

ภาษา C++ เรื่อง โครงสร้างคลาสและอ้อม
เจกต์ 10 จาก 10
คะแนน

พิจารณาโค้ดต่อไปนี้ เพื่อตอบคำถาม ในส่วนนี้

```

1  #include <iostream>
2  using namespace std;
3
4  class Book {
5  private:
6  data [ string name;
7  mem [ string author;
8  int quantity; // จำนวนในสต็อก
9  public:
10 constructor [ Book(string n, string a) {
11     name = n;
12     author = a;
13     quantity = 10;
14 }
15 void Buy(int buy_quantity) { mutator
16     if (buy_quantity <= quantity)
17         quantity = quantity - buy_quantity;
18 }
19 int GetQuantity() const { accessor
20     return quantity;
21 }
22 };
23
24 int main() {
25 object [ Book c_book("C Programming", "Dennis");
26 Book python_book("Python Cookbook", "Brian"); ] constructor
27     private
28     c_book.quantity = 25; X
29     c_book.Buy(5);
30     Buy(c_book,5); X
31 }

```

- ✓ โค้ดนี้สร้าง object ทั้งหมดกี่ object และบรรทัดใด มีการสร้าง object

- สร้าง 1 object ในบรรทัด 4
- สร้าง 3 object ในบรรทัด 6 - 8
- สร้าง 2 object ในบรรทัด 25 - 26
- สร้าง 1 object ในบรรทัด 30
- ไม่มีการสร้าง object ในโค้ดนี้

✓ โค้ดบรรทัด ได้ประกาศ data member

1/1

- บรรทัด 4
- บรรทัด 6 - 8 ✓
- บรรทัด 10 - 14
- บรรทัด 15 - 21
- บรรทัด 25 - 26
- ไม่มีการประกาศ data member ในโค้ดนี้

✓ โค้ดบรรทัด ได้ประกาศ/นิยาม constructor

1/1

- บรรทัด 4
- บรรทัด 6 - 8
- บรรทัด 10 - 14 ✓
- บรรทัด 15 - 21
- บรรทัด 25 - 26
- บรรทัด 28 - 30
- ไม่มีการนิยาม constructor

✓ โค้ดบรรทัด ได้มีการเรียกใช้ constructor

1/1

- บรรทัด 4

✓ โค้ดบรรทัด ใดมีการเรียกใช้ constructor

1/1

- บรรทัด 4
- บรรทัด 6 - 8
- บรรทัด 10 - 14
- บรรทัด 15 - 21
- บรรทัด 25 - 26
- บรรทัด 28 - 30
- ไม่มีการเรียกใช้ constructor ในโค้ดนี้

✓ โค้ดบรรทัด ใดประกาศ/นิยาม member function

1/1

- บรรทัด 4
- บรรทัด 6 - 8
- บรรทัด 10 - 14
- บรรทัด 10 - 21
- บรรทัด 15 - 21
- บรรทัด 25 - 26
- บรรทัด 29 - 30
- ไม่มีการนิยาม member function ในโค้ดนี้



✓ โค้ดบรรทัด ใดประกาศ/นิยาม member function

1/1

- ไม่มีการนิยาม member function ในโค้ดนี้

- ✓ พิจารณาการเรียก member function Buy() ในบรรทัด 29 - 30 จนตอบว่าบรรทัดใดถูกต้อง 1/1

- บรรทัดที่ 29 ถูกต้อง เนื่องจาก Buy เป็นฟังก์ชันภายในคลาส ต้องเรียกผ่าน dot operator ✓
- บรรทัดที่ 30 ถูกต้อง เนื่องจากการเรียกฟังก์ชันต้องใส่ implicit parameter และ explicit parameter ให้ครบ
- สามารถเรียกได้ทั้งสองแบบ เนื่องจากภาษา C++ มีความยืดหยุ่น

- ✓ member function Buy() เป็นฟังก์ชันประเภทใด 1/1

- accessor function
- mutator function ✓

- ✓ member function GetQuantity() เป็นฟังก์ชันประเภทใด 1/1

- accessor function ✓
- mutator function

- ✓ ข้อใดอธิบายโค้ดบรรทัดที่ 28 ได้ถูกต้อง 1/1

mutator function

✓ ข้อใดอธิบายโค้ดบรรทัดที่ 28 ได้ถูกต้อง 1/1

โค้ด `c_book.quantity = 25;` นั้นถูกต้อง การใช้ dot operator เป็นการกำหนดให้ data member quantity

