

1. 类与结构体

类的定义：

```
class Person
{
    private:
        int age, height;
        double money;
        string books[100];

    public:
        string name;

        void say()
        {
            cout << "I'm " << name << endl;
        }

        int get_age()
        {
            return age;
        }

        void add_money(double x)
        {
            money += x;
        }
};
```

类中的变量和函数被统一称为类的成员变量。

`private` 后面的内容是私有成员变量，在类的外部不能访问；`public` 后面的内容是公有成员变量，在类的外部可以访问。

类的使用：

```

#include <iostream>

using namespace std;

const int N = 1000010;

class Person
{
private:
    int age, height;
    double money;
    string books[100];

public:
    string name;

    void say()
    {
        cout << "I'm " << name << endl;
    }

    int set_age(int a)
    {
        age = a;
    }

    int get_age()
    {
        return age;
    }

    void add_money(double x)
    {
        money += x;
    }
} person_a, person_b, persons[100];

int main()
{
    Person c;

    c.name = "yxc";           // 正确! 访问公有变量
    c.age = 18;               // 错误! 访问私有变量
    c.set_age(18);           // 正确! set_age()是共有成员变量
    c.add_money(100);

    c.say();
    cout << c.get_age() << endl;

    return 0;
}

```

结构体和类的作用是一样的。不同点在于类默认是 `private` ，结构体默认是 `public` 。

```
struct Person
{
    private:
        int age, height;
        double money;
        string books[100];

    public:
        string name;

        void say()
        {
            cout << "I'm " << name << endl;
        }

        int set_age(int a)
        {
            age = a;
        }

        int get_age()
        {
            return age;
        }

        void add_money(double x)
        {
            money += x;
        }
} person_a, person_b, persons[100];
```

2. 指针和引用

指针指向存放变量的值的地址。因此我们可以通过指针来修改变量的值。

```
#include <iostream>

using namespace std;

int main()
{
    int a = 10;
    int *p = &a;

    *p += 5;
    cout << a << endl;

    return 0;
}
```

数组名是一种特殊的指针。指针可以做运算：

```
#include <iostream>

using namespace std;

int main()
{
    int a[5] = {1, 2, 3, 4, 5};

    for (int i = 0; i < 5; i ++ )
        cout << *(a + i) << endl;

    return 0;
}
```

引用和指针类似，相当于给变量起了个别名。

```
#include <iostream>

using namespace std;

int main()
{
    int a = 10;
    int &p = a;

    p += 5;
    cout << a << endl;

    return 0;
}
```

3. 链表

```
#include <iostream>

using namespace std;

struct Node
{
    int val;
    Node* next;
} *head;

int main()
{
    for (int i = 1; i <= 5; i ++ )
    {
        Node* p = new Node();
        p->val = i;
        p->next = head;
        head = p;
    }

    for (Node* p = head; p; p = p->next)
        cout << p->val << ' ';
    cout << endl;

    return 0;
}
```