



扫雷英雄榜——C++常见错误及解决方案

1. 在源码中遗失 “;”

» 错误信息

```
syntax error: missing ‘;’
```

» 错误示例

```
int test, number,  
test = 12;
```

» 解决方案

找到出错的相应位置，补上 ‘;’

2. 缺少命名空间定义

» 错误信息

```
error c2065: ‘cout’: undeclared identifier
```

» 错误示例

```
#include <iostream>  
int main()  
{  
    cout << “hello, world”<<endl;  
    return 0;  
}
```

» 解决方案

在主程序头，添加命名空间使用定义。

```
using namespace std;
```

3. 变量未定义直接使用

» 错误信息

```
error c2065: ‘xxxx’ : undeclared identifier
```





» 错误示例

```
int main()
{
    int age;
    cout << "age:\t" << age << endl
    cout << "name:\t" << name << endl;
}
```

» 解决方案

使用一个变量必须提起声明。

4. 程序中使用中文标示符

» 错误信息

```
error c2018: unknown character '0xa3'
```

» 错误示例

```
int age; //中文下的分号
```

» 解决方案

如果将英文 ‘;’， 错输入成 ‘；’，找到标号错误的地方，改成英文的。

5. 错误地使用输入输出符

» 错误信息

```
error c2676: binary '>>':class std::basic_ostream<char, atruct
std::char_traits<char>>'does not define this operator or a conversion to a type acceptable
to the acceptable to the predefined operator
```

» 错误示例

```
cout >> a;
```

» 解决方案

把错误的输入输出符改成相应的，如 `cout >> a`, 改成 `cin >> a`。

6. 变量类型不匹配

» 错误信息



warning c4305: 'initializing':truncation from 'const double' to 'float'

» 错误示例

```
float pi =3.412345245656245
```

» 解决方案

在给变量赋值的时候，看清变量类型。

7. 变量在赋值之前使用

» 错误信息

warning c4700: local variable 'a' used without having been initialized

» 错误示例

```
int i, j, k;  
k = i + j;  
cin >> i >> j;
```

» 解决方案

这种错误主要是对面向过程的程序没理解透彻，变量使用前，先初始化，对其进行赋值。

8. 在一个工程中出现多个 main 函数

» 错误信息

error c2556: 'int _cdecl main(void)': overloaded function differs only by return type from 'void _cdecl main(void)' e:\tmp\tsing.cpp(4): see declaration of 'main'e:\tmp\bigd.cpp(15):error c2371: 'main': redefinition; different basic types

» 错误示例

```
//tsing.cpp  
.....  
int main()  
{  
.....  
}  
// bigd.cpp  
.....  
int main()
```

```
{  
.....  
}
```

» 解决方案

删除另外一个 main 函数，一个工程只能有一个 main 函数。

9. 在函数定义后面使用分号

» 错误信息

```
error C2447: missing function header (old-style formal list?)
```

» 错误示例

```
void test();  
{  
.....  
}
```

» 解决方案

删除多余的分号。

10. 函数定义/使用/声明参数个数不匹配

» 错误信息

```
error c2660: 'chang':function does not take 2 parameters
```

» 错误示例

```
void chang(int a);  
int main()  
{  
.....  
chang(1, 2);  
}  
void chang(int a, int b)  
{  
.....  
}
```

» 解决方案

函数声明和定义尽量放在两个文件里，以便管理参数个数和类型要用对。

11. 未包含指定头文件

» 错误信息

```
fatal error c1010: unexpected end of file while looking for precompiled header directive.
```

» 错误示例

```
//CMyClass 定义在 MyClass.h 中
#include <iostream>
using namespace std;
int main()
{
    .....
    CMyClass myClass;
    .....
}
```

» 解决方案

找到因为哪个函数导致未包含头文件，然后找到该函数的头文件，包含指定头文件即可。

12. 不能打开指定头文件

» 错误信息

```
fatal error c1083: Cannot open include file: 'R.....h' No such file or directory
```

