

### Exercise 30

#### 1. จงหาว่าคำสั่งใดบ้างที่ผิด

```
#include<iostream>
using namespace std;
class A {
    int a;
protected:
    A addOne() { a+=1; return *this; }
public:
    A() { a=1; }
    A(int i) { set(i); }
    A& set(int i) { a=i; return *this; }
    int get() { return a; }
    int addA(A x) { return a+x.a; }
    A addNum(int i) { a+=i; return *this; }
};
class B {
    int b;
protected:
    int addA(A x) { return b+x.get(); }
public:
    B() { b=2; }
    B* set(int y) { b=y; return this; }
    int g1(A x) {
        return addA(x)+x.addNum(b).get();
    }
    int g2(A &x) { return x.addOne().get(); }
    B& g3(int x) {
        for (int i=0; i<x; i++)
            b++;
        return *this;
    }
    A g4(A &x) {
        return A(addA(x.addNum(b)));
    }
    int get() { return b; }
};
```

```
class C: public B {
    A k;
protected:
    int addA(A x) {
        return k.addA(x)+addA(x);
    }
public:
    C() { k.set(4); }
    C(int i) { k.set(i); }
    void h1(A &m) { k.set(g4(m)).get(); }
    C h2(A *k) { return C(this->k.addA(*k)); }
    B h3(int i) {
        return g3(k.addNum(i).get());
    }
    C h4(B b) { k.addNum(b.get()); }
    friend int f(A x);
}
int f(A x) {
    C c;
    int i;
    c.addA(x);
    c.h2(&x).k.addNum(2);
    return c.k.addOne().get();
}
int main() {
    A a(5);
    B b;
    C c(3);
    int k;
    b = c.h2(&a).g3(a.get());
    cout<<c.h3(b.get()).g4(a)<<endl;
    cout<<b.set(a.get())->get()<<endl;
    k=c.g3(b.get()).g3(a.get()).get();
    k=c.h2(&a).get()+b.g1(a);
    k=c.h4(b).g4(&a).get();
}
```

2. โปรแกรมข้างล่างนี้มีคำตอบทั้งหมด 26 บรรทัด จะแสดงผลลัพธ์ทั้ง 26 บรรทัด

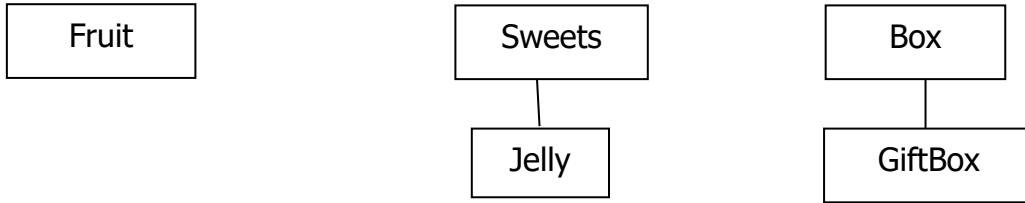
```
#include<iostream>
using namespace std;
class A {
    int a;
public:
A() { setA(2); cout<<"A"<<endl; }
A(int i) {
    setA(i); cout<<"AA"<<endl;
}
~A() { cout<<"~A "<<a<<endl; }
void setA(int i) { a=i; }
int getA() { return a; }
A f1(int i) { a+=i; return *this; }
A& f2(A x) {
    a=++x.a; return *this;
}
A* f3(A &a2) {
    f2(a2).f1(a2.a); return this;
}
void show() {
    cout<<"a="<<a<<endl;
}
};

class B {
A *b;
public:
B() { b=0; cout<<"B"<<endl; }
B(A *x) {
    setB(x); cout<<"BB"<<endl;
}
~B() { cout<<"~B "<<endl; }
void setB(A *x) { b=x; }
A* getB() { return b; }
};
```

```
class C: public B {
    int c;
public:
C() { c=0; cout<<"C"<<endl; }
C(A* x) {
    setB(x);
    setC(x->getA());
    cout<<"CC"<<endl;
}
~C() { cout<<"~C "<<c<<endl; }
C& setC(int i) { c=i; return *this; }
int getC() { return c; }
};

int main() {
A a1;
A a2(5);
B b(&a1);
a1.setA(7);
a2.f2(a1).show();
cout<<b.getB()->getA()<<endl;
a1.f1(2).f1(3).show();
a1.show();
a1.setA(1);
b.getB()->f3(a2);
b.getB()->show();
a2.show();
A *a3 = new A;
C c(a3);
c.setC(3).getB()->f1(4).show();
delete a3;
cout<<"--- oop ---"<<endl;
}
```

3. จงสร้างคลาสตามรูปด้านล่าง โดยมีการรับทดสอบตามรูป โดยกำหนดให้การรับทดสอบทั้งหมดเป็นการรับทดสอบแบบ public ทุกคลาสให้เก็บและทริบิวต์ทั้งหมดไว้ในส่วนของ private



แต่ละคลาสประกอบด้วยแอดทริบิวต์และความหมาย ดังนี้ (ห้ามเปลี่ยนชื่อแอดทริบิวต์)

	แอดทริบิวต์	ความหมาย
Fruit	int type;	1.สตรอเบอรี่ 2.กีวี 3.เชอร์รี่
Sweets	double price; int num; Fruit **fruit;	ราคาต่อชิ้น ขนาดของอาร์เรย์ fruit อาร์เรย์ขนาด กุบาท เก็บพอยเตอร์ที่ชี้ไปยัง Fruit
Jelly	int typeJelly;	1.วันน้ำสัด 2.วันกะทิ
Box	Jelly *jelly; int numPieces; int typeBox;	พอยเตอร์ที่ชี้ไปยัง Jelly จำนวนชิ้น 1.กล่องสีเหลือง 2.กล่องกลม
GiftBox	int bowColor;	สีของโบว์ 1.ชมพู 2.แดง 3.ขาว 4.น้ำเงิน

