4-8

```cpp
#include <bits/stdc++.h>

using namespace std;

class Dog
{
public:
    Dog();
    void setAge(int a);
    void setWeight(double w);
    int getAge();
    double getWeight();
private:
    int age;
    double weight;
};

Dog::Dog() {
    age = 0;
    weight = 0;
}
void Dog::setAge(int a) {
    age = a;
}
void Dog::setWeight(double w) {
    weight = w;
}
int Dog::getAge() {
    return age;
}
double Dog::getWeight() {
    return weight;
}


//
// Created by xiwen on 2023/4/10.
//

```

4-9

```cpp
#include <bits/stdc++.h>

using namespace std;

typedef pair<int, int> PII;

class Rectangle {
public:
    Rectangle();
    int getSpace();
    void setSpot(PII s, PII ss);

private:
    PII up;
    PII down;
};

Rectangle::Rectangle() {
    up = {10, 10};
    down = {20, 20};
}

int Rectangle::getSpace() {
    int space = (down.first - up.first) * (down.second - up.second);
    return space;
}
void Rectangle::setSpot(PII s, PII ss) {
    up = s;
    down = ss;
}

```

4-10

```
#include <bits/stdc++.h>
#include <utility>

using namespace std;

class Date {
    int year, month, day;
public:
    Date() = default;
    Date(int y, int m, int d);
    void setDate(int y, int m, int d);
    string getData() const;
};
Date::Date(int y, int m, int d) {
    year = y, month = m, day = d;
}

void Date::setDate(int y, int m, int d) {
    year = y, month = m, day = d;
}

string Date::getData() const {
    string date = to_string(year) + "." + to_string(month) + "." + to_string(day);
    return date;
}

class Staff {
    int code, gender;
    Date* birthday{};
    string id;
public:
    Staff(): birthday(new Date()) {
        this->code = 100;
        this->gender = 1;
        this->id = "3233";
    }
    ~Staff() {
        cout<<"staff destructed"<<endl;
    }
    Staff(Staff& s) {
        cout<<"copy"<<endl;
    }
    void setInfo(string id, Date *day, int code = 1, int gender = 1);
    void showInfo();
};

void Staff::setInfo(string id_ref, Date *day, int code_ref, int gender_ref) {
    this->id = std::move(id_ref); // 右值引用用以高效传值
    this->gender = gender_ref;
    this->code = code_ref;
    birthday = day;
}

void Staff::showInfo() {
    cout<<"code:"<<code<<endl;
    cout<<"gender:"<<(gender == 1 ? "male" : "female")<<endl;
    cout<<"code:"<<code<<endl;
    cout<<"birthday:"<<birthday->getData()<<endl;
}
```

4-11+

```
#include <bits/stdc++.h>

using namespace std;
//4-11
class Rectangle{
    int len, width;
public:
    Rectangle();
    int getSpace();
};

Rectangle::Rectangle() {
    len = 10;
    width = 10;
}

int Rectangle::getSpace() {
    return len * width;
}

//4-12
class DataType {
    string type;
public:
    DataType(int data);
    DataType(char data);
    DataType(double data);
    string getType();
};
DataType::DataType(int data) {
    type = "int";
}
DataType::DataType(char data) {
    type = "char";
}
DataType::DataType(double data) {
    type = "double";
}
string DataType::getType() {
    return type;
}

//4-13

class Circle {
    int radius;
public:
    Circle(int r): radius(r) {};
    double getArea();
};

double Circle::getArea() {
    return 3.1415 * radius * radius;
}

//4-14

class Tree {
    int ages;
public:
    Tree() = default;
    void grow(int years);
    void age() {
        cout<<ages<<endl;
    }
};

void Tree::grow(int years) {
    ages += years;
}

```

4-19

```
#include <bits/stdc++.h>

using namespace std;

enum Core {Single, Dual, Quad};
enum Words {Bits32, Bits64};
enum HyperThread {Support, NotSupport};

```

```cpp
class CPU {
public:
    CPU(unsigned frequence, Core type, Words length, HyperThread mode): frequence(frequence), CoreType(type), WordLen(length), mode(mode){}
    void show();
private:
    unsigned frequence: 32;
    Core CoreType: 3;
    Words WordLen: 2;
    HyperThread mode: 2;
};

void CPU::show() {
    cout<< "frequence:"<<frequence<<endl;
    cout<<"core:";
    switch ((unsigned)CoreType) {
        case Single: cout<<"single-core"; break;
        case Dual: cout<<"dual-core"; break;
        case Quad: cout<<"quad-core"; break;
    }
    cout<<endl;
    cout<<"word:";
    switch ((unsigned)WordLen) {
        case Bits32: cout<<"32bits"; break;
        case Bits64: cout<<"64bits"; break;
    }
    cout<<endl;
    cout<<"hyperthreads:";
    switch (mode) {
        case Support: cout<<"support"; break;
        case NotSupport: cout<<"not support"; break;
    }
    cout<<endl;
}

int main() {
    CPU c(3000000000, Quad, Bits64, Support);
    cout<<"size of cpu: "<<sizeof(CPU)<<endl;
    c.show();
    return 0;
}
```

4-20

```cpp
#include <bits/stdc++.h>

using namespace std;

class Complex {
    double real;
    double i;
public:
    Complex(double a, double b): real(a), i(b){}
    Complex(double a): real(a), i(0){}
    void add(Complex c) {
        real += c.real;
        i += c.i;
    }
    void show() {
        cout<<real<<'+'<<i<<'i'<<endl;
    }
};

int main()
{
    Complex c1(1.5, 2.5);
    Complex c2 = 4.5;
    c1.show();
    c1.add(c2);
    c1.show();

    return 0;
}
```