

Exercise 3

1. จงติดตามการทำงานของโปรแกรมข้างล่างนี้ว่าได้ผลลัพธ์อะไรบ้าง

```
#include<iostream>
using namespace std;
class A {
    int x;
    public:
        A() {x=1; cout<<'A'<<endl; }
        A(int a) {
            x=a;
            cout<<x<<endl;
        }
};
class B: public A {
    int y;
    protected:
        int k;
    public:
        B() { y=2; cout<<'B'<<endl; }
        B(int a, int b):A(a) { y=b; cout<<y<<endl; }
};
class C: public B {
    int z;
    public:
        C() { z=3; cout<<'C'<<endl; }
};
class D:public B {
    public:
        D() { cout<<'D'<<endl; }
        D(int k):B(k, 3) { this->k=k*2; cout<<k<<endl; }
};
void main() {
    A a;
    B *b;
    C c;
    D d(2);
    b = new D;
}
```

2. จงติดตามการทำงานของโปรแกรมข้างล่างนี้ว่ามี error ที่ใดบ้าง จงให้เหตุผลประกอบ

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

class B;
class A {
    int a;
    protected:
        int k;
    public:
        A() { a=1; k=2; }
        void show() { cout<<a<<endl; }
        friend int f(A, B);
};
class B: A {
    int b;
    public:
        B() { b=2; k=3; }
};
class C: protected B {
    int c;
    public:
        C() { c=3; k=1; }
};
class D {
    int d;
    public:
        D() { d=4; }
        friend int f(A, B);
};
int f(A p, B q) {
    D t;
    int s, r=0;
    s = t.d+p.a;
    r = q.b+p.k;
    return (s+r);
}
```

```

void main() {
    A a;
    B b;
    C c;
    a.show();
    b.show();
    c.show();
    cout<<f(a,b)<<endl;
}

```

3. จงติดตามการทำงานของโปรแกรมข้างล่างนี้ว่าได้ผลลัพธ์อะไรบ้าง

```

#include<iostream>
using namespace std;

class A {
    int a;
    public:
        A() {a=3; cout<<'A'<<endl; }
        ~A() { cout<<"bye A"<<endl; }
        A &f(A &x) { a=a+x.a; return *this; }
        void show() { cout<<a<<endl; }
};

class B: public A {
    int b;
    public:
        B() { b=0; cout<<'B'<<endl; }
        B(int a) { b=a; }
        ~B() { cout<<"bye B"<<endl; }
        void set(int a, int b) { b=b+a; show(); }
        void show() { cout<<b<<endl; }
};

class C:public B {
    int c;
    public:
        C() { c=0; cout<<'C'<<endl; }
        C(int k):B(k) { c=1; }
        ~C() {cout<<"bye C"<<endl;}
        int get() { return c; }
};

```

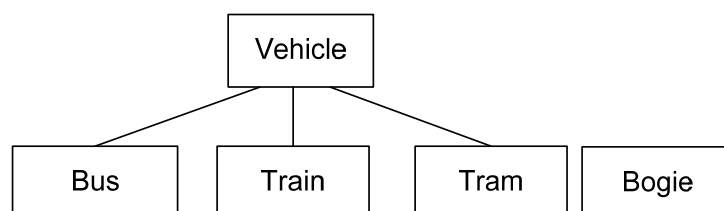
```

class D: public C {
    int d;
    public:
        D():C(5) { d=2; cout<<'D'<<endl; }
        ~D() { cout<<"bye D"<<endl; }
        void show() { cout<<d<<"-"<<get()<<"-"; B::show(); }
};

