

- 01 - Polymorph Shape3D
- 02 - Polymorph Transportation Fare 1
- 03 - Polymorph Transportation Fare 2
- 04 - Polymorph Transportation Fare 3
- 05 - Polymorph Transportation Fare 4

CurrentRecentAll

### Polymorph Shape3D

จงเขียน โปรแกรมสำหรับคลาส Shape3D, RightRectangularPyramid และ Sphere ตามข้อกำหนดต่อไปนี้  
(ทั้งสามคลาส จะต้องมีคุณสมบัติ **polymorphic** จึงจะผ่าน โจทย์ข้อนี้)

คลาส Shape3D แทนคลาสของทรงสามมิติ ที่มี member function ได้แก่

- volumn() เป็น member function ที่คืนค่าปริมาตรของทรงสามมิติ
- surfaceArea() เป็น member function ที่คืนค่าพื้นที่ผิวทั้งหมดของทรงสามมิติ

คลาส Shape3D ไม่มี data member ไต ๆ และไม่มี constructor

จากนั้นใช้ Shape3D เป็น base class ของคลาส RightRectangularPyramid และคลาส Sphere

คลาส RightRectangularPyramid แทนคลาสของทรงพีระมิดตรงฐานสี่เหลี่ยมผืนผ้า

data member ประกอบด้วย

- width คือ ความกว้างของฐานสี่เหลี่ยมผืนผ้า
- length คือ ความยาวของฐานสี่เหลี่ยมผืนผ้า
- height คือ ความสูงตรงของพีระมิดตรง

member function ประกอบด้วย

- constructor สำหรับกำหนดค่า width, length และ height ตามลำดับ โดยมี default value เป็น 0 สำหรับแต่ละ parameter
- volumn() และ surfaceArea() สืบทอดจากคลาส Shape3D แต่ override การคำนวณสำหรับทรงพีระมิดตรงฐานสี่เหลี่ยมผืนผ้า

คลาส Sphere แทนคลาสของทรงกลม

data member ประกอบด้วย

- radius คือ ความยาวของรัศมีทรงกลม

member function ประกอบด้วย

- constructor สำหรับกำหนดค่า radius
- volumn() และ surfaceArea() สืบทอดจากคลาส Shape3D แต่ override การคำนวณสำหรับทรงกลม

[hide line #]

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

class Shape3D
{
public:
    ;
};

int main()
{
    int tc;
    cin >> tc;
    if (tc == 1) {
        Shape3D **shapes = new Shape3D*[3];
        shapes[0] = new RightRectangularPyramid(1, 5.2, 7);
        shapes[1] = new Sphere(10.6);
        shapes[2] = new RightRectangularPyramid(3, 3.5, 4.13);
        for (int i = 0; i < 3; i++) {
            cout << "Volumn: " << (*shapes)->volumn() << endl;
            cout << "Surface Area: " << (*shapes)->surfaceArea() << endl;
            cout << "-----\n";
            *shapes++;
        }
    } else if (tc == 2) {
        RightRectangularPyramid rectangle;
        Sphere sphere;
        cout << rectangle.volumn() << endl;
        cout << rectangle.surfaceArea() << endl;
        cout << sphere.volumn() << endl;
        cout << sphere.surfaceArea() << endl;
    } else if (tc == 3) {
        RightRectangularPyramid rectangle(171.5, 45.33, 31.02);
        cout << rectangle.volumn() << endl;
        cout << rectangle.surfaceArea() << endl;
    } else if (tc == 4) {
        Sphere sphere(34.25);
        cout << sphere.volumn() << endl;
        cout << sphere.surfaceArea() << endl;
    } else if (tc == 5) {
        RightRectangularPyramid rectangle(171.5, 45.33, 31.02);
        Sphere sphere(34.25);
        Shape3D *shape = &rectangle;
        cout << shape->volumn() << " " << shape->surfaceArea() << endl;
        shape = &sphere;
        cout << shape->volumn() << " " << shape->surfaceArea() << endl;
    } else if(tc == 6) {
    }

