();

    cout<<"\n----- count Obj = "<<endl;
    cout<<"\n----- count Obj = "<<Point::count()<<endl;
    Point e(5,7,"GUJABAH");
    e.show();

    a.setX(3.0);
    a.setY(1.5);
    a.setName1("Habibi");

    cout<<"\n----- count Obj = "<<endl;
    cout<<"\n----- count Obj = "<<Point::count()<<endl;
    cout<<"\n----- get -----";
    cout<<"\nx = "<<a.getX()<<endl<<"y = "<<a.getY()<<endl<<"name = "<<a.getName1()<<endl;

    cout<<"\n----- destructor -----";
    cout<<Point::count();
}
```

```
----- get -----
count Obj = 0
-----
count Obj = 1
----- get -----
name : noname
X = 0
Y = 0
-----
count Obj = 2
----- get -----
name : noname
X = 2
Y = 3
-----
count Obj = 3
----- get -----
name : Kingkong
X = 0
Y = 0
-----
count Obj = 4
----- get -----
name : GUJABAH
X = 5
Y = 7
-----
count Obj = 5
----- get -----
x = 3
y = 1.5
name = Habibi
-----
===== destructor =====
5
-----
count Obj = 4
-----
count Obj = 3
-----
count Obj = 2
-----
count Obj = 1
-----
count Obj = 0
-----
count Obj = 4
```

int width int height

2.3 จงสร้างคลาส Rectangle ที่ประกอบด้วยความกว้างและความยาว และจงสร้างคีพออล์ค
คอนสตรัคเตอร์ คอนสตรัคเตอร์โอเวอร์โหลดดัง คีสตรัคเตอร์ พังก์ชันคำนวนพื้นที่สี่เหลี่ยม
พังก์ชันคำนวนความยาวรอบสี่เหลี่ยม รวมทั้งพังก์ชัน set ใน การกำหนดค่าให้กับข้อมูล
ทั้งหมด พังก์ชัน get ในการเรียกค่าของข้อมูลแต่ละด้าน พังก์ชัน show ในการแสดงค่าต่างๆ
ออกทางจอภาพ และพังก์ชันในการนับจำนวนอีกครั้งของคลาส Rectangle

```
#include<iostream>
#include<string.h>
using namespace std;

class Rectangle {
private:
    static int num;
    int width;
    int length;
public:
    //destructor
    Rectangle(){
        num++;
        set(0, 0);
    }
    Rectangle(int a , int b ){
        num++;
        set( a , b );
    }
    ~Rectangle(){
        num--;
        cout<<"\nCount Obj = "<<Rectangle::count()<<endl;
    }
    void set(int a , int b){
        width=a;
        length=b;
    }
    void setWidth(int a){
        width=a;
    }
    void setLength(int b){
        length=b;
    }
    int getWidth(){
        return width;
    }
    int getLength(){
        return length;
    }
    int area(){
        return width*length;
    }
    int perimeter(){
        return 2*(width+length);
    }
}
```

} attribute object

} default con } constructor

→ destructor

→ set |||

} กูบ show

```
void show();
//static ฟังก์ชัน
static int count(){
    return num;
}

void Rectangle::show(){
    cout<<"\n===== Show ====="<<endl;
    cout<<"\nWidth = "<<width<<endl;
    cout<<"Length = "<<length<<endl;
    cout<<"Area of Rectangle = "<<area()<<endl;
    cout<<"Perimeter = "<<perimeter()<<endl;
}

int Rectangle::num;

main(){
    cout<<"\n-----"<<endl;
    cout<<"\nCount Obj = "<<Rectangle::count()<<endl;
    Rectangle r;

    cout<<"\n-----"<<endl;
    cout<<"\nCount Obj = "<<Rectangle::count()<<endl;
    cout<<"\n-----"<<endl;
    Rectangle r1;
    r1.show();
    cout<<"\n===== Get ====="<<endl;
    cout<<"\nWidth = "<<r1.getWidth()<<endl;
    cout<<"Length = "<<r1.getLength()<<endl;
    cout<<"Area of Rectangle = "<<area()<<endl;
    cout<<"Perimeter = "<<perimeter()<<endl;

    cout<<"\n-----"<<endl;
    cout<<"\nCount Obj = "<<Rectangle::count()<<endl;
    cout<<"\n-----"<<endl;
    Rectangle r2(5,6);
    r2.show();
    cout<<"\n===== Get ====="<<endl;
    cout<<"\nWidth = "<<r2.getWidth()<<endl;
    cout<<"Length = "<<r2.getLength()<<endl;
    cout<<"Area of Rectangle = "<<area()<<endl;
    cout<<"Perimeter = "<<perimeter()<<endl;

    cout<<"\n-----"<<endl;
    cout<<"\nCount Obj = "<<Rectangle::count()<<endl;
    cout<<"\n-----"<<endl;
    Rectangle r3;
    r3.setWidth(3);
    r3.setLength(7);
    r3.show();
    cout<<"\n===== Get ====="<<endl;
    cout<<"\nWidth = "<<r3.getWidth()<<endl;
    cout<<"Length = "<<r3.getLength()<<endl;
    cout<<"Area of Rectangle = "<<area()<<endl;
    cout<<"Perimeter = "<<perimeter()<<endl;

    cout<<"\n===== Destructor ====="<<endl;
    cout<<"\nCount Obj = "<<Rectangle::count()<<endl;
}
```

