**C++重载**

**一、初始重载**

1. 重载：
   1. 函数名相同
   2. 传进去的参数、列表、类型不同
   3. 严格匹配，找到则进行匹配
2. 简单实现
   1. 比如我们有print函数第一个接收 int 第二个接收 char
      1. void print(int a){

cout << “this is int ”;

}

* + 1. void print(char b){

cout << “this is char ”;

}

* + 1. int main(){

int a = 10;

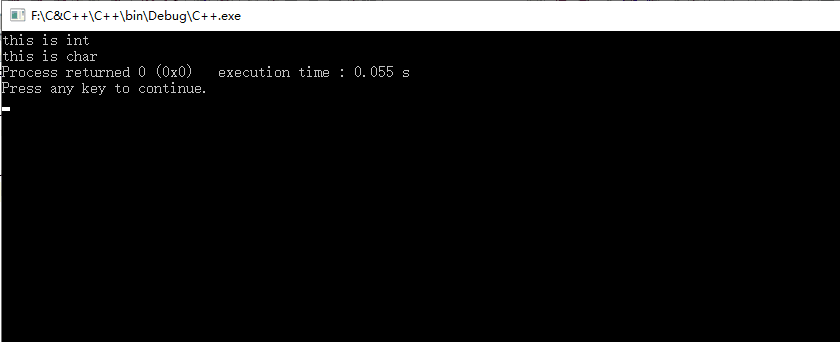
char b = ‘c’;

print(a);

print(b);

}

* + 1. 运行截图



1. **操作符的重载，我们只举例加号**
2. ’+’ 重载
   * 1. 现在我们有一个需求，有两个学生结构体，里面有 name 和 score 那我们现在要求 + 能够进行两个结构体之间的分数相加
        1. 结构体

struct Student{

string name;

float score;

}

* + - 1. 那我们现在就对加号进行重载一下

//首先我们需要返回的是float 因为我们只要分数就够了

float operator+(Student s1, Student s2){

float sum;

sum = s1.score + s2.score;

return sum;

}

1. **我们来对iostream 的cout 来进行重载一下，让它能支持Student的输出**
2. 既然我们学习了重载，我们就不想用cout << student.name << “ ” << student.score << endl;
3. 那我们应该对 cout 进行重载
   * 1. 代码

ostream& operator<<(ostream &os,const Student &s){

os << s.name << " " << s.score << endl;

}

* + 1. 那下面大家就可以进行输出看一下了