}
```

Submit

- 01 - Polymorph Shape3D
- 02 - Polymorph Transportation Fare 1
- 03 - Polymorph Transportation Fare 2
- 04 - Polymorph Transportation Fare 3
- 05 - Polymorph Transportation Fare 4

Current Recent All

## Transportation Fare part 1

คลาส Transportation เป็นคลาสแทนการให้บริการรถโดยสารชนิดต่าง ๆ มีพฤติกรรม

- fare() คำนวณค่าใช้จ่ายในการให้บริการรถโดยสารชนิดต่าง ๆ

คลาส Taxi เป็นคลาสแทนการให้บริการรถแท็กซี่ ซึ่งสืบทอดจากคลาส Transportation และให้มีสมบัติ polymorphism

การคำนวณค่าใช้จ่ายในการให้บริการรถแท็กซี่ จะเริ่มต้นที่ 35 บาท และเพิ่มขึ้นตามระยะทางกิโลเมตรละ 2 บาท

Taxi มี method setDistance(double) ที่รับระยะทางหน่วยกิโลเมตร

และมี method fare() ที่คำนวณค่าใช้จ่ายในการให้บริการรถแท็กซี่

จงเขียนคลาส Transportation และคลาส Taxi ให้เป็นไปตามข้อกำหนดข้างต้น

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

class Transportation
{
public:
    fare() = 0;
};

class Taxi
{
};

int main()
{
    int tc;
    cin >> tc;
    if (tc == 1) {
        double distance;
        cin >> distance;
        Taxi taxi;
        taxi.setDistance(distance);
        cout << "Taxi fare: "
              << taxi.fare() << endl;
    } else if (tc == 2) {
        double distance;
        cin >> distance;
        Taxi taxi;
        taxi.setDistance(distance);
        Transportation *transport = &taxi;
        cout << "Transportation fare: "
              << transport->fare() << endl;
    }
}
```

- 01 - Polymorph Shape3D
- 02 - Polymorph Transportation Fare 1
- 03 - Polymorph Transportation Fare 2
- 04 - Polymorph Transportation Fare 3
- 05 - Polymorph Transportation Fare 4

Current Recent All

## Transportation Fare part 2

คลาส Transportation เป็นคลาสแทนการให้บริการรถโดยสารชนิดต่าง ๆ มีพฤติกรรม

- fare() คืนค่า ค่าใช้จ่ายในการใช้บริการรถโดยสารชนิดต่าง ๆ

คลาส BmtaBus เป็นคลาสแทนการให้บริการรถขสมก. ซึ่งสืบทอดจากคลาส Transportation และให้มีสมบัติ polymorphism

การคำนวณค่าใช้จ่ายในการใช้บริการรถขสมก. ครั้งละ 6.50 บาท ผ่าน method fare()

จงเขียนคลาส Transportation และคลาส BmtaBus ให้เป็นไปตามข้อกำหนดข้างต้น

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

class Transportation
{
public:
    fare() = 0;
};

class BmtaBus
{
};

int main()
{
    int tc;
    cin >> tc;
    if (tc == 1) {
        BmtaBus bus;
        cout << "BMTA Bus fare: "
              << bus.fare() << endl;
    } else if (tc == 2) {
        BmtaBus bus;
        Transportation *transport = &bus;
        cout << "Transportation fare: "
              << transport->fare() << endl;
    }
}
```

[hide line #]

Submit

- 01 - Polymorph Shape3D
- 02 - Polymorph Transportation Fare 1
- 03 - Polymorph Transportation Fare 2
- 04 - Polymorph Transportation Fare 3
- 05 - Polymorph Transportation Fare 4

Current Recent All

### Transportation Fare part 3

คลาส Transportation เป็นคลาสแทนการให้บริการรถโดยสารชนิดต่าง ๆ มีพฤติกรรม

- fare() คำนวณค่าใช้จ่ายในการให้บริการรถโดยสารชนิดต่าง ๆ

คลาส BTS เป็นคลาสแทนการให้บริการรถไฟฟ้าบีทีเอส ซึ่งสืบทอดจากคลาส Transportation และให้มีสมบัติ polymorphism

การคำนวณค่าใช้จ่ายในการให้บริการรถไฟฟ้าบีทีเอส จะเริ่มต้นที่ 15 บาท และเพิ่มขึ้นตามจำนวนสถานี สถานีละ 5 บาท

คลาส BTS มี method setStation(int) ที่รับจำนวนสถานีที่จะโดยสารไป และมี method fare() ที่คำนวณค่าใช้จ่ายในการให้บริการรถไฟฟ้าบีทีเอส

จงเขียนคลาส Transportation และคลาส BTS ให้เป็นไปตามข้อกำหนดข้างต้น

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

class Transportation
{
public:
    fare() = 0;
};

class BTS
{
    // ...
};

int main()
{
    int tc;
    cin >> tc;
    if (tc == 1) {
        int station;
        cin >> station;
        BTS bts;
        bts.setStation(station);
        cout << "BTS fare: "
             << bts.fare() << endl;
    } else if (tc == 2) {
        int station;
        cin >> station;
        BTS bts;
        bts.setStation(station);
        Transportation *transport = &bts;
        cout << "Transportation fare: "
             << transport->fare() << endl;
    }
}
```