```
Count Obj = 0
-----
Count Obj = 1
===== Show =====
Width = 0
Length = 0
Area of Rectangle = 0
Perimeter = 0
===== Get =====
Width = 0
Count Obj = 2
===== Show =====
Width = 5
Length = 6
Area of Rectangle = 30
Perimeter = 22
===== Get =====
Width = 5
Length = 6
Area of Rectangle = 30
Perimeter = 22
===== Show =====
Width = 3
Length = 7
Area of Rectangle = 21
Perimeter = 20
===== Get =====
Width = 3
Length = 7
Area of Rectangle = 21
Perimeter = 20
===== Destructor =====
Count Obj = 4
Count Obj = 3
Count Obj = 2
Count Obj = 1
Count Obj = 0
```

lab → Rectangle *5A

3.6 จงสร้างคลาส MyRect ในการเก็บข้อมูลออบเจกต์ของคลาส Rectangle (ข้อ 2.3) ในรูปแบบของอาร์เรย์พลวัต 2 มิติ โดยสามารถเปลี่ยนขนาดของแท่งและคอลัมน์ของอาร์เรย์ และจงสร้างฟังก์ชันสครัปเตอร์ คุณสครัปเตอร์ในการทำหนทางด้านการแก้ไขเพื่อให้กับอาร์เรย์ ตัวอย่างเช่นเพื่อเปลี่ยนพื้นที่ที่ต้องไว้ พังก์ชัน reset ในการเปลี่ยนขนาดของอาร์เรย์ รวมทั้งของพื้นที่ใหม่ให้ถูกต้อง พังก์ชัน setRect ในการกำหนดค่าให้กับออบเจกต์ในอาร์เรย์ พังก์ชัน getRect ในการเรียกค่าที่ออบเจกต์ในอาร์เรย์ พังก์ชัน show ในการแสดงข้อมูลต่างๆ ของการจราจร

ก ex>>3-7, 3-6

func

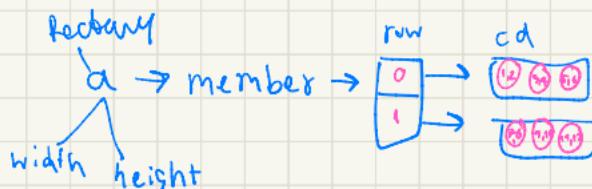
- Set → กำหนดค่าให้ obj ที่อยู่ใน array → เช่นกำหนดค่าใน class Rectangle
 - get → return obj ที่อยู่ใน array
 - show
 - reset
- ↓ เมื่อเขียนแล้ว set ที่เป็น 0

Rectangle rect
rect.setRect(4, 5, 6)
int arr[4][5];

หลักการจัดองค์ประกอบ

Main

• copy class Rectangle ทางจราจร



- myRect → จราจร ตัวอย่าง Rectangle
Rectangle → จราจร
- ผู้ใช้งาน rectangle ที่ set, get

แบบฝึกหัดที่ 4

4.1 จากคลาส Point (ข้อ 2.2) จงเพิ่มฟังก์ชัน dot(Point&) ในการรีเทิร์นค่าผลคูณเชิงสเกลาร์ $(x_1, y_1) \cdot (x_2, y_2) = x_1 \cdot x_2 + y_1 \cdot y_2$ และฟังก์ชัน midpoint(Point&, Point&) ในการคำนวณค่าจุดที่กลางระหว่างจุด 2 จุด

Class Point

dot func

$$\begin{aligned} X_1 &= 1 \\ Y_1 &= 2 \end{aligned}$$

mid func

$$\begin{aligned} X_2 &= 3 \\ Y_2 &= 4 \end{aligned}$$

```
//4.1 dot & midpoint func
double dot(Point&a){
    //((x1 * x2) + (y1 * y2))
    double scalar = (x * a.x) + (y * a.y);
    return scalar;
}

Point midpoint(Point&a, Point&b){ //a & b เป็นตัวแปรในพาราที่เราจะส่งค่ามาจาก main
    //x = ((x1 + x2)/2) y = ((y1 + y2)/2)
    x = ((a.x + b.x)/2);
    y = ((a.y + b.y)/2); //x , y นี้จะ mid
    Point mid(x,y,"MidPoint");
    return mid;
}

