

1.以下哪项不属于 STL container? ()

A stack

B queue

C multimap

D string

2 类模板的使用实际上是类模板实例化成一个具体的_____。

A 类

B 函数

C 模板类

D 对象

5.代码可以通过编译吗? 如果不能应该如何修改?

```
template<class T> class Foo{  
    T tVar;  
public:  
    Foo(T t) : tVar(t) { }  
};  
template<class T> class FooDerived:public Foo<T>  
{  
};  
int main()  
{  
    FooDerived<int> d(5);  
    return 0;  
}
```

A 代码可以正确通过编译。

B 编译错误, FooDerived 是一个继承模板类的非模板类, 它的类型不能改变。

C 编译错误, tVal 变量是一个不确定的类型。

D 编译错误, 可以在 FooDerived 类中添加一个构造函数解决问题。

6.下面关于迭代器失效的描述哪个是错误的（ ）

A vector 的插入操作不会导致迭代器失效

B map 的插入操作不会导致迭代器失效

C vector 的删除操作只会导致指向被删除元素及后面的迭代器失效

D map 的删除操作只会导致指向被删除元素的迭代器失效

7. 以下程序

```
#include<iostream>

using namespace std;

template <typename T>

T Max(T* a, int n) {

    T max=_____;

    for(int i=1;i<n;i++)

        if(a[i]>max) max=a[i];

    return max;

}

int main() {

    int a[10]={3,7,5,0,2,1,8,4,9,6},*p=a+3;

    cout<<Max(p,7);

    return 0;

}
```

运行结果是 9,请为横线处选择合适的程序（ ）

A.*a

B.a[0]

C.0

D.a[3]

8. 有以下程序

```
#include <iostream>

#include <vector>
```

```

using namespace std;

int main()
{
    vector<int> A(10);

    int count=0,n;

    cout<<"请输入 n 的值: ";

    cin>>n;

    A._____(n);

    for(int i=2;i<=n;i++)

        if(i%3==0&& i%5==0) A[count++]=i;

    for(i=0;i<count;i++)

        cout<<A[i]<<" ";

    cout<<endl;}

```

当键盘输入 20，程序的运行结果是 15，请为横线处选择合适的程序（ ）

A.size B.reserve C.resize D.length

9. 下列代码的输出结果为（ ）

```

#include<iostream>

#include<vector>

using namespace std;

int main(void)
{
    vector<int>array;

    array.push_back(100);

    array.push_back(300);

    array.push_back(300);

    array.push_back(500);

    vector<int>::iterator itor;

```

```

for (itor = array.begin(); itor != array.end(); itor++)
{
    if (*itor == 300)
    {
        itor = array.erase(itor);
    }
}

for (itor = array.begin(); itor != array.end(); itor++)
{
    cout << *itor << " ";
}

return 0;
}

```

A.100 300 300 500

B.100 300 500

C.100 500

D.程序错误

10.以下哪项不属于 STL container? ()

A stack

B queue

C Multimap

D string

11.关于类模板，描述错误的是 ()。

A 一个普通基类不能派生类模板

B 类模板可以从普通类派生，也可以从类模板派生

C 根据建立对象时的实际数据类型，编译器把类模板实例化为模板类

D 函数的类模板参数需生成模板类并通过构造函数实例化

12.以下代码实现了从表中删除重复项的功能，请选择其中空白行应填入的正确代码 ()

```

template<typename T>

```

```
void removeDuplicates(list<T> &aList)
```

```
{
```

```
    T curValue;
```

```
    list<T>::iterator cur, p;
```

```
    cur = aList.begin();
```

```
    while (cur != aList.end())
```

```
    {
```

```
        curValue = *cur;
```

```
        //空白行
```

```
        while (p != aList.end())
```

```
        {
```

```
            if (*p == curValue)
```

```
            {
```

```
                //空白行
```

```
            }
```

```
            else
```

```
            {
```

```
                p++;
```

```
            }
```

```
        }
```

```
    }
```

```
}
```

A p=curr+1;aList.erase(p++);