[hide line #]

- 01 - Polymorph Shape3D
- 02 - Polymorph Transportation Fare 1
- 03 - Polymorph Transportation Fare 2
- 04 - Polymorph Transportation Fare 3
- 05 - Polymorph Transportation Fare 4

Current Recent All

Transportation Fare part 4

คลาส Transportation เป็นคลาสแทนการให้บริการรถโดยสารชนิดต่าง ๆ มีพฤติกรรม

- fare() คำนวณค่า ค่าใช้จ่ายในการให้บริการรถโดยสารชนิดต่าง ๆ

คลาส Taxi เป็นคลาสแทนการให้บริการรถแท็กซี่ ซึ่งสืบทอดจากคลาส Transportation และให้มีสมบัติ polymorphism

การคำนวณค่าใช้จ่ายในการให้บริการรถแท็กซี่ จะเริ่มต้นที่ 35 บาท และเพิ่มขึ้นตามระยะทางกิโลเมตรละ 2 บาท

Taxi มี method setDistance(double) ที่รับระยะทางหน่วยกิโลเมตร และมี method fare() ที่คำนวณค่า ค่าใช้จ่ายในการให้บริการรถแท็กซี่

คลาส BmtaBus เป็นคลาสแทนการให้บริการรถขสมก. ซึ่งสืบทอดจากคลาส Transportation และให้มีสมบัติ polymorphism

การคำนวณค่าใช้จ่ายในการให้บริการรถขสมก. ครั้งละ 6.50 บาท ผ่าน method fare()

คลาส BTS เป็นคลาสแทนการให้บริการรถไฟฟ้าบีทีเอส ซึ่งสืบทอดจากคลาส Transportation และให้มีสมบัติ polymorphism

การคำนวณค่าใช้จ่ายในการให้บริการรถไฟฟ้าบีทีเอส จะเริ่มต้นที่ 15 บาท และเพิ่มขึ้นตามจำนวนสถานี สถานีละ 5 บาท

คลาส BTS มี method setStation(int) ที่รับจำนวนสถานีที่จะโดยสารไป และมี method fare() ที่คำนวณค่า ค่าใช้จ่ายในการให้บริการรถไฟฟ้าบีทีเอส

คลาส Passenger แทนคลาสของผู้โดยสาร มี method ดังนี้

- void addTransportation(Transportation &) ผู้โดยสารเลือกใช้บริการรถโดยสารแบบใดก็ได้จากรถแท็กซี่ รถขสมก. หรือรถไฟฟ้าบีทีเอส และคำนวณค่าใช้จ่ายรวมในการโดยสารรถทุกแบบ
- double getTotalFare() คำนวณค่า ค่าใช้จ่ายรวมในการโดยสารรถทุกแบบ

จงเขียนโปรแกรมเพื่อคำนวณค่าใช้จ่ายในการเดินทางของผู้โดยสารคนหนึ่ง โดยรับข้อมูลการให้บริการแบบต่าง ๆ ดังนี้

- A แทนการเดินทางด้วยรถแท็กซี่ และรับระยะทางที่โดยสารด้วยรถแท็กซี่ครั้งนั้น
- B แทนการเดินทางด้วยรถขสมก.
- C แทนการเดินทางด้วยรถไฟฟ้าบีทีเอส และรับจำนวนสถานีที่โดยสารด้วยรถไฟฟ้าบีทีเอสครั้งนั้น
- การโดยสารรูปแบบอื่นถือว่าสิ้นสุดการเดินทาง เมื่อสิ้นสุดการเดินทางให้แสดงผลรวมของค่าโดยสารทั้งหมด

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

class Transportation
{
public:
    fare() = 0;
};

class Taxi
{
    // ...
};

class BmtaBus
{
    // ...
};

class BTS
{
    // ...
};

class Passenger
{
    // ...
};

int main()
{
    Passenger passenger;
    while(1) {
        char transportType;
        cin >> transportType;
        if (transportType == 'A') {
            Taxi *taxi = new Taxi();
            double distance;
            cin >> distance;
            // ...
        } else if (transportType == 'B') {
            BmtaBus *bus = new BmtaBus();
            // ...
        } else if (transportType == 'C') {
            BTS *bts = new BTS();
            int station;
            cin >> station;
            // ...
        } else {
            break;
        }
    }
    cout << "Total Fare: " << passenger.getTotalFare() << endl;
}
```

Submit