//static นับจุด
static int count(){
    return num;
}

int Point::num;//ต้องมีเฉพาะ static variable

void Point::show(){
    cout<<"\n===== show ====="<<endl;
    cout<<"name : "<<name2<<endl;
    cout<<"X = "<<x<<endl;
    cout<<"Y = "<<y<<endl;
}

main(){
    cout<<"\n-----"<<endl;
    cout<<"count Obj = "<<Point::count()<<endl;

    double a,b;
    char name2[20];
    cout<<"\n===== Input ====="<<endl;
    cout<<"Point name : ";cin>>name2;
    cout<<"X1 = ";cin>>a;
    cout<<"Y1 = ";cin>>b;
    Point myPoint;
    myPoint.set(a,b,name2);
    cout<<"\ncount Obj = "<<Point::count()<<endl;
    myPoint.show();

    cout<<"\n===== Input ====="<<endl;
    cout<<"Point name2 : ";cin>>name2;
    cout<<"X2 = ";cin>>a;
    cout<<"Y2 = ";cin>>b;
    Point urPoint;
    urPoint.set(a,b,name2);
    cout<<"\ncount Obj = "<<Point::count()<<endl;
    urPoint.show();

    double result;
    result = myPoint.dot(urPoint);
    cout<<"\n==== Result Scalar ===="<<endl;
    cout<<"Scarlar = "<<result<<endl;

    Point midresult, mid1;
    midresult = mid1.midpoint(myPoint,urPoint);
    midresult.show();
    return 0;
}
```

formula for midpoint

$$x = \frac{x_1 + x_2}{2} \quad | \quad y = \frac{y_1 + y_2}{2}$$

$$\begin{aligned} \text{dot} &= (x_1 \cdot x_2) + (y_1 \cdot y_2) & \text{mid } x &= (1+3)/2 \\ &= (1 \cdot 3) + (2 \cdot 4) & &= 2 \\ &= 3 + 8 & \text{mid } y &= (2+4)/2 \\ &= 11 & &= 3 \\ \therefore \text{mid}(x,y) & & & \end{aligned}$$

```
count Obj = 0

===== Input =====
Point name : m
X1 = 1
Y1 = 2

count Obj = 1

===== show =====
name : m
X = 1
Y = 2

===== Input =====
Point name2 : n
X2 = 3
Y2 = 4

count Obj = 2

===== show =====
name : n
X = 3
Y = 4

===== Result Scalar ====
Scarlar = 11

count Obj = 4

===== show =====
name : MidPoint
X = 2
Y = 3

count Obj = 3

count Obj = 2

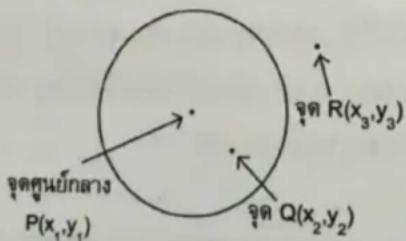
count Obj = 1

count Obj = 0
```

radius & center

5.5 จงสร้างคลาส **Circle** ที่ประกอบด้วยรัศมีและจุดศูนย์กลาง โดยจุดศูนย์กลางเป็นออบเจกต์ของคลาส **Point** จงสร้างเฟรนเดิล์ฟังก์ชันในการตรวจสอบว่าจุดใดๆ ซึ่งเป็นออบเจกต์ของคลาส **Point** อยู่ในวงกลมซึ่งเป็นออบเจกต์ของคลาส **Circle** หรือไม่ (เช่น จุด P เป็นจุดศูนย์กลาง

ของวงกลม จุด Q อยู่ในวงกลม และจุด R ไม่อยู่ในวงกลม) นอกจากนี้จงสร้างคณสต์ว์คเดอร์ โอลเวอร์荷ลดดิ้ง คิสครัฟเตอร์ ฟังก์ชัน **set**, **get** และ **show**



```
//friend function
Friend bool checkCircle(Circle&, Point&);

//static นับเลข
static int count(){
    return num;
}

int Point::num;//ต้องมีเพิ่ลากิ้ง static variable

void Point::show(){
    cout<<"\n===== show ====="<<endl;
    cout<<"Point name : "<<name<<endl;
    cout<<"X = "<<x<<endl;
    cout<<"Y = "<<y<<endl;
}

class Circle
{
private:
    double radius;
    Point center;
    char name[20];
public:

/*ด้วยการเรียก setCircle ด้วยอาร์กิวเม้นต์ 0 ส่าหริน center ซึ่งอาจไม่ถูกต้องหาก Point ไม่รองรับการกำหนดค่าแบบนี้ (ควรเป็นการกำหนดค่าเป็นออบเจกต์ Point มากกว่าค่าดั้งเดิม)
Circle(){
    setCircle(0,0,"noname");
}
*/
//ที่ถูกผิดง ต้องแบบนี้
/*
Circle(){
    Point c(0, 0);
    setCircle(0, c, "noname");
}
*/
/*ด้วยการกำหนดค่า center โดยการเรียก center.setX(0); และ center.setY(0);
ซึ่งสมมติว่า Point มีเมธอดเหล่านี้เพื่อกำหนดค่าพิกัด x และ y*/
Circle():name("noname"){
    radius = 0;
    center.setX(0);
    center.setY(0);
}
```

บรรทัดแรก `circle():name("noname")` เป็นการประกาศตัวสร้าง (constructor) สำหรับคลาส `circle` ซึ่งมีชื่อว่า `noname` โดยทั่วไป ตัวสร้างจะถูกใช้เพื่อกำหนดค่าเริ่มต้นให้กับวัตถุ (object) ของคลาสนั้นๆ ในกรณีนี้ ตัวสร้างกำลังถูกใช้งานเพื่อกำหนดค่าเริ่มต้นให้กับวงกลมที่มีรัศมี (`radius`) เป็น 0 และจุดศูนย์กลาง (`center`) อยู่ที่พิกัด (0, 0)