โค้ด `c_book.quantity = 25;` ไม่ถูกต้อง เนื่องจากเรา
● ประกาศ data member เป็น private จึงไม่สามารถ
กำหนดค่าผ่าน dot opeator ได้ ✓

✓ พิจารณาการใช้ private ในบรรทัดที่ 5 และใช้ public ในบรรทัดที่ 9 ข้อใดไม่ถูกต้อง 1/1

โดยทั่วไป ควรประกาศ data member ให้เป็น private
เพื่อป้องกันการแก้ไขค่าที่ไม่ถูกต้อง

เราชาระบุประกาศทุกอย่าง ให้เป็น public เพื่อให้
● กำหนดค่าและอ่านค่าได้ง่าย ผ่าน dot operator เรา
ประกาศ private ในบางกรณีเท่านั้น ✓

การประกาศแบบ private ทำให้ไม่สามารถเข้าถึงค่าหรือ
ฟังก์ชันด้วย dot operator

การประกาศแบบ private ทำให้ต้องสร้าง member
function เพิ่มเติมเพื่ออ่านหรือกำหนดค่า แต่จะตรวจสอบ
ค่าก่อนการกำหนดได้ ทำให้ค่ามีความถูกต้อง

ภาษา C++ เรื่องคลาสและออบเจกต์

7 จาก 7 คะแนน

✓ แบบใดถูกต้องที่สุด หากต้องการประมวลผลข้อมูล 1/1
ร้านสุกี้เอ็มเค (Restaurant แปลว่า ร้านอาหาร)

- การประกาศแบบ private ทำให้ต้องสร้าง member function เพิ่มเติมเพื่ออ่านหรือกำหนดค่า แต่จะตรวจสอบค่าก่อนการกำหนดได้ ทำให้ค่ามีความถูกต้อง

ภาษา C++ เรื่องคลาสและออบเจกต์

7 จาก 7 คะแนน

- ✓ แบบได้ถูกต้องที่สุด หากต้องการประมวลผลข้อมูล 1/1
ร้านสุกี้เอ็มเค (Restaurant แปลว่า ร้านอาหาร)

```
class MKSuki {  
public:  
    string name;  
    // ...  
};  
  
int main() {  
    MKSuki mk_suki;  
    mk_suki.name = "MK Suki";  
}
```

- นิยามคลาส MKSuki

```
class Restaurant {  
public:  
    string name;  
    // ...  
};  
  
int main() {  
    Restaurant mk_suki;  
    mk_suki.name = "MK Suki";  
}
```

- นิยามคลาส Restaurant



- ✓ แบบได้ถูกต้องที่สุด หากต้องการประมวลผลข้อมูล 1/1
โทรศัพท์

Ⓐ นิยามคลาส Restaurant



- ✓ แบบได้ถูกต้องที่สุด หากต้องการประมวลผลข้อมูล 1/1
โทรศัพท์

```
class MobilePhone {
public:
    string phone_number;
    string type;
};
int main() {
    MobilePhone my_phone;
    my_phone.phone_number = "0811234567";
    my_phone.type = "iPhone";

    MobilePhone mom_phone;
    mom_phone.phone_number = "0819876543";
    mom_phone.type = "Andriod";
}
```

Ⓑ คลาส MobilePhone



```
class IPhone {
public:
    string phone_number;
};

class Android {
public:
    string phone_number;
};

int main() {
    IPhone my_phone;
    my_phone.phone_number = "0811234567";

    Android mom_phone;
    mom_phone.phone_number = "0819876543";
}
```

Ⓒ คลาส IPhone และคลาส Android

- ✓ หากเราต้องการสร้างคลาส Movie ที่แสดงถึง
ภาพยนตร์ ข้อใดน่าจะเป็น data member ของ
คลาส 1/1

Ⓓ หนัง ภาพยนตร์ ซีรีส์ ละคร

```

        Android mom_phone;
        mom_phone.phone_number = "0819876543";
    }
}

```

- คลาส IPhone และคลาส Android

- ✓ หากเราต้องการสร้างคลาส Movie ที่แสดงถึง
ภาพยนตร์ ข้อใดน่าจะเป็น data member ของ
คลาส 1/1

