

Code

```
#include <iostream>
```

```
#include <fstream>
```

```
#include <string>
```

```
using namespace std;
```

```
class Fitness { // base class
```

```
public:
```

```
    string name;
```

```
    int age;
```

```
    double height;
```

```
    double weight;
```

```
    int vip;
```

```
    double price;
```

```
public:
```

```
    Fitness(string n, int a, double h, double w, double p) {
```

```
        name = n;
```

```
        age = a;
```

```
        height = h;
```

```
        weight = w;
```

```

        vip = 0;

        price = p;
    }

Fitness(string n, int a, double h, double w, int v, double p) {

    name = n;

    age = a;

    height = h;

    weight = w;

    vip = v;

    price = p;
}

Fitness() {

    name = "Anonymous";

    age = 0;

    height = 0;

    weight = 0;

    vip = 0;

    price = 0;
}

virtual void display() {

    cout << "Name : " << name << endl;

```

```

        cout << "Age : " << age << endl;

        cout << "Height : " << height << " meter " << endl;

        cout << "Weight : " << weight << " kilograms " << endl;

        cout << "Price : " << price << " baht " << endl;

    }

    virtual void calculateBMI() {

        double bmi = weight / (height * height);

        cout << "BMI : " << bmi << endl;

        cout << "-----" << endl;

    }

};

```

```

class ExercisePlan : public Fitness { // Single Inheritance

public:

    int exerciseTime;

public:

    ExercisePlan(string n, int a, double h, double w, double p, int et) : Fitness(n, a, h, w, p) {

        exerciseTime = et;

    }

}

```

```

        ExercisePlan(string n, int a, double h, double w, int v, double p, int et) : Fitness(n, a, h, w,
p) {

        exerciseTime = et;

    }

    void display() { // Scope resolution operator

        Fitness::display();

        cout << "Exercise Time : " << exerciseTime << " minutes" << endl;

    }

};

```

```

class DietPlan : public Fitness { // Single Inheritance

public:

    int calorielntake;

public:

    DietPlan(string n, int a, double h, double w, int v, double p, int ci) : Fitness(n, a, h, w, p) {

        calorielntake = ci;

    }

    void display() {

        Fitness::display(); // Scope resolution operator

        cout << "Calorie Intake : " << calorielntake << " calories" << endl;

    }

}

```

```
};
```

```
void writeFitnessPlanToFile(Fitness* plan, ofstream& outFile) {
```

```
    outFile << "Name : " << plan->name << endl;
```

```
    outFile << "Age : " << plan->age << endl;
```

```
    outFile << "Height : " << plan->height << " meter" << endl;
```

```
    outFile << "Weight : " << plan->weight << " kilograms" << endl;
```

```
    if (ExercisePlan* exercisePlan = dynamic_cast<ExercisePlan*>(plan)) {
```

```
        outFile << "Exercise Time : " << exercisePlan->exerciseTime << " minutes" << endl;
```

```
    }
```

```
    else if (DietPlan* dietPlan = dynamic_cast<DietPlan*>(plan)) {
```

```
        outFile << "Calorie Intake : " << dietPlan->calorieIntake << " calories" << endl;
```

```
    }
```

```
    outFile << "BMI : " << (plan->weight) / (plan->height * plan->height) << endl;
```

```
    outFile << "price : " << plan->price << " baht" << endl;
```

```
}
```

```
int main() {
```

```
    ExercisePlan exercisePlan("Chonnikarn", 20, 1.5 , 46, 1200, 60); // Overloading
```

```
ExercisePlan exercisePlanVIP("Natnicha", 20, 1.65, 70, 1, 9600, 200); // Overloading
```

```
DietPlan dietPlanVIP("Natnicha", 20, 1.65, 70, 1, 9600, 1700); // Overloading
```

```
// write to file
```

```
ofstream outFile;
```

```
outFile.open("fitness_output.txt");
```

```
// Exercise Plan
```

```
exercisePlan.display();
```

```
exercisePlan.calculateBMI();
```

```
outFile << "Exercise Plan :\n";
```

```
outFile << "-----\n";
```

```
writeFitnessPlanToFile(&exercisePlan, outFile);
```

```
outFile << "\n";
```

```
// Exercise Plan VIP
```

```
exercisePlanVIP.display();
```

```
exercisePlanVIP.calculateBMI();
```

```
outFile << "\nExercise Plan VIP :\n";
```

```

outFile << "-----\n";

writeFitnessPlanToFile(&exercisePlanVIP, outFile);

outFile << "\n";


// Diet Plan VIP

dietPlanVIP.display();

dietPlanVIP.calculateBMI();


outFile << "\nDiet Plan VIP :\n";

outFile << "-----\n";

writeFitnessPlanToFile(&dietPlanVIP, outFile);

outFile << "\n";


// Close the file

outFile.close();

return 0;

}


/* นางสาวชลนิกานต์ ส้างวัง 6530200070 sec830

นางสาวณัฐนิชา นนตระอุดร 6530200134 sec830 */

```

Output หน้าจอการทำงาน

```
Name : Chonnikarn
Age : 20
Height : 1.5 meter
Weight : 46 kilograms
Price : 1200 baht
Exercise Time : 60 minutes
BMI : 20.4444
-----
Name : Natnicha
Age : 20
Height : 1.65 meter
Weight : 70 kilograms
Price : 9600 baht
Exercise Time : 200 minutes
BMI : 25.7117
-----
Name : Natnicha
Age : 20
Height : 1.65 meter
Weight : 70 kilograms
Price : 9600 baht
Calorie Intake : 1700 calories
BMI : 25.7117
-----
```

```
main.cpp  fitness_output.txt :
1  Exercise Plan :
2  -----
3  Name : Chonnikarn
4  Age : 20
5  Height : 1.5 meter
6  Weight : 46 kilograms
7  Exercise Time : 60 minutes
8  BMI : 20.4444
9  price : 1200 baht
10
11
12  Exercise Plan VIP :
13  -----
14  Name : Natnicha
15  Age : 20
16  Height : 1.65 meter
17  Weight : 70 kilograms
18  Exercise Time : 200 minutes
19  BMI : 25.7117
20  price : 9600 baht
21
22
23  Diet Plan VIP :
24  -----
25  Name : Natnicha
26  Age : 20
27  Height : 1.65 meter
28  Weight : 70 kilograms
29  Calorie Intake : 1700 calories
30  BMI : 25.7117
31  price : 9600 baht
```

Link project : <https://www.onlinegdb.com/S0KUMb6nH>