```
Circle(double r, Point & c){
    setCircle(r, c, "noname");
}

Circle(double r, Point & c, char * n){
    setCircle(r, c, n);
}

~Circle(){
    cout << "delete circle : "<< name<<endl;
}

//set
void setCircle(double r, Point & c, char * n){
    radius=r;
    center=c;
    strcpy(name, n);
}

void setRadius(double r){
    radius=r;
}

void setCenter(Point & c){
    center=c;
}

void setName(char * n){
    strcpy(name, n);
}

//get
double getRadius(){
    return radius;
}

Point getCenter(){
    return center;
}

char* getName(){
    return name;
}

//show
void showCircle(){
    cout<<"\n===== Show Circle ====="<<endl;
    cout<<"Circle Name : "<< name << endl;
    cout <<"X = "<< center.getX()<<endl;
    cout <<"Y = "<< center.getY() << endl;
    cout <<"radius = "<< radius << endl;
}
```

NEXT

```

//friend function
bool checkCircle(Circle & c, Point & p){ //c/x1,y1 รับอุปกรณ์ p/x2,y2 รับจุดที่จะเช็คว่าอยู่ในหรือนอกวงกลม
    double distance;
    distance = sqrt(pow(p.x - c.center.x, 2) + (pow(p.y - c.center.y, 2)));
    //distance = sqrt(pow(p.x - c.getCenter().x, 2) + pow(p.y - c.getCenter().y, 2)); กรณีไม่ใช้friend function
    if(distance <= c.getRadius()){
        return true;
    } else{
        return false;
    }
}

main(){
    //show x,y และกิจกรรมของจุด
    cout<<"count Obj = "<<Point::count()<<endl;
    Point mid(0,0,"noname");
    mid.show();

    //show x,y,radius และกิจกรรมของจุดในวงกลม
    cout<<"-----"<<endl;
    double r;
    cout<<"Input radius : ",cin>>r;
    Circle a(r,mid,"Point A");
    a.showCircle();

    //check point that we wanna know its in circle or not
    Point b(2,5,"Point B");
    b.show();

    bool resultN;
    resultN = checkCircle(a, b);
    if(resultN == 1){
        cout<<"Point "<<b.getName2()<<" is in the circle "<<endl;
    } else {
        cout<<"Point "<<b.getName2()<<" is out side the circle "<<endl;
    }
}

//สร้างจุดศูนย์กลางของวงกลมและจุดที่จะคำนวณ เพื่อศึกษาผลกระทบทาง
cout<<"\n===== Create center point ====="<<endl;
float x, y;
char n[20];
cout<<"Input Point Name : ",cin>>n;
cout<<"Input x1 : ",cin>>x;
cout<<"Input y1 : ",cin>>y;
cout<<"Input radius : ",cin>>r;
Point mid1(x,y);
Circle c(r,mid1,n);
c.showCircle();
cout<<"Obj = "<<Point::count()<<endl;

//อีกจุดเพื่อศึกษา
cout<<"-----"<<endl;
cout<<"Input Point Name : ",cin>>n;
cout<<"Input x1 : ",cin>>x;
cout<<"Input y1 : ",cin>>y;
Point d(x,y,n);
d.show();
cout<<"Obj = "<<Point::count()<<endl;

cout<<"=====Get===== "<<endl;
bool result;
result = checkCircle(c,d);
if(result == 1){
    cout<<"Point "<<d.getName2()<<" is in the circle "<<endl;
} else {
    cout<<"Point "<<d.getName2()<<" is out side the circle "<<endl;
}

cout<<"===== destructor ====="<<endl;
return 0;
}

```

in Circle class

```

count Obj = 0
=====
Point name : noname
X = 0
Y = 0

Input radius : 9
==== Show Circle ====
Circle Name : Point A
X = 0
Y = 0
radius = 9

=====
Point name : Point B
X = 2
Y = 5
distance = 5.38516
Point Point B is in the circle

===== Create center point =====
Input Point Name : kimchi
Input x1 : 2
Input y1 : 6
Input radius : 4

==== Show Circle ====
Circle Name : kimchi
X = 2
Y = 6
radius = 4
Obj = 5

=====
Input Point Name : Habibi
Input x1 : 3
Input y1 : 2

===== show =====
Point name : Habibi
X = 3
Y = 2
Obj = 6
=====Get=====
distance = 4.12311
Point Habibi is out side the circle
===== destructor =====