- หนัง ภาพยนตร์ ซีรี ละคร
- The Batman, Harry Potter, แม่นาค
- หนังไทย หนังสยองขวัญ หนังโรแมนติก (thai, horror, romantic)
- ชื่อภาพยนตร์ ผู้กำกับ ประเภท ปีที่เผยแพร่ (name, director, type, year) ✓

- ✓ หากเราต้องการสร้างคลาส Actor ที่แสดงถึงนัก
แสดง ข้อใดน่าจะเป็น data member ของคลาส 1/1

- แอน ทองประสม, ญาญ่า อุรัสยา, นาย ณภัทร
- ชื่อนักแสดง อายุ หนังที่แสดง รางวัลที่ได้ ✓
- นักแสดงอิสระ นักแสดง ในสังกัด
- นางเอก พระเอก ตัวประกอบ

- ✓ ข้อใดอธิบายการใช้ private และ public ได้ ** ถูก 1/1
ต้อง **

- ✓ ข้อใดอธิบายการใช้ private และ public ได้ ** ถูก 1/1
ต้อง **

- เราชรบประภากทุกอย่าง ให้เป็น public เพื่อให้กำหนดค่า และอ่านค่าได้ง่าย ผ่าน dot operator เราประภาก private ในบางกรณีเท่านั้น
- เราไม่ควรประภาก data member แบบ private เพราะทำให้ต้องสร้าง GetName() และ SetName() เพื่ออ่าน หรือกำหนดค่า ทำให้โค้ดยาวเกินจำเป็น
- โดยทั่วไป ควรประภาก data member ให้เป็น public และ member function ให้เป็น private

- การประภาก data member แบบ private จะช่วยให้ ✓
 เราตรวจสอบค่าได้เมื่อมีกำหนดค่า data member ส่งผลให้ค่าถูกต้องเสมอ

พิจารณาคลาส Rectangle ในการตอบคำถาม 2 ข้อถัดไป

```
class Rectangle {
private:
    int width;
    int height;
public:
    Rectangle(int size) {
        width = size;
        height = size;
    }
    Rectangle(int w, int h) {
        width = w;
        height = h;
    }
    void ChangeSize(int delta_width) {
        width = width + delta_width;
    }
    void ChangeSize(int delta_height) {
        height = height + delta_height;
    }
};
```

**con
struc
tor**

superparameter

datatype **เดียวกัน**

✓ โค้ดด้านบนมี constructor 2 อัน C++ อนุญาตให้ 1/1 ทำแบบนี้ได้หรือไม่

- ทำได้ เรียกว่า constructor replication ช่วยเช็คความถูกต้องของฟังก์ชัน
- ทำได้ เรียกว่า constructor override ช่วยให้โปรแกรมมีความยืดหยุ่น
- ทำได้ เรียกว่า constructor overload ช่วยให้โปรแกรมมีความยืดหยุ่น ✓
- ทำไม่ได้ C++ ไม่อนุญาตให้มี constructor มากกว่า 1 อันในคลาสเดียวกัน

✓ โค้ดด้านบนมีฟังก์ชันซื้อเดียวกัน 2 ฟังก์ชัน C++ 1/1 อนุญาตให้ทำแบบนี้ได้หรือไม่ เลือกข้อที่ถูกต้องที่สุด

- ในโค้ดนี้ทำได้ เนื่องจาก 2 ฟังก์ชันนี้รับพารามิเตอร์ต่างกัน คือ delta_width กับ delta_height
- ปกติจะทำได้ถ้า type ของพารามิเตอร์ต่างกัน แต่ในโค้ดนี้จะเกิด compile error เนื่องจาก 2 ฟังก์ชันนี้รับพารามิเตอร์ที่มี type เดียวกัน แม้ว่าชื่อพารามิเตอร์จะต่างกันก็ตาม ✓
- ทำไม่ได้ เนื่องจาก C++ จะอนุญาตให้มีฟังก์ชันซื้อเดียวกัน 2 ฟังก์ชันหากจำนวนพารามิเตอร์ต่างกันเท่านั้น ในโค้ดนี้รับ 1 พารามิเตอร์เท่ากัน จึงทำไม่ได้
- ทำไม่ได้ C++ ไม่อนุญาตให้มีฟังก์ชันซื้อเดียวกันในคลาสเดียวกัน