จงสร้างฟังก์ชันต่างๆ ในส่วนของ public ดังนี้ (ห้ามสร้างฟังก์ชันเพิ่มเติมถ้าโจทย์ไม่ได้สั่ง)

#### ทุกคลาส จงสร้าง

- default constructor ในการกำหนดค่าให้กับแอดทริบิวต์ทุกด้วย โดยมีการกำหนดค่าเริ่มต้น ดังนี้
  - Fruit ให้เป็นสตรอเบอรี่
  - Sweets ให้ price=10 num=0 fruit=0
  - Jelly วันน้ำสัด ราคา 5 บาท ไม่ใส่ผลไม้ num=0 fruit=0
  - Box กล่องสีเหลือง ยังไม่ใส่ขนม numPieces=0 sweets=0
  - GiftBox โบว์สีขาว กล่องสีเหลือง ยังไม่ใส่ขนม numPieces=0 sweets=0
- สร้างฟังก์ชัน set ในการกำหนดค่าให้กับแอดทริบิวต์แต่ละตัว
  - กรณี price ราคายอดคงมากกว่าหรือเท่ากับศูนย์
  - กรณี fruit ให้สร้างฟังก์ชัน setFruit(int, Fruit\*) ในการกำหนดค่าให้กับ fruit โดยกำหนดค่าในอาร์เรย์ที่ลีบ 1 ค่า โดยระบุตำแหน่งที่เก็บในอาร์เรย์
  - กรณี gnum และ gnumPieces ค่าต้องมากกว่าหรือเท่ากับศูนย์ และกรณี setNum ให้คืนพื้นที่เดิมและจะคงพื้นที่ใหม่ให้กับ fruit ด้วย
  - กรณี type ต่างๆ และ bowColor ให้กำหนดค่าให้อยู่ในขอบเขตที่กำหนดตามคำอธิบาย
- ฟังก์ชัน get ในการรีเทิร์นแอดทริบิวต์แต่ละค่า เช่น getPrice()
  - สำหรับ fruit ให้รีเทิร์นค่าสมาชิกในอาร์เรย์ที่ลีบ 1 ค่าโดยรับพารามิเตอร์เป็นตำแหน่งที่เก็บในอาร์เรย์

### คลาส Box, GiftBox จงสร้าง

- constructor โดยรับข้อมูลจากพารามิเตอร์ในการกำหนดค่าให้กับแอทริบิวต์ทุกด้วยรวมทั้ง base class โดยใช้วิธีการไปเรียกใช้ constructor ของ base class และให้ใช้เงื่อนไขเดียวกับฟังก์ชัน set
- ฟังก์ชัน show ในการแสดงค่าทุกอย่างรวมทั้งค่าใน base class และรายละเอียดของ jelly ออกทางจอภาพ

### คลาส Sweets จงสร้าง

- destructor ในกรณีพื้นที่ของแอทริบิวต์ที่มีการจอง
- ฟังก์ชันในการนับจำนวนอ้อมเจกต์ที่เป็น Sweets ว่า ณ ปัจจุบันมีจำนวนอ้อมเจกต์คงอยู่ในโปรแกรมทั้งหมดเท่าไร ถึงแม้ว่าจะไม่มีอ้อมเจกต์อยู่เลยก็ต้องสามารถบอกได้ด้วย ให้สร้างแอทริบิวต์และฟังก์ชันที่เกี่ยวข้องเพิ่มเติมได้

### ใน main

- จงสร้าง strawberry, kiwi, cherry เป็นอ้อมเจกต์ของ Fruit โดยกำหนดค่าเป็นส่วนเบอร์กีวี และ เชอรี่ ตามลำดับ
- จงสร้าง jelly เป็นอาร์เรย์ขนาด 3 เก็บอ้อมเจกต์ของ Jelly โดยกำหนดค่าดังนี้
  - o jelly[0] เป็นรุ้นนมสด ราคาชิ้นละ 6 บาท ใส่ strawberry กับ kiwi
  - o jelly[1] เป็นรุ้นกะทิ ราคาชิ้นละ 8 บาท ใส่ strawberry kiwi และ cherry
  - o jelly[2] เป็นรุ้นกะทิ ราคาชิ้นละ 3 บาท ไม่ใส่อะไรเลย
- จงสร้าง box เป็นพอยเตอร์ที่ซึ่งไปยัง Box จากนั้นให้ Jong พื้นที่ขนาด 2 เก็บอ้อมเจกต์ของ Box โดยกำหนดค่าดังนี้
  - o box[0] เป็นกล่องกลม ใส่ jelly[0] จำนวน 12 ชิ้น
  - o box[1] เป็นกล่องสี่เหลี่ยม ใส่ jelly[2] จำนวน 15 ชิ้น
- จงสร้าง giftBox เป็นอ้อมเจกต์ของ GiftBox เป็นกล่องกลม ผูกโบว์สีแดง ใส่ jelly[1] จำนวน 10 ชิ้น
- จงแสดงค่าทั้งหมดใน box ออกทางจอภาพโดย ไม่ ให้เรียกใช้ฟังก์ชัน show
- จงหาว่า box[1] กับ giftBox กล่องไหนมีราคารวมของรุ้นแพงกว่ากัน
- จงเปลี่ยนให้ jelly[2] ใส่ kiwi ราคาชิ้นละ 6 บาท และแสดงค่าของ jelly[2] ออกทางจอภาพ
- จงคืนพื้นที่ที่มีการจอง