B p=++curr;aList.erase(p++);

C p=curr+1;aList.erase(p);

D p=++curr;aList.erase(p);

13.下面有关函数模板和类模板的说法正确的有？

A 函数模板的实例化是由编译程序在处理函数调用时自动完成的

B 类模板的实例化必须由程序员在程序中显式地指定

C 函数模板针对仅参数类型不同的函数

D 类模板针对仅数据成员和成员函数类型不同的类

14.有如下类模板定义：（）

```
template<class T> class  BigNumber{  
    long  n;  
public:  
    BigNumber(T i):n(i){}  
    BigNumber  operator+(BigNumber  b)  
    {  
        return  BigNumber(n+b.n);  
    }  
};
```

已知 b1,b2 是 BigNumber 的两个对象,则下列表达式中错误的是?

A 3+3

B b1+3

C b1+b2

D 3+b1

15 关于 vector<>初始化问题下面那个是非法的?

A vector<string> sVec;

B vector<vector<int>> ivvec;

C vector<vector<string>> svvec("hello");

16.代码可以通过编译吗? 如果不能应该如何修改?

```
template<class T> class Foo{  
  
    T tVar;  
  
    public:  
  
    Foo(T t) : tVar(t) { }  
  
};  
  
template<class T> class FooDerived:public Foo<T>  
{  
  
};  
  
int main()  
{  
  
    FooDerived<int> d(5);  
  
    return 0;  
  
}
```

A 代码可以正确通过编译。

B 编译错误, FooDerived 是一个继承模板类的非模板类, 它的类型不能改变。

C 编译错误, tVal 变量是一个不确定的类型。

D 编译错误, 可以在 FooDerived 类中添加一个构造函数解决问题。

17.下面关于 stl 的数据结构的哪些说法正确

A std::map 底层是由 rb-tree 实现

B std::set 底层是由 hashtable 实现

C std::unordered_map 底层是由 rb-tree 实现

D std::multiset 用 rb-tree 的 insert_equal 来实现 insert

18 下列哪段代码是实现在一个 vector 中查找匹配值 value 并删除:

A

```
for (std::vector::iterator iter = numbers.begin(); iter != numbers.end(); ++iter)
{
    if (*iter == value)
    {
        numbers.erase(iter);
    }
}
```

B

```
for (std::vector::iterator iter = numbers.begin(); iter != numbers.end(); ++iter)
{
    if (*iter == value)
    {
        numbers.erase(iter);
        iter--;
    }
}
```

C

```
for (std::vector::iterator iter = numbers.begin(); iter != numbers.end(); )
{
    if (*iter == value)
    {
        iter = numbers.erase(iter);
    }
    else
    {
        ++iter;
    }
}
```


}

}

D 以上代码都不对

19. 以下哪项不属于 STL container? ()

A stack

B queue

C multimap

D string

20. 下面哪几种是 STL 容器类型()

A. vector

B. set

C. multivector

D. multiset

E. array

21. 有如下模板定义:

```
template <class T>
```

```
T fun(T x,T y){
```

```
    return x*x+y*y;
```

```
}
```

在下列对 fun 的调用中, 错误的是 ()

A.fun(1, 2)

B.fun(1.0, 2)

C.fun(2.0, 1.0)

D.fun<float>(1, 2.0)

22、在 C++STL 中常用的容器和类型, 下面哪些支持下标"[]"运算?

A 、 vector

B 、 list

C、 deque

D、 map

E 、 set

F unordered_map

G unordered_set

H、 stack

I、 string

23、下面哪一个是 sort 的 template 的正确写法

A void sort(class A first, class A last, class B pred)

B void template(class A, class B)sort(A first, A last, B pred)

C template<class A><class B> void sort(A first, A last, B pred)

D template<class A, class B> void sort(A first, A last, B pred)

24、假设声明了以下的函数模板：

```
template<class T>
T max (T x, T y)
{ return(x>y)?x: y;
}
```

并定义了 int I; char c;

错误的调用语句是 ()。

- A. max(i,i); B. max(c,c); C. max((int)c,i); D. max(i,c);