เดียวกัน

ภาษา C++ เรื่อง Handles, new, delete

2 จาก 2 คะแนน

- ✓ จงตอบว่ามีอับเจกต์ของคลาส Car จำนวนกี่อับ 1/1
เจกต์ในโค้ดการสร้างอับเจกต์ต่อไปนี้

```
1 #include <iostream>
2 using namespace std;
3
4 class Car {
5 private:
6     string license;
7     int power;
8
9 public:
10    Car(string l) {
11        license = l;
12        power = 100;
13    }
14    void Drive(int distance) {
15        power -= distance;
16    }
17    int GetPower() const {
18        return power;
19    }
20};
21
22 int main() {
23     Car *car1 = new Car("1nn1234");
24     Car car2("8nn8765");
25     Car *car3 = car1;
26     Car &car4 = car2;
27     car2.Drive(10);
28     car3->Drive(5);
29 }
```



1



2



3



4

- ✓ จากโค้ด Car ด้านบน หากมีการพิมพ์ค่าตามโค้ด 1/1
ดังนี้ จงตอบลิ่งที่โปรแกรมพิมพ์ออกมาทางหน้าจอ

3 4

- ✓ จากโค้ด Car ด้านบน หากมีการพิมพ์ค่าตามโค้ด 1/1
ดังนี้ จะตอบลิ่งที่โปรแกรมพิมพ์ออกมาทางหน้าจอ

```
int main() {
    Car *car1 = new Car("1nn1234");
    Car car2("8กช8765");
    Car *car3 = car1;
    Car &car4 = car2;
    car2.Drive(10);
    car3->Drive(5);

    cout << car1->GetPower() << endl;
    cout << car2.GetPower() << endl;
    cout << car3->GetPower() << endl;
    cout << car4.GetPower() << endl;
}
```

power	drive
100	95
100	90
100	95
100	90

 100 100 100 100 100 90 95 100 95 90 95 90 ✓ 95 90 95 100

Composition

1 จาก 1 คะแนน

- ✓ โค้ดข้อใดแสดงถึงการทำงานร่วมกันของคลาสแบบ 1/1
composition (Garage แปลว่าที่จอดรถ/โรงรถ)

Composition

1 จาก 1 คะแนน

- ✓ โค้ดข้อใดแสดงถึงการทำงานร่วมกันของคลาสแบบ 1/1
composition (Garage แปลงว่าที่จอดรถ/โรงรถ)

```
class Garage {  
private:  
    string car1;  
    string car2;  
  
    // . . . code . . .  
};
```

- คลาสมี data member เป็น primitive type

```
class Garage {  
private:  
    Car car1;  
    Car car2;  
  
    // . . . code . . .  
};  
  
class Car {  
    // . . . code . . .  
};
```

- คลาสมี data member เป็นคลาส ✓
อื่น

```
class Garage {  
private:  
    class Car {  
        // . . . code . . .  
    }  
};
```

```
private:  
    Car car1;  
    Car car2;  
  
    // . . . code . . .  
};  
  
class Car {  
    // . . . code . . .  
};
```

- คลาสมี data member เป็นคลาส ✓
อื่น

```
class Garage {  
private:  
    class Car {  
        // . . . code . . .  
    }  
};
```

- ประกาศคลาสอยู่ในคลาสอื่น

```
class Garage {  
    // . . . code . . .  
};  
  
class Car : public Garage {  
    // . . . code . . .  
};
```

- คลาสเชื่อมกับคลาสอื่นด้วย
เครื่องหมาย :

แบบฟอร์มนี้ถูกสร้างขึ้นภายใน มหาวิทยาลัยเกษตรศาสตร์

Google フォーム

6510450640

ภาษา C++ เรื่อง Composition และ Inheritance

4 จาก 4

คะแนน

- ✓ ความสัมพันธ์ของคลาสระหว่าง Restaurant กับ Food แบบใดเหมาะสมกว่ากัน 1/1

```
class Restaurant {
private:
    Food **menu;
    // ...
};

class Food {
    // ...
};
```

- คลาส Restaurant มี data member ประเภท Food ✓

```
class Restaurant {
    // ... code ...
};

class Food : public Restaurant {
    // ... code ...
};
```

- คลาส Food รับทอดมาจากคลาส Restaurant

- ✓ ความสัมพันธ์ของคลาสระหว่าง Employee, FullTimeEmployee, PartTimeEmployee, ฯลฯ 1/1