» 错误示例

```
#include "E:\Test.h"
//Test.h 不在 E 目录下，或者名字不对
```

» 解决方案

指定头文件名错误，或者指定路径错误，找到该头文件的正确名字或者路径。

13. 类重定义

» 错误信息

```
error c2011: c·····: class type redefinition
```

» 错误示例

```
//CMyClass.h 首次定义 CMyClass
class CMyClass{
.....
}
//COtherClass.h 重复定义 CMyClass
class CMyClass {
.....
}
```

» 解决方案

出错信息说明，该类已经定义过了，所以删除对该类的定义，或者给该类换个类名。

14. switch 语句的 case 分支中使用变量

» 错误信息

```
error c2057: expected constant expression
```

» 错误示例

```
int a, b;
switch(a){
.....
case(b):
break;
.....
}
```

» 解决方案

把相应的 case 分支，改成常量即可。

15. 函数参数在函数体中重定义

» 错误信息



error c2082:redefinition of formal parameter 'bRet'

» 错误示例

```
int Add(int val1, int val2)
{
    int val1 = val1;
    .....
}
```

» 解决方案

在函数体内部有变量和参数名字一样，修改下函数体的局部变量的名字即可。

16. 句法错误：“{”前缺少“:”

» 错误信息

error c2143: syntax error :missing ':' before '{'

» 错误示例

```
int main()
{
    .....
    return 0 // 忘记 ';'
}
```

» 解决方案

一般是因为编程疏忽，在相应语句后面添加“;”。

17. 语法错误：‘50’该值已被使用

» 错误信息

error C2196: case value '69' already used。

» 错误示例

```
swotcj(val){
    case(10):
    .....
    case(50):
    .....
}
```



```
case(50):  
.....  
}
```

» 解决方案

一般都是疏忽所致，删除多余的 case 分支。

18. 成员函数未声明

» 错误信息

```
error C2509: 'testFunc' : member function not declared in 'CHello'.
```

» 错误示例

```
class CHello{  
public:  
    CHello();  
    ~CHello();  
}  
int test()  
{  
    CHello test;  
    test.testFunc();  
}
```

» 解决方案

在类内部提供该函数的声明，以及实现。

19. 函数重载错误

» 错误信息

```
error C2555: 'B::f1': overriding virtual function differs from 'A::f1' only by return  
type or calling convention
```

» 错误代码

```
class A{  
.....  
public:  
    virtual int f1(){}  
}
```



```
.....  
}  
class B:public A{  
.....  
public:  
bool f1(){}  
.....  
}
```

» 解决方案

函数重载只能通过参数的不同来识别，不能根据返回值来确定，所以重载函数的时候注意这一点。

20. 函数参数传递错误

» 错误信息

```
error C2660: 'test' : function does not take 2 parameters
```

» 错误代码

```
int spt(int num)  
{  
    return num*num;  
}  
int main()  
{  
    int result = sqt(2, 4);  
}
```

» 解决方案

传递给函数的参数和函数定义不符，调用函数的时候根据具体函数参数来传递，这样就不会出错，上面的问题只需传递一个参数就行。

21. 连接错误：不能打开某文件

» 错误信息

```
LINK : fatal error LNK1168: cannot open Debug/P1.exe for writing
```

» 错误代码



无

» 解决方案

这中问题一般是因为要编译的程序正在执行，或者被其他程序调用，关闭正在执行的程序和调用该程序的程序即可。

22. 数组访问越界

» 错误信息

无

» 错误代码

```
int arr[10];
for(int k = 0; k < 10; k++)
    arr[k] = k;
cout << arr[12]<<endl;
```

» 解决方案

越界访问数组，C++编译器并不提示错误，但这是程序致命的弱点。有可能修改关键数据，所有访问数组的时候不要直接用常量标志下界。

23. 头文件格式不正确

» 错误信息

EERROR: wrong header for namespace std

» 错误代码

```
#include <iostream.h>
using namespace std;
```

» 解决方案

删除 `iostream` 后面的 `.h` 即可

24. 无效的操作符

» 错误信息

warning C4553: ' = ' : operator has no effect; did you intend ' =' ?





