

Notepad instead:

- 1)、cin 单个变量
- 2)、cin.get() 单个字符 cin.get(字符数组名,接收字符数目)用来接收一行字符串,可以接收空格
- 3)、cin.getline() cin.getline(字符数组名,接受个数,/结束字符)
- 4)、getline() (字符数组名,接受个数,/结束字符)
 getline() (输入流, 字符数组名。/结束字符)
- 5)、gets()

endl输出一个换行符, 并立即刷新缓冲区

C++的algorithm库常用函数

swap(a,b);

max(a,b);

min(a,b);

abs(x); x = int

sort(a,a+n,/cmp);/比较函数 返回值为bool

qsort(a,a+n,sizeof(a[0]),cmp);

vector (v.begin(),v.end())

lower_bound(a,a+n,x) 查找大于或等于x的第一个位置的指针 一般-a得到位置

upper_bound(a,a+n,x)查找大于的第一个位置的指针 一般-a得到位置

fill(a,a+n,x)

reverse(a,a+n)

find(a,a+n,x) 返回元素位置指针 未找到返回a+n

struct 里可以有函数, 并可有定义, 无返回值, 称其为构造函数 结构体成员函数中this 指向当前对象的指针

函数可以拥有默认值,默认参数靠右

模板 template<typename T>

string.size() string.length()

容器

namespace 变量空间 防止名字冲突

:: 限定变量属于哪个名字空间

```

namespace first
{
    int a;
}
namespace second
{
    int a;
}

```

```

first::a=1;
second::a=2;

```

using namespace first;引用整个名字空间 后变量为first空间

using first::a; 后 a为first的

C++的标准库名字都属于名字空间 std,所以一般开始 using namespace std;

C++变量即用即定义定义

for类似C99

访问内部同名全局变量::

引用类型 另一个变量的别名 & (不是取地址)

& x x为引用变量 常作为形参修改实参

```

void swap(int &a,int &b)

```

```

{
    int i;
    i=a;
    a=b;
    b=i;
}

```

inline 函数定义开头 内联函数 编译时类似于预处理替换代码,降低函数调用开销

try-catch技术

```

try
{
    if (...)
    {
        throw ...100
    }
}

```

```
}  
catch(int...)  
{
```

```
}
```

函数重载 参数不同（函数返回值不同无法判断） 函数名同 根据变量类型调用对应函数
struct 不用 typedef

运算符重载（运算符可以看作一个函数）

可以在类的内部 也可以在外部 但注意函数原型的改变（最好外部）

```
T operator + - * << >> ()
```

```
{
```

```
    return ;
```

```
}
```

```
ostream& operator << (operator & o,v data)
```

```
{
```

```
    o << v.s;
```

```
    return o;
```

```
}
```

```
istream& operator >> (istream & in,v& data)
```

```
{
```

```
    in >> v.s >> v.d;
```

```
    return in;
```

```
}
```

```
struct v
```

```
{
```

```
    double x;
```

```
    double y;
```

```
};
```

```
v operator * (double a,v b)
```

```

{
    v r;
    r.x = a * b.x;
    r.y = a * b.y;
    return r;
}
v operator + (v a,v b)
{
    v r;
    r.x=a.x+b.x;
    r.y=a.y+b.y;
    return r;
}
v operator - (v a ,v b)
ostream& operator << (ostream& o,v a)
{
    o <<"("<<a.x<<")"<<"("<<a.y<<")";
    return o;
}

```

模板 template 定义函数 替代类型

```

template<class T>
template<class T1,class T2,,,,,>
T min(T a,T b)
{

```

```

    return a>b?a:b;
}

```

使用时自动生成对应类型的具体函数

动态内存分配 new,delete;

```
int *p;
```

```
p = new int;
```

```
p = new int[100];
```

delete p;释放第一个

delete []p;释放全部

p = new [n];

char *p;

p = new char[n];

n可以为变量

类：类似于结构类型，增加了成员函数。（面向对象思想）struct 成员默认公有 class 成员默认私有 用private 私有，public设为公有//构成接口

用户定义类型（自定义类）

可在内部定义函数原型 外部编写函数体

this 类的成员函数隐含一个指针 指向调用成员函数的对象的指针

类的成员函数一些特性

默认构造函数：创建类对象；； 无任何参数，函数名和类名相同 无返回值 时自动调用这个函数

构造函数：参数自定义 构造对象同形式

friend 友元函数 在类内部声明 friend 函数原型 函数在外部使用时可访问私有函数

拷贝构造函数 复制对象的时候自动调用//硬拷贝

类名（类名&）

{

}

析构函数 销毁时自动调用 ~类名（）{

}

容器：

向量：vector<int,double,string> x; x.size() x.resize x.push_back() x.pop_back
x.empty() x.clear()....

集合：set

映射：map

栈

队列

优先队列