count Obj = 5
delete circle : kimchi

count Obj = 4

count Obj = 3

count Obj = 2
delete circle : Point A

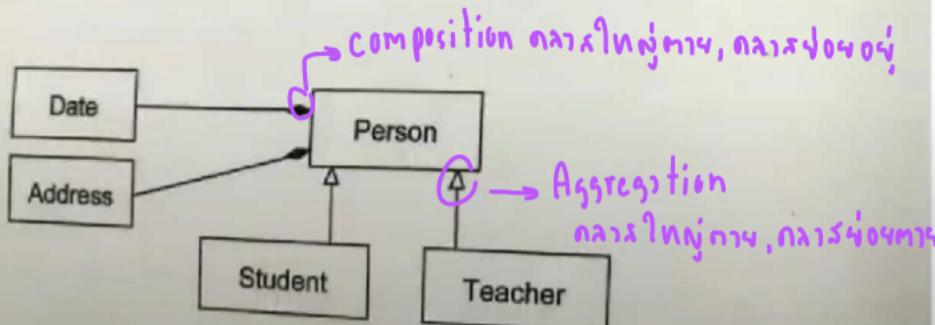
count Obj = 1

```

แบบฝึกหัดที่ 6

(ให้นักศึกษาฝึกหัดข้อมูลและฟังก์ชันต่างๆ ได้เองตามความเหมาะสม)

6.1 จากคลาส Person, Student และ Address ที่สร้างขึ้นในแบบฝึกหัดที่ 2 นั้น จงปรับปรุงคลาส Person โดยให้เก็บข้อมูลวันเดือนปีเกิดเป็นออบเจกต์ของคลาส Date และเพิ่มข้อมูลที่อยู่ซึ่งเป็นออบเจกต์ของคลาส Address ด้วย สำหรับคลาส Student ให้สืบทอดคุณสมบัติจากคลาส Person และจะสร้างคลาส Teacher ให้สืบทอดคุณสมบัติมาจากคลาส Person ด้วย



class Date & Address

```

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

class Date
{
private:
    int day, month, year;

public:
    Date()
    {
        set(0, 0, 0);
    }

    Date(int d, int m)
    {
        set(d, m, 0);
    }

    Date(int d, int m, int y)
    {
        set(d, m, y);
    }

    ~Date()
    {
        cout << "delete " << day << "/" << month << "/" << year << endl;
    }

    // set
    void set(int d, int m, int y)
    {
        day = d;
        month = m;
        year = y;
    }

    void setDay(int d)
    {
        day = d;
    }

    void setMonth(int m)
    {
        month = m;
    }

    void setYear(int y)
    {
        year = y;
    }

    // get
    int getDay()
    {
        return day;
    }

    int getMonth()
    {
        return month;
    }

    int getYear()
    {
        return year;
    }

    // show
    void showDate()
    {
        cout << "\n===== show Date =====" << endl;
        cout << day << "/" << month << "/" << year << endl;
    }
};

```

```

class Address
{
private:
    static int num;
    char Hnum[30];           // บ้านเลขที่
    char subDistrict[40];    // แขวง, ตำบล
    char district[40];       // อำเภอ, เขต
    char province[40];
    int postcode;

public:
    Address()
    {
        setAddress("none", "none", "none", "none", 0);
    }

    Address(char *n, char *s, char *d, char *p, int c)
    {
        setAddress(n, s, d, p, c);
    }

    ~Address()
    {
        cout << "delete Address " << endl;
    }

    void setAddress(char *n, char *s, char *d, char *p, int c)
    {
        strcpy(Hnum, n);
        strcpy(subDistrict, s);
        strcpy(district, d);
        strcpy(province, p);
        postcode = c;
    }

