1. 在用函数function时应对函数进行声明，如：int Square(int)或void Cube(int)等等，后面还应对函数进行具体定义。如：int Square(int n){return n\*n；}或void Cube(int n){cout<<”I have ”<<n<<” books.”;}。
2. 命名空间std包含许多编程用语，如:”cout”、”endl”等，除了用using namespace std进行声明外还可不嫌麻烦的用”std::cout”、”std::endl”等。
3. Char是个int型，如”int n=char ch;”是没问题的，其中n值是ch的ASC码，反过来用cout.put(n)得到n对应的ch字符。
4. 对很小或很大的数可用科学技术法表示，如1.23E+08、1.23E-07，通常受计算机系统限制只显示有限个位数。另外，const常量通常用大写字母变量，如const float MAX\_SCORE=100.0。
5. 介绍几种常用的函数：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 头文件 | 函数 | 变量类型 | 函数值类型 | 函数说明 |
| <cstdlib> | abs(x) | int | int | 算int 数的绝对值 |
| <cmath> | cos(x) | float | float | 算弧度的余弦值 |
| <cmath> | fabs(x) | float | float | 算float数的绝对值 |
| <cstdlib> | labs(x) | long | long | 算long数的绝对值 |
| <cmath> | pow(a,b) | a是float， b是 int | float | 算float(如3.0)的int(如16)次幂 |
| <cmath> | sin(x) | float | float | 算弧度的正弦值 |
| <cmath> | sqrt(x) | float | float | 算float数的二次方根 |

6、在头文件<iomanip>中有个函数volid setw(int)对数和字符串起作用，控制输出值前的空格数，如float n=4.85;cout<<setw(5)<<n;则n前有一个空格；此例中当函数变量<=4时，输出值前没有空格，而且是一次性操作，一次操作后归0。也是在这个文件中，setprecision(int)强制结果保留几位小数，变量是几，就保留几位小数。

1. 在头文件<iostream>中fixed的作用是强制非常大的或非常小的数不以科学技术法的形式输出，并且保证小数点后有六位小数。也是在这个文件头里，showpoint用来强制浮点数小数点后显示结尾0，除非该小数包括小数点总位数>=7,在这种情况下showpoint失去作用。
2. 下面是上面的表格化小结：

|  |  |
| --- | --- |
| 头文件 | 操作 |
| <iostream> | showpoint |
| <iostream> | fixed |
| <iomanip> | setw(int) |
| <iomanip> | setprecision(int) |

1. 计算string的长度时用string.size()或string.length()；有一个奇怪的变量类型string::size\_type ，用来表示string的长度，不管有没有头文件<string>，使用时都要出现string::size\_type，如string str;string::size\_type len=str.length();等。

10、.find()用来表示两个字符串的关系，如：str1=”12345678”;str2=”67”;int n=str1.find(str2);则n=5等等。因为第一位从0开始，而且该函数显示字符串第一个字母的位置；如果没有找到，则显示一个常量，当然函数括号里也可以是字符，情况和上面相同。另外.substr(x,y)用来截取字符，x是第一个字符的位置，y是子字符串的长度，两者都是string::size\_type类型的变量。如str1=”123456789”;str2=str1.substr(4,3)=“567“;如果y足够大，则可以取到母字符的整个后半截，且y可以很大。

11、在读取字符(串)的时候，比如string类的变量，绝不会包含空格字符，也就是说当输入流中遇到空格，就意味着读取的终止。但是，cin.get(char ch)不会忽略空格符，它把整行一个一个读取到变量ch中，而且包括换行符“\n”，因此该函数可以用来处理整行输入，而不管输入中有没有空格。同时，有个cin.ignore(a,b)，其中a是int型，b是字符char型，cin.ignore(100,’\n’)表示在一行输入中无视前100个(包括第100个)字符，如果还没到100个而’\n’先到，则到这个字符为止，此后不再忽略任一字符；同理如果在固定字符到来之前100个先到，则到第100 个为止，此后不再忽略任一字符。

12、函数getline(cin,str)也是读取一整行字符到变量str中，str是个string类的字符串，变量被赋予值后，其中并不包含换行符‘\n’。

13、与我们常规编程相比，另一种方法显得很是迷人，它就是“文件输入和文件输出”，意思是程序运行时自动从文件取出需要的输入而不用人为输入，同时输出的时候也不弹出黑屏，只是把输出显示在某一特定文件中，这种方法特别适用于大批量输入和输出。具体方法是：头文件<fstream>,<ifstream> indata,<ofstream> outdata,其中indata想当是cin，outdata相当是cout，从文件中读取数据，先存好数据再给文件命名，如”loan.in”，再有indata.open(“loan.in”)，outdata.open(“loan.out”)，indata.close()，outdata.close()。当然其中的变量indata、outdata也可以其他如：infile、outfile。infile、outfille相当于cin、cout，所以有关cin和cout的函数对他们也适用，如infile.get(somechar)等。打开输入文件时，括号里是C string，把string变成C string用函数str.c\_str()，也就是说str是string，而str.c\_str()是个C string，因此可以借此来运行中打开我们指定的文件。

14、比较string之间的大小。首先，要知道什么是string，什么是C string。比较时操作符两头不能都是C string，比较时一个字母一个字母的比较。如：string str1=”Tremendous”;string str2=”Small”;则有str1>str2。另外，str2<”Smaller”。

15、注意if else和else if的区别，else是紧跟着它前面的那个if的。

16、测试终止输入或者说测试输入格式是否正确。有以下方式：if(cin)，if(!indata)。上面两种是针对正常输入和文件输入的，当然形式上不具有全面性。

17、ch=getchar()和cin.get(ch)一样，gets(char \*str)和getline(cin,string str)一样。

18、C++里空指针、内存泄露、野指针以及数组越界问题

空指针可用if判断

内存泄露是指在堆内存里申请了空间但是使用完了没有释放，这部分内存是不会被回收的，只能通过手动释放。malloc(free)、new(delete)。

野指针是指 指针指向了垃圾内存空间，比如为了避免内存泄露而是用了free或delete，这些堆内存变成垃圾空间，但是原来指向这些空间的指针没有变成Null,这就是野指针，是用if判断不出来野指针，容易被当成正常指针使用而出错，所以delete或free后需要把原来指针置为null。

对于数组越界问题，可以采用多申请额外的两个空间作为标记，当发生数组越界读写时这两个单位的内容会改变，通过判断这两个元素的内容和初始值是否一样能避免数组越界问题。

19、C++中数据栈内存和堆内存问题

普通声明的数据都在栈内存里，使用new或malloc声明的数据在堆内存里。拿对象来说，普通声明的对象存储在栈内存里，使用动态分配对象指针的话，指针保存在栈内存里，指向堆内存里的对象区域，使用对象的析构函数会释放对象的内存空间，不管是在栈里还是堆里的内存空间。