void main() {
    A a;
    D d;
    d.show();
    d.f(a);
    d.A::show();
    C c;
    c.set(1,2);
}

```

4.



จงเขียนโปรแกรมภาษา C++ ในการสร้าง class Vehicle, class Bus, class Train และ class Tram โดยที่ class ทั้งสี่มีความสัมพันธ์ดังแผนภาพที่แสดงข้างบนนี้ โดยที่ Bus, Train และ Tram สืบทอดมรดกแบบ public มาจาก class Vehicle นอกจากนี้จงสร้าง class Bogie โดยแต่ละ class มีรายละเอียดดังนี้

class Vehicle ประกอบด้วยข้อมูล

Num จำนวนยานพาหนะทั้งหมด

class Bus ประกอบด้วยข้อมูล

busID รหัสรถบัส
wheelNum จำนวนล้อ

class Train ประกอบด้วยข้อมูล	trainID bogieNum bogie	รหัสรถไฟ จำนวนตู้รถไฟ ตู้รถไฟ (เป็น pointer ที่ชี้ไปยัง Bogie)
class Tram ประกอบด้วยข้อมูล	tramID	รหัสรถแถม
class Bogie ประกอบด้วยข้อมูล	bogieID type	รหัสตู้รถไฟ ประเภทของตู้รถไฟ เช่น ตู้เสบียง ตู้นอน เป็นต้น

จงสร้างฟังก์ชันในส่วนของ public ของแต่ละคลาสตามเงื่อนไขต่างๆ ดังนี้

1. จงกำหนดรหัสรถบัส รหัสรถไฟ และ รหัสรถแถม ให้กับอ็อบเจกต์ที่เป็นรถบัส รถไฟ และรถแถม โดยอัตโนมัติ โดยค่าของรหัสรถแต่ละชนิดจะต้องถูกบวกเพิ่มทีละหนึ่งทุกครั้งที่มีการสร้างอ็อบเจกต์แต่ละชนิดในโปรแกรม
2. จงสร้างฟังก์ชันในการนับจำนวนยานพาหนะทั้งหมดที่ถูกสร้างขึ้น ในกรณีที่ไม่มียานพาหนะใดถูกสร้างขึ้นเลยก็ต้องสามารถแสดงได้ว่าไม่มีอ็อบเจกต์ใดๆ ถูกสร้างขึ้นเลยในโปรแกรม
3. ทุก class จงสร้าง
 - 3.1 default constructor ในการกำหนดค่าเริ่มต้นทุกค่าให้เป็นค่าว่าง หรือ 0 (ยกเว้นรหัสรถซึ่งต้องเป็นการกำหนดค่าโดยอัตโนมัติ)
 - 3.2 constructor ในการกำหนดค่าทุกค่า (รวมทั้งค่าของ base class) โดยรับข้อมูลจาก keyboard ในกรณีของ class Train ให้จองพื้นที่ให้กับ bogie ให้มีขนาดเท่ากับ bogieNum
 - 3.3 destructor ให้พิมพ์ชื่อคลาสที่กำลังถูกทำลายออกทางหน้าจอ นอกจากนี้ถ้ามีการจองพื้นที่ให้กับ attribute ของ class ก็ให้มีการคืนพื้นที่ให้หมดด้วย
 - 3.4 จงสร้างฟังก์ชัน show() ให้กับทุกๆ คลาส ในการพิมพ์ค่าทุกค่าของ object อย่างถูกต้อง (รวมทั้งค่าของ based class)
 - 3.5 จงสร้างฟังก์ชัน set() ให้กับทุกๆ คลาส ในการกำหนดค่าทุกค่าของ object อย่างถูกต้อง (รวมทั้งค่าของ based class)
4. ใน main() จงสร้างอ็อบเจกต์ของคลาสทุกคลาส โดยแต่ละอ็อบเจกต์ให้ไปเรียกใช้ constructor ที่มีการกำหนดค่าเริ่มต้นโดยการรับจาก keyboard (สำหรับ Train ให้กำหนด bogieNum=5) จากนั้นให้แสดงผลลัพธ์ของทุกอ็อบเจกต์ออกทางจอภาพ

หมายเหตุ: ทุกๆ object จะต้องถูกสร้างและถูกทำลายได้อย่างถูกต้อง สามารถสร้าง data และ function เพิ่มเติมได้ตามความเหมาะสม รหัสยานพาหนะแต่ละชนิดให้เริ่มต้นด้วยรหัส 1