ຮ່າຍ ແລະ ປົບປອນ ລວມຫຍ່າຍຫນ ໄກ ໃຕ້າມເລີດ ໂດຍ
Restaurant

- ✓ ความສັນພັນຂອງຄລາສະໝວງ Employee, 1/1
FullTimeEmployee, PartTimeEmployee ແນບໃດ
ເທົາມສົມກວ່າກັນ

```
class Employee {
private:
    FullTimeEmployee **full_time;
    PartTimeEmployee **part_time;
    // ... code ...
};

class FullTimeEmployee {
|    // ... code ...
};

class PartTimeEmployee {
|    // ... code ...
};
```

- ຄລາສ Employee ມີ data
member ປະເກດ
FullTimeEmployee ແລະ
PartTimeEmployee

```
class Employee {
|    // ... code ...
};

class FullTimeEmployee : public Employee {
|    // ... code ...
};

class PartTimeEmployee : public Employee {
|    // ... code ...
};
```

- ຄລາສ FullTimeEmployee ແລະ
PartTimeEmployee ຮັບໂຄດມາ
ຈາກຄລາສ Employee ✓



ໄດ້ຮັບຜົນໄວ້ແລ້ວ ເຊິ່ງໄດ້ຮັບຜົນໄວ້ແລ້ວ ເຊິ່ງໄດ້ຮັບຜົນໄວ້ແລ້ວ

- ✓ โค้ดข้อใดแสดงถึงการทำงานร่วมกันของคลาสแบบ 2/2 composition (Garage แปลงว่าที่จอดรถ/โรงรถ)

```
class Garage {  
private:  
    string car1;  
    string car2;  
  
    // . . . code . . .  
};
```

- คลาสมี data member เป็น primitive type

```
class Garage {  
private:  
    Car car1;  
    Car car2;  
  
    // . . . code . . .  
};  
  
class Car {  
    // . . . code . . .  
};
```

- คลาสมี data member เป็นคลาส ✓
อื่น

```
class Garage {  
private:  
    class Car {  
        // . . . code . . .  
    }  
};
```

- ประกาศคลาสอยู่ในคลา索ื่น

```

class Garage {
    // . . . code . .
};

class Car : public Garage {
    // . . . code . .
};

```

- คลาสเชื่อมกับคลาสอื่นด้วยเครื่องหมาย :

ภาษา C++ เรื่อง Inheritance

6 จาก 6 คะแนน

- ข้อใดอธิบาย base class / derived class ไม่ถูก 1/1
ต้อง

- base class เป็นคลาสที่มีความทั่วไป มีฟังก์ชันที่ derived class หลายคลาสต้องมีเหมือนกัน
- derived class เป็นคลาสที่เป็น data member ของ base class ✓
- derived class เป็นคลาสที่ต่อขอดมาจาก base class
- แต่เฉพาะเจาะจงกว่า ต้องใช้ฟังก์ชันจาก base class และสามารถนิยามฟังก์ชันเพิ่มได้
- derived class ได้รับทดสอบ data member และ member function จาก base class และสามารถ override ฟังก์ชันจาก base class ได้

พิจารณาโค้ดต่อไปนี้ เพื่อตอบคำถามถัด ๆ ไป

```

1 #include <iostream>
2 using namespace std;
3

```

พงกชนจาก base class เด

พิจารณาโค้ดต่อไปนี้ เพื่อตอบคำถามถัด ๆ ไป

```

1  #include <iostream>
2  using namespace std;
3
4  class Monster {
5  private:
6      int health;
7  protected:
8      int power;
9  public:
10     Monster(int h, int p)
11         : health(h), power(p) {}
12
13     int GetHealth() {
14         return health;
15     }
16     void Refill(int p) {
17         power = power + p;
18     }
19 };
20
21 class Boss : public Monster {
22 public:
23     Boss(int h, int p)
24         : health(h), power(p) {}
25
26     void Refill(int p) {
27         power = power + (p * 2);
28     }
29     void Heal() {
30         health = health + 5;
31     }
32 };

```

- ✓ โค้ดข้างต้นมี error อよู่ 2 จุด จงเลือก 2 ข้อต่อไปที่ 2/2 ระบุ error ที่ถูกต้อง (ย้ำ ให้เลือก 2 ข้อ ไม่ใช่ข้อเดียว)

