1. 以下表达式错误的是（C）
2. bool flag ;

if(flag)

1. int flag;

if(flag != 0)

1. double flag;

if(flag == 0)

1. int \*point;

if(point == null)

1. 以下选项中可能发生隐式类型转换的是（B）
2. int 到 char
3. int 到 float
4. int \* 到 char \*
5. unsigned int 到 int
6. 以下语句中正确的是（B）
7. if(flag = 0)
8. while(flag);
9. if(flag > 0)

a =0;

b=0;

1. a > b;
2. 以下函数函数调用错误的是（D）

int func(int iArg= 0, char cArg=’a’, string sArg=”hello”, int \*ipArg = null);

1. func(1);
2. func(1,”nihao”);
3. func(1,’b’);
4. func(1,”nihao”,”hello”);
5. 以下函数调用会选择哪个重载函数（B）

func(1,2);

1. int func(int a)
2. int func(int a, char b);
3. int func(int a, float b);
4. int func(int a, char \*b);
5. 以下容器中，迭代器不支持“+=”操作符的有哪些（A、B）多选
6. vector
7. deque
8. list
9. map
10. 以下程序中，当输入6个字符后，容器的容量是多少（B）

int main()

{

vector<char> vec;

vec.reserve(1024);

string tmp;

cin >> tmp;

string::iterator it = tmp.begin();

for(; it!=tmp.end(); it++)

{

vec.push\_back(\*it);

}

vec.resize(vec.size() +vec.size()/2);

}

1. 6
2. 9
3. 1024
4. 1536
5. 以下说法中错误的是（A、B）
6. 类必须提供一个构造函数；
7. 默认构造函数没有形参；
8. 一个类已经定义了构造函数，编译器将不在合成默认构造函数；
9. 如果一个类没有提供构造函数，编译器会默认生成，并把每个数据成员初始化成相关默认值
10. 以下哪些操作符适合定义为类成员（A、C、D）
11. =
12. +
13. []
14. ->
15. 下列声明中正确的是（A）

class base(){};

1. class Der : public base

{};

1. class Der:base;

{};

1. class Der:public base;
2. class Der

{

public : base

};

## 问答题：

1、以下程序出现了什么错误？

switch（flag）

{

case 1：

a=0;

case 2:

b =1;

case 3:

c=’a’;

}

答：1、缺少break； 2、缺少default分支；

2、以下程序错在那里，请给出修改意见？

int func()

{

string str;

return str;

}

答：1、返回值不匹配。 2、string不建议作为返回值直接返回。

3、内存建议由函数外开辟。

bool func(string \*str)

{

bool ret = false;

//do something with str;

//do something with ret;

return ret;

}

4、以下程序有错误吗？解释程序的功能？

map<int, char> testMap;

testMap[0] =0;

答：没有错误。

map的下标里面是key，如果key 0存在，修改key 0所对应的值为0；

如果key 0不存在，新建一个元素，key为0，值为0；

5、以下程序错在哪里，如何修改？

vector<int> iVec;

list<int> iLst(10);

copy(iLst.begin(), iLst.end(), iVec.begin());

答：iVec是空的，不能直接用copy操作。

改法1：vector<int> iVec(10);

改法2：copy(iLst.begin(), iLst.end(),back\_inserter(iVec));

6、以下程序有什么问题？

class Test()

{

public:

Test(int val);

private:

int val1;

int val2;

};

Test::Test(int val) : val1(val), val2(val1)

{

}

答：构造函数初始化列表随着编译器的不同，初始化的先后顺序也不同。所以，上面的程序尝试用val1去初始化val2是错误的，可能val1还没有被初始化而val2先进行初始化。

正确做法是：Test::Test(int val) : val1(val), val2(val)

7、以下程序错在哪里，怎么修改？

class Test

{

public:

Test()

{

test = 1;

}

static int m\_tscVal

};

int main()

{

cout << Test ::m\_tscVal <<endl;

}

答：1、静态成员m\_tscVal没有被初始化

2、构造函数不能直接使用test；

class Test

{

public:

Test();

static int m\_tscVal

};

Test::Test()

{

test =1;

}

int Test:: m\_tscVal = 0;

int main()

{

cout << Test ::m\_tscVal <<endl;

}

8、确定如下语句运行时输出什么？

class base

{

public:

void func(){cout << “base” << endl;}

};

class derived : public base

{

public

void func(){cout << “derived” << endl;}

}

base bs;

base \*pbs = &bs;

base &bs2 = bs;

derived dr;

derived \*pdr =&dr;

base &bs3 = dr;

bs.func();

pbs->func();

bs2.func();

dr.func();

pdr->fuc();

bs3->func();

答：

bs.func(); base

pbs->func(); base

bs2.func(); base

dr.func(); derived

pdr->fuc(); derived

bs3->func(); base

9、确定如下语句运行时输出什么？

class base

{

public:

virtual void func(){cout << “base” << endl;}

};

class derived : public base

{

public

void func(){cout << “derived” << endl;}

}

derived dr ;

base \*pbs =&dr;

pbs->func();

答：

pbs->func(); derived

以下程序中val的类型是什么？

typedef int T;

template<typename T>

T func(T &a)

{

T val = a;

return val;

}

double dVal;

func(dVal)

答：double