    void setHouseNum(char *n)
    {
        strcpy(Hnum, n);
    }

    void setSubDistrict(char *s)
    {
        strcpy(subDistrict, s);
    }

    void setDistrict(char *d)
    {
        strcpy(district, d);
    }

    void setProvince(char *p)
    {
        strcpy(province, p);
    }

    void setPostCode(int c)
    {
        postcode = c;
    }
};

```

```

// get
char *getHouseNum()
{
    return Hnum;
}

char *getSubDistrict()
{
    return subDistrict;
}

char *getDistrict()
{
    return district;
}

char *getProvince()
{
    return province;
}

int getPostCode()
{
    return postcode;
}

void showAddress()
{
    cout << "\n===== show Address =====" << endl;
    cout << "\nHouse Number: " << Hnum << endl;
    cout << "SubDistrict: " << subDistrict << endl;
    cout << "District: " << district << endl;
    cout << "Province: " << province << endl;
    cout << "Post code: " << postcode << endl;
}

```

1. สร้าง class Date & Address มาเก็บข้อมูล

2. สร้าง class Person ที่มี属性 Address และ Date

ในดัวอย่างโค้ดที่คุณให้มา `bday.set(0,0,0);` อธิบายในคลาส `Person` ดังนี้ `bday` เป็นออบเจกต์ของคลาส `Date` และ `set` เป็นเมธอดของคลาส `Date` ซึ่งถูกเรียกใช้โดยออบเจกต์ `bday` นั้นเอง

3. สร้าง class Student รับข้อมูลรุ่นพี่จาก Person

4. สร้าง class Teacher

1. Composition:

- `Date` และ `Address` ใน `Person` ใช้แนวคิดของ Composition ซึ่งหมายความว่าข้อมูลใน `Date` และ `Address` จะถูกจัดการโดย `Person` และจะถูกสร้างและทำลายพร้อมกับ `Person`
- ถ้า `Person` ถูกทำลาย, `Date` และ `Address` ที่เป็นสมาชิกของ `Person` จะถูกทำลายไปด้วย
- ดังนั้นถ้าคุณใช้ `Student` หรือ `Teacher` ซึ่งมีการอ้างอิงถึง `Person`, ข้อมูล `Date` และ `Address` ที่เก็บไว้ของใน `Student` และ `Teacher` จะไม่สามารถเข้าถึงได้หาก `Person` ที่จัดการข้อมูลเหล่านี้ถูกทำลาย

2. Aggregation:

- `Student` และ `Teacher` ใช้แนวคิดของ Aggregation กับ `Date` และ `Address` โดย `student` และ `Teacher` สามารถเข้าถึงข้อมูลเหล่านี้ได้ แต่ไม่รับผิดชอบต่อการจัดการหรือทำลายข้อมูลเหล่านี้
- ถ้า `Date` และ `Address` เป็นสมาชิกของ `Person` และ `Person` ถูกทำลาย, ข้อมูล `Date` และ `Address` ที่จัดการโดย `Person` จะหายไป
- ดังนั้นในกรณีที่ `Person` ถูกทำลาย, ข้อมูล `Date` และ `Address` ที่เก็บไว้ของใน `Student` และ `Teacher` ก็จะหายไปด้วยเนื่องจาก `Student` และ `Teacher` ที่พึ่งพาข้อมูลที่มาระหว่าง `Person`

class Person & Student

```

class Person
{
private:
    static int num;
    char fname[30];
    char lname[30];
    Date bday; // Composition
    Address add; // Composition
public:
    Person()
    {
        setName("Firstname", "Lastname");
        bday.set(0, 0, 0); // เมื่อobjของDateที่เรียกมีset()
        add.setAddress("none", "none", "none", "none", 0);
        num++;
    }

    Person(char *f, char *l, Date &bd, Address &a) /*:bday(bd), add(a) ส่วนนี้เป็นComposition*/
    {
        setPerson(f, l, bd, a);
        num++;
    }

    ~Person()
    {
        cout << "delete " << fname << " " << lname << endl;
    }

    // set
    void setPerson(char *f, char *l, Date &bd, Address &a)
    {
        strcpy(fname, f);
        strcpy(lname, l);
        bday = bd;
        add = a;
    }

    void setName(char *f, char *l)
    {
        strcpy(fname, f);
        strcpy(lname, l);
    }
}

```

```

class Student : public Person
{
private:
    char stdNum[10];
    char fac[20];
    char university[20];
    // Person* person; Aggregation

public:
    /*Student(Person* p, const char* id, const char* f, const char* u) :person(p){} Aggregation */
    Student()
    {
        setStudent("none", "none", "none");
    }

    Student(char *id, char *f1, char *u)
    {
        setStudent(id, f1, u);
    }

    // base ต้องใช้ส่วนเฉพาะเพื่อไปเรียกconstructor ของคลาสแม่ก่อนหากต้อง
    Student(char *f, char *l, Date &d, Address &a, char *id, char *fac, char *u)
    {
        setPerson(f, l, d, a);
        setStudent(id, fac, u);
    }

    // base Person(f, l, d, a)->เรียกใช้งาน constructor ของคลาส Person เซียนได้เลย
    /*Student(char *f, char *l, Date &d, Address &a, char *id, char *fac, char *u) :Person(f, l, d, a){
        setStudent(id, fac, u);
    }*/

    ~Student()
    {
        cout << "bye Student" << endl;
    }

    // set
    void setStudent(char *id, char *f1, char *u)
    {
        strcpy(stdNum, id);
        strcpy(fac, f1);
        strcpy(university, u);
    }

    void setStdNum(char *id)
    {
        strcpy(stdNum, id);
    }

    void setFaculty(char *f1)
    {
        strcpy(fac, f1);
    }
}

```

```

void setBday(Date &d)
{
    bday = d;
}

void setAdd(Address &a)
{
    add = a;
}

// get
char *getFname()
{
    return fname;
}

char *getLname()
{
    return lname;
}

Date getBday()
{
    return bday;
}

Address getAddress()
{
    return add;
}

void showPerson()
{
    cout << "\n==== show Person =====" << endl;
    cout << "\nPerson no. " << Person::count() << endl;
    cout << "Name: " << fname << " " << lname << endl;
    cout << "Birth Day: " << bday.getDay() << "/" << bday.getMonth() << "/" << bday.getYear() << endl;
    cout << "House Number: " << add.getHouseNum() << endl;
    cout << "SubDistrict: " << add.getSubDistrict() << endl;
    cout << "District: " << add.getDistrict() << endl;
    cout << "Province: " << add.getProvince() << endl;
    cout << "Post code: " << add.getPostCode() << endl;
}

static int count()
{
    return num;
}

int Person ::num; // นับคน

```

```

void setUniversity(char *u)
{
    strcpy(university, u);
}

// get
char *getStdNum()
{
    return stdNum;
}

char *getFaculty()
{
    return fac;
}

char *getUniversity()
{
    return university;
}

void showStudent()
{
    cout << "\n==== Student ====" << endl;
    cout << "\nStudent: " << getFname() << " " << getLname() << endl;
    cout << "university: " << university << endl;
    cout << "faculty: " << fac << endl;
    cout << "Student ID : " << stdNum << endl;
}