» 错误代码

```
int main(int argc, char **argv)
{
    if(argc = 1)
        cout << "one paramter:"<<endl;
    if(argc = 2)
        cout << "two paramter:"<<endl;
    return 0;
}
```

» 解决方案

疏忽所致比较运算符误写成赋值运算符，把‘=’改成‘==’。良好的编程习惯就是把比较常量作为左值。

25. 函数无返回值

» 错误信息

```
error C4716: 'test' : must return a value
```

» 错误代码

```
int add(int a, int b)
{
    cout<<"Add two numbers!"<<endl;
}
int main()
{
    int nRet = add(1, 2);
    cout <<"The value is:"<<nRet<<endl;
    return 0;
}
```

» 解决方案

函数原型声明为有返回值，但是在函数实现的时候忘记写返回值了，就出现这个错误，只要按函数声明的返回类型，返回相应的值即可。

26. 类定义出错





» 错误信息

new types may not be defined in a return value,extraneous 'int' ignored, 'main' must return 'int'

» 错误代码

```
#include <iostream>
using namespace std;
class A{
public:
void func(){cout <<"hello"<<endl;}
} // 此处 忘记 ‘;’
int main()
{
    A a;
    a.func();
    system("PAUSE");
    return 0;
}
```

» 解决方案

导致原因就是 在类定义介束的时候忘了 ‘;’, 在类定义介绍的时候补上 ‘;’

27. 错误的使用指针

» 错误信息

in function 'int main()', base operator of '->' has non-pointer type 'Test'

» 错误代码

```
#include <iostream>
using namespace std;
class Test{
public:
    void Func(){cout <<"Func"<<endl;}
};
int main()
{
    Test tA;
    tA->Func();// 此处应该使用 ‘.’ 操作符。 ‘->’ 是指针操作符
}
```



```

    system("PAUSE");
    return 0;
}

```

» 解决方案

对于指针，操作数据成员要用 ‘->’ 操作符；对于像结构体，数组，联合等，操作数据成员要用 ‘.’ 操作符。

28. 类成员访问错误

» 错误信息

```
In function 'int main()', 'int A::age' is private within this context
```

» 错误代码

```

#include <iostream>
using namespace std;
class A{
public:
    A(){age = 0;}
    void SetAge(int val){age = val;}
    int GetAge(){return age;}
private:
    int age;// age 是该类的私有成员，外部无法访问。
};
int main()
{
    A a;
    cout<<"age:"<<a.age<<endl;// 此处的 a.age 改成 a.GetAge() 即可
    a.SetAge(23);
    cout<<"Age:"<<a.GetAge()<<endl;
    system("PAUSE");
    return 0;
}

```

» 解决方案

类的 private 成员不能被类外部访问，只能通过该类提供的访问该变量的内部函数来访问。



29. 函数返回类型错误

» 错误信息

In function 'int main()': 'class Binary' has no member named 'GetBt'

» 错误代码

```
#include <iostream>
#include <bitset>
using namespace std;
class Binary{
private:
    bitset<8> bt;
public:
    Binary(){bt = 0;}
    void SetBt(int val){bt = val;}
    bitset &GetBt(){return bt;}// 此处返回值错误，应该是 bitset<8>
};
int main()
{
    Binary b;
    b.SetBt(2);
    cout << b.GetBt()<<endl;
    system("PAUSE");
    return 0;
}
```

» 解决方案

返回类型定义错误，此例中，GetBt 的返回值 bitset 就定义错误，导致出现这一问题，在使用 STL 的时候，一定要注意定义。在这里把 bitset 改成对应得 bitset<8> 即可。

30. 参数引用类型不正确

» 错误信息

invalid initialization of non-const reference of type 'std::string&' from a temporary of type 'const char *' in passing argument 1 of 'void printStr(std::string &)'

» 错误代码

```
#include <iostream>
```



```
#include <string>
using namespace std;
void printStr(string &str)
{
    cout << str<<endl;
}
int main()
{
    string tmp("hello");
    printStr(tmp);
    printStr("world");//此处使用临时对象，导致错误
    system("PAUSE");
    return 0;
}
```

►► 解决方案

C++中像“world”这样的临时对象都是 const 类型的，而 printStr 的参数是一个 std::string 的引用，所有只要不传递一个临时字符串都行。