

# **CPP Primer Note**



# Contents

<b>Chapter 1: Pass Array Parameters.....</b>	<b>5</b>
<b>Chapter 2: Add Parameters to CMD via the main() Function.....</b>	<b>7</b>
<b>Chapter 3: Values Returned.....</b>	<b>9</b>
<b>Chapter 4: Include &lt;fstream&gt;.....</b>	<b>11</b>



---

# Chapter 1

---

## Pass Array Parameters

---

二维

定义:

- `void printValues(int *x, size_type size){}`
- `void printValues(int x[], size_type size){}`
- `void printValues(int x[10], size_type size){}`

调用: `printValues(arr,8);`

三维

定义: `void printValues_2(int (*x)[10],int rowSize){}`

调用:

```
int m[][10] =
{
    {1,1,1,1,1,1,1,1,1,1},
    {1,1,1,1,1,1,1,1,1,1},
    {1,1,1,1,1,1,1,1,1,1}
};
printValues_2(m,3);
```

C++标准库的遍历，以二维为例

定义: `void printValues_3(const int *beg, const int *end){};`

调用: `printVaues_3(arr,arr+列数);`

以C风格传字符串，以二维为例

定义:

```
void printChars(const char *x)
{
    while (*x != NULL)
        cout << *x++;
    cout << endl;
};
```

调用: `printChars(s);`



---

# Chapter

# 2

---

## Add Parameters to CMD via the main() Function

---

```
int main(int argc, char **argv) //第二个参数是指向指针的指针，等同于*argv[]
{
    if (argc != 3)
    {
        cout << "Must contain three arguments!" << endl;
        return -1;
    }
    cout << "SUM is:"
         << (atof(argv[1]) + (atof(argv[2]))) << endl;
}
```





---

# Chapter

# 3

---

## Values Returned

---

```
return 0; //函数结束，main()函数一般以此结尾
```

```
return -1; //返回错误值，只要不是0，效果都一样
```

```
return; //void()函数专用，可以不写
```



### Note:

- 不可返回局部变量的引用
- 不可返回指向局部对象的指针



---

# Chapter

# 4

---

## Include <fstream>

---

```
#include <fstream>
```

**Table 1: ofstream**基本语句

语句	含义
<code>ofstream outfile("文件名");</code>	输出一个文件，名字为“文件名”。
<code>outfile &lt;&lt; ("内容");</code>	将“内容”写入“文件名”文件。
<code>outfile.close();</code> <code>outfile.clear();</code>	关闭输出流。

**Table 2: ifstream**基本语句

语句	含义
<code>ifstream infile("文件名");</code>	读取“文件名”文件，此时“文件名”与infile已形成绑定关系。
<code>string s;</code> <code>while (infile &gt;&gt; s, !</code> <code>infile.eof);</code> <code>infile.close();</code> <code>infile.clear();</code>	将“文件名”的“内容”写入字符串变量s中。
<code>string s;</code> <code>ifstream infile(s.c_str());</code> <code>infile.close();</code> <code>infile.clear();</code>	同上，不同的写法
<code>ifstream infile;</code> <code>string s;</code> <code>infile.open(s.c_str());</code> <code>//</code> <code>若是指针，用it-&gt;c_str();</code>	同上，不同写法。目的是不在开 端绑定infile和文件。

往向量里塞文件

```
#include <iostream>
#include <fstream>
```

```

#include <vector>
#include <string>

vector<string> files;
files.push_back("文件1");
files.push_back("文件2");
files.push_back("文件3");
vector<string>::const_iterator itr =
    files.begin();
while (itr != files.end())
{
    ifstream infile(itr->c_str()); //C语言的传参风格，历史遗留问题
    if (!infile)
    {
        cerr << "Error" << *itr << endl;
        infile.clear();
        ++itr;
        continue;
    }
    string s;
    while (infile >> s)
        process s; //在头文件定义的void()类型函数，等同于cout
    infile.close();
    infile.clear();
}

```