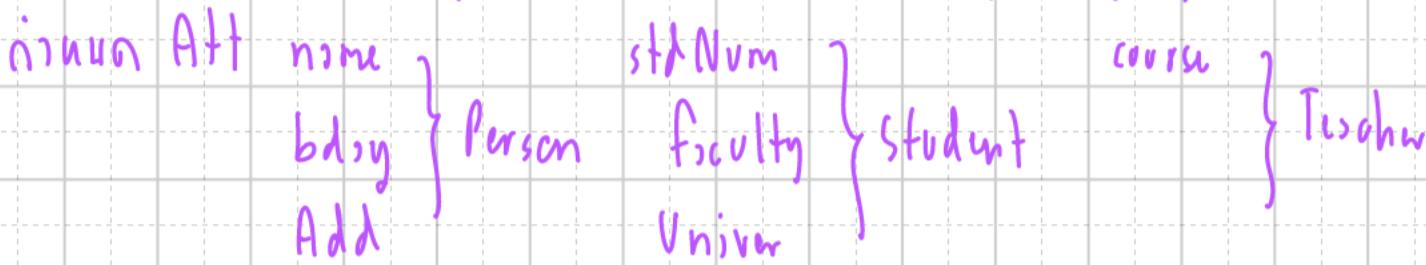
```


7.2 จงสร้างคลาส **TeachingAssistant** (อาจารย์ผู้ช่วยสอน) โดยสืบทอดคุณสมบัติมาจากการคลาส **Student** และคลาส **Teacher** ในข้อ 6.1 ซึ่งอาจารย์ผู้ช่วยสอนจะเป็นนักศึกษาที่มาช่วยอาจารย์ในการเรียนการสอน

Person



- ✓ 1. กำหนดให้ class Person เป็น virtual ก่อน (ไปแก้ code เก่า)
- ✓ 2. สร้าง class TA ที่ inherit กับ class stu, Teacher
- ✓ 3. สร้าง constructor, default con, destructor, set, get, show



func => email, year of study, course assigned, supervising-professor;

```

TeachingAssistant(char *fn, char *l, Date& d, Address& a
                  ,char *i, char *fac ,char *u
                  ,char *f, char *uni, char *c, double s
                  ,char *n, char *ap): Student(fn, l, d, a, i, fac, u), Teacher(fn, l, d, a
{
    setTA(n, ap);
}
  
```

ในตัวอย่างที่ให้นำ การเข้าค่าจะเป็นการเข้าให้กับคลาสแม่สองคลาส `Student` และ `Teacher` เท่านั้น โดยไม่มีการเข้าค่าที่คลาส `Person`

```

cpp
TeachingAssistant(char *i, char *fac ,char *u
                  ,char *f, char *uni, char *c,double s
                  ,char * n , char *ap):Student(i,fac,u) , Teacher(f,uni,c,s)
{
    setTA(n,ap);
}
  
```

แบบที่สอง (Multiple Inheritance Base Constructor):

ในตัวอย่างนี้ การเข้าค่าจะเป็นการเข้าให้กับคลาสแม่สามคลาส `Person`, `Student` และ `Teacher`