คลาส Boss ไม่สามารถนิยามฟังก์ชัน Refill ช้ากับคลาส

- โค้ดข้างต้นมี error อยู่ 2 จุด จงเลือก 2 ข้อต่อไปที่ 2/2 ระบุ error ที่ถูกต้อง (ย้ำ ให้เลือก 2 ข้อ ไม่ใช่ข้อเดียว)

คลาส Boss ไม่สามารถนิยามฟังก์ชัน Refill ซ้ำกับคลาส Monster ได้

คลาส Boss ต้องเรียกใช้ constructor ของคลาส Monster ด้วย ซึ่งในโค้ด คลาส Boss ไม่ได้เรียกใช้ ✓

ตัวแปร health ของคลาส Monster เป็น private ✓
คลาส Boss ไม่สามารถเรียกใช้ได้โดยตรง ทำให้เกิด error ที่บรรยาย 30

ตัวแปร power ของคลาส Monster เป็น protected
คลาส Boss ไม่สามารถเรียกใช้ได้โดยตรง ทำให้เกิด error ที่บรรยาย 27

- จัดโค้ดข้างต้น จะเห็นว่าคลาส Boss ไม่ได้นิยาม 1/1 ฟังก์ชัน GetHealth() ดังนั้น โค้ดด้านล่างจึงไม่ถูก ต้องใช้หรือไม่ จงเลือกข้อที่ถูกต้องที่สุด (สมมติว่าได้แก้ error เรียบร้อยแล้ว)

```
int main() {
    Boss boss1(10,10);
    cout << boss1.GetHealth() << endl;
}
```

โค้ด boss1.GetHealth() จะทำให้เกิด compile error
เนื่องจากคลาส Boss ไม่ได้นิยามฟังก์ชันนี้ในคลาส จึง
เรียกใช้ไม่ได้

โค้ด boss1.GetHealth() จะทำให้เกิด compile error
เนื่องจากคลาส Boss ได้รับฟังก์ชันนี้จาก base class แต่
ไม่สามารถเรียกใช้ได้ เพราะ access modifier ไม่
อนุญาต

โค้ด boss1.GetHealth() ถูกต้องแล้ว เนื่องจาก
คลาส Boss เป็น derived class ของคลาส Monster
ทำให้ได้รับทอดฟังก์ชัน GetHealth() มา และ
สามารถเรียกใช้ ✓

- โค้ด boss1.GetHealth() ถูกต้องแล้ว เนื่องจากฟังก์ชัน
GetHealth() เป็น public สามารถเรียกใช้จากอ้อมเจ็กต์
ของคลาสได ๆ ได้ที่ไม่ใช่ derived class

พิจารณาโค้ดต่อไปนี้ เพื่อตอบคำถามถัด ๆ ไป

```
1 #include <iostream>
2 using namespace std;
3
4 class Vehicle {
5 protected:
6     int distance, oil;
7 public:
8     Vehicle(int d, int o)
9         : distance(d), oil(o) {}
10
11    virtual int Drive() {
12        return distance;
13    }
14    int Fill() {
15        return oil;
16    }
17 };
18
19 class Car : public Vehicle {
20 public:
21     Car(int d, int o)
22         : Vehicle(d, o) {}
23
24     virtual int Drive() {
25         return distance + 1;
26     }
27     int Fill() {
28         return oil + 1;
29     }
30 };
```

- ✓ จากคลาสในโค้ดด้านบน จงตอบผลการรันฟังก์ชัน main() ต่อไปนี้ 1/1

- ✓ จากคลาสในโค้ดด้านบน จงตอบผลการรันฟังก์ชัน 1/1
main() ต่อไปนี้

```
int main() {
    Car vehicle1(10,10);
    Car *vehicle2 = &vehicle1;
    cout << vehicle2->Drive() << " "
        << vehicle2->Fill() << endl;
}
```

- 10 10
- 10 11
- 11 10
- 11 11



- ✓ จากคลาสในโค้ดด้านบน จงตอบผลการรันฟังก์ชัน 1/1
main() ต่อไปนี้

```
int main() {
    Car vehicle1(20,20);
    Vehicle *vehicle2 = &vehicle1; +1
    cout << vehicle2->Drive() << " "
        << vehicle2->Fill() << endl;
}
```

$vehicle1 = 20, 20$
 $*vehicle2 = 20, 20 \rightarrow fill()$
 $\downarrow drive()$
 $= 21$

- 20 20
- 20 21
- 21 20
- 21 21