25.关于函数模板，描述错误的是 ()。

- A 函数模板必须由程序员实例化为可执行的函数模板
B 函数模板的实例化由编译器实现
C 一个类定义中，只要有一个函数模板，这个类就是类模板
D 类模板的成员函数都是函数模板，类模板实例化后，成员函数也随之实例化

27.下列关于 stl 的说法正确的是 ()

- A map 的迭代器的 key 是 const 类型，无法对其进行修改
B stl 的排序算法一般比较传统的快速排序快是因为其选取中值的算法好
C list 是双链表实现，插入的元素的复杂度为 O(1)
D vector 的大小会增大或者减少，但容量只会增大不会减少

28.下列代码可以通过编译吗？如何修改使其通过编译？

```
template <class T>
struct sum {
    static void foo(T op1 , T op2){
        cout << op1 <<op2;
    }
};

sum::foo(1,3);
```

A 编译通过

B 应该去掉 static 关键字

C 调用应该如下: `sum<int>::foo(1,3)`

D 调用应该如下: `sum::<int>foo(1,3)`

29. 下列的模板声明中, 其中几个是正确的 ()

1) `template`

2) `template<T1,T2>`

3) `template<class T1,T2>`

4) `template<class T1,class T2>`

5) `template<typename T1,T2>`

6) `template<typename T1,typename T2>`

7) `template<class T1,typename T2>`

8) `<typename T1,class T2>`

A 2 B 3 C 4 D 5

30. 假设有函数模板定义如下:

```
template <typename T>
```

```
Max( T a, T b ,T &c)
```

```
{ c= a + b; }
```

下列选项正确的是 ()。

A `int x, y; char z;`

`Max(x, y, z);`

B `double x, y, z;`

`Max(x, y, z);`

C `int x, y; float z;`

`Max(x, y, z);`

D `float x; double y, z;`

`Max(x, y, z);`

31. 以下关于 STL 的描述中，____是错的。

A STL 容器是线程不安全的

B 当容量不够时，vector 内部内存扩展方式是翻倍

C std::sort 是稳定排序

D std::bitset 不是一个 STL 容器

E std::stack 默认是用 deque 实现的

F std::string 中可以存储多个 '\0' 字符

32. 有如下模板定义：template <class T>T fun(T x,T y){return x*x+y*y;}

在下列对 fun 的调用中，错误的是：

A.fun(2,8)

B.fun(2.0,8.2)

C.fun(2.3,8)

D.fun<float>(2,8.3)

33. 以下对于方法覆盖的说法正确的有 ()

A.方法覆盖发生在同一类中

B.方法的覆盖发生在子类型中

C.方法名一定要一样

D.参数类型一定要一样

E.返回类型一定要一样

F.访问权限只能一样

34. 在下列模板说明中，正确的是 ()。

A. template < typename T1, T2 >

B. template < class T1, T2 >

C. template < typename T1, typename T2 >

D. template (typedef T1, typedef T2)

35. 以下关于函数模板叙述正确的是（ ）。

- A. 函数模板也是一个具体类型的函数
- B. 函数模板的类型参数与函数的参数是同一个概念
- C. 通过使用不同的类型参数，函数模板可以生成不同类型的函数
- D. 用函数模板定义的函数没有类型

36. 关于类模板，描述错误的是（ ）。

- A. 一个普通基类不能派生类模板
- B. 类模板可以从普通类派生，也可以从类模板派生
- C. 根据建立对象时的实际数据类型，编译器把类模板实例化为模板类
- D. 函数的类模板参数需生成模板类并通过构造函数实例化

37. 以下程序的输出是:

```
#include <iostream>

using namespace std;

template <typename T>

void print(T t){ cout<<"The value is "<<t<<endl; }

template <>

void print<char *>(char* c) { cout<<"The string is " << c <<endl; }

int main()

{

    char str[] = "TrendMicro[char]";
```

```
    unsigned char ustr[] = "TrendMicro[unsigned char]";  
  
    