```

cpp
TeachingAssistant(char *fn, char *l, Date& d, Address& a
                  ,char *i, char *fac ,char *u
                  ,char *f, char *uni, char *c,double s
                  ,char * n , char *ap):Student(i,fac,u) , Teacher(f,uni,c,s) , Person(fn,l
{
    setTA(n,ap);
}
  
```

สรุป:

- แบบแรก: เข้าค่าให้กับคลาสแม่สองคลาส (`Student`, `Teacher`)
- แบบที่สอง: เข้าค่าให้กับคลาสแม่สามคลาส (`Person`, `Student`, `Teacher`)

ดังนั้น ในแบบที่สอง (Multiple Inheritance Base Constructor) เราจะสามารถเข้าค่าให้กับคลาสแม่ได้มากกว่าหนึ่งคลาส ซึ่งอยู่กับว่าเรามีคลาสแม่กี่คลาสที่เราต้องการเข้าค่าด้วยการเรียก constructor.

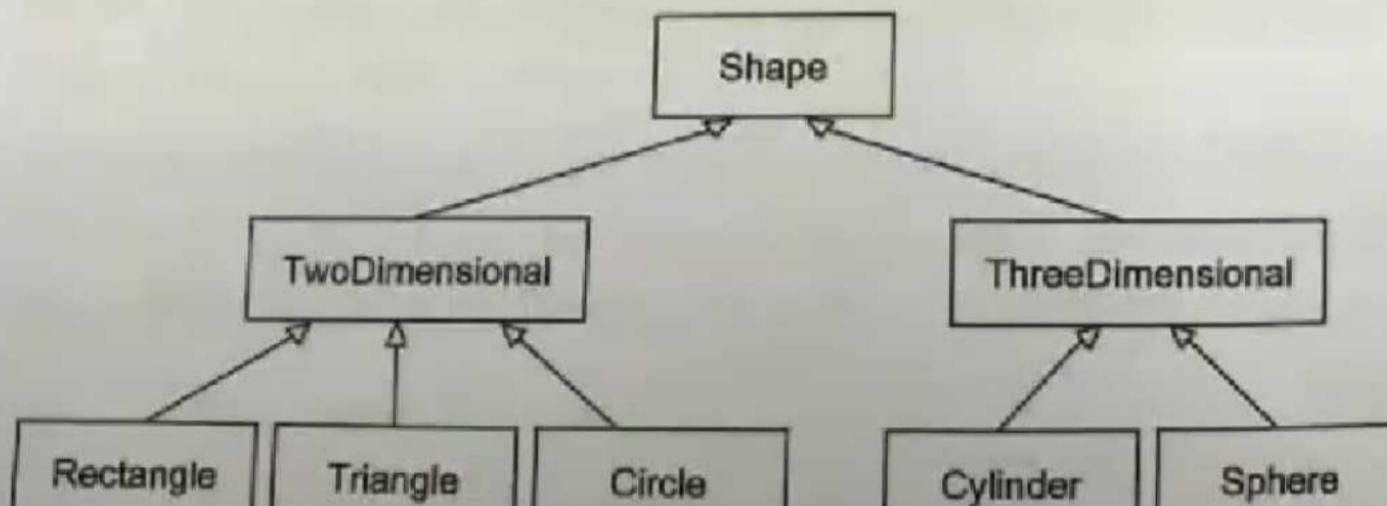
- ให้ทำการบ้านข้อ 7 มาปรับปรุง

- กำหนดให้แต่ละคลาสมี attribute ที่เป็น Pointer โดยจะปรับแก้ attribute เดิมหรือสร้างใหม่ก็ได้
- สร้าง assignment operator, copy constructor, ostream operator, istream operator ในทุกๆ คลาส โดยต้องกำหนดค่าให้ครบและถูกต้อง
- กรณี Pointer ให้จดพื้นที่และคืนพื้นที่ให้ถูกต้องด้วย

- ให้นำการบ้านข้อ 9 มาปรับปรุง

- โดยให้สามารถเรียกใช้ **operator<<** และ **operator>>** โดยใช้หลักการ **dynamic binding** ได้

10.4 จงสร้างคลาสตามโครงสร้างข้างล่างนี้ โดยกำหนดให้ **Shape** **TwoDimensional** และ **ThreeDimensional** เป็นคลาสนามธรรม จงสร้างฟังก์ชัน **area()** ในการหาพื้นที่ของ **Rectangle** **Triangle** **Circle** **Cylinder** และ **Sphere** โดยใช้หลักการ **dynamic binding** และจงสร้าง ฟังก์ชัน **volume()** ในการหาปริมาตรของ **Cylinder** และ **Sphere** โดยใช้หลักการ **dynamic binding** ในสิ่ง นอกจากนี้จงสร้างโอเปอเรเตอร์เพิ่มพิมพ์ค่าและโอเปอเรเตอร์รับค่าสำหรับคลาสทุกประนทุก คลาสโดยใช้หลักการ **dynamic binding** ไปด้วย



แบบฝึกหัดที่ 11

(ให้น.ศ.กำหนดข้อมูลและฟังก์ชันต่างๆ ได้เองตามความเหมาะสม)

11.1 จงสร้างฟังก์ชันใหม่เพลท ในการหาค่ามากที่สุดจากข้อมูล 3 จำนวน

11.2 จงสร้างคลาสใหม่เพลท ในการหาค่ามากที่สุด และน้อยที่สุด จากข้อมูลที่เก็บในอาร์เรย์

แบบฝึกหัดที่ 12

12.1 จงเขียนโปรแกรมให้มีการตรวจสอบโดย catch หากหลายชนิดของความผิดปกติ เช่น int double char และ char* เป็นต้น

12.2 จงเขียนโปรแกรมให้ความผิดปกติที่ถูกโยนออกมานั้นถูกดักจับโดย catch(...)

12.3 จงเขียนโปรแกรมที่แสดงถึงความสำคัญของลำดับของการตรวจสอบความผิดปกติ ซึ่งต้องเปลี่ยนลำดับแล้วจะได้ผลลัพธ์ที่แตกต่างกัน

12.4 จงเขียนโปรแกรมที่มีการโยนความผิดปกติออกจากฟังก์ชันที่เป็นการซ้อนกันของฟังก์ชันหลายชั้น โดยให้มีการดักจับความผิดปกติได้ในทุกชั้น

12.5 จงสร้างฟังก์ชันໂອເວອຣໃຣດິງที่มีการโยนความผิดปกติที่สอดคล้องกับชนิดของความผิดปกติที่ถูกโยนออกจากฟังก์ชันในคลาสแม่

12.6 จงสร้างคลาสที่สืบทอดคุณสมบัติมาจากการ exception ในการจัดการความผิดปกติเอง