C++的作用域分为：局部作用域（块作用域），函数作用域，函数原型作用域，文件作用域和类作用域。

1．局部作用域：一般局部变量的作用域都为局部作用域，要注意某些语句中变量的作用域，如if语句和switch语句，如果在条件测试的表达式中声明了标识符，则该标识符的作用域在该语句内。注意：在C语言中则没有语句作用域的情况，因为C语言中所有变量的定义都必须在函数开头就完成。例1：

if (int i = 5)

{

i = 2 \* i;

}

else

{

i = 100; // i的作用域到此结束

}

// cout << i << endl; //error i无定义

例2：

int m = 6;

if (int i = 5)

{

i = 2 \* i;

int m = 5; // 表达式内部定义的变量，作用域也随着if语句的结束而结束

}

else

{

cout << m << endl;

i = 100;//i的作用域到此结束

}

//cout<<i<<endl; // error i无定义

cout << m <<endl; // 输出为6

例3：

注意下面这种情况

int n = 5;

if (n > 5)

{

int i = n;

cout << i << endl; // 整型i的作用域到此结束

}

else

{

double i = n + 0.5;

cout <<i << endl; // 双精度的i的作用域到此结束

}

对于switch语句

switch (int i = 1)

{

case 1:

int m = 5;

cout << i << endl;

cout << m << endl;

break;

} // i的作用域到此结束，m的作用域也到此结束

// cout<<i<<endl; // error

对于for语句

int number = 0;

for (int i = 0; i < 5; ++i) // i的作用域从此开始

{

int half\_number = 0;

if (i % 2)

{

number += 1;

}

} // half\_number的作用域到此结束，i的作用域也到此结束

// half\_number=number/2;//error

cout << number << endl;

2．函数作用域：标号是唯一具有函数作用域的标识符。goto语句使用标号，使得该标识符可以在一个函数内的任何位置均可以被使用，而局部变量不具有函数作用域，用extern声明也不行，extern只能扩展全局变量的作用域。

3．函数原型作用域：在函数原型声明（不是函数定义）中所作的参数声明的作用域从函数原型声明的左括号到右括号结束。如：

void f(int n, int m);

m = 5;//error，m无定义

4．文件作用域：又称全局作用域，静态全局变量，全局变量和静态函数都是文件作用域的。例：

#include <iostream>

using namespace std;

int number = 1;

static int value = 2;

int main(int argc, char\* argv[])

{

extern int a;

a = 1; // 不用extern扩展，则error

number = 0;

value = 9;

return 0;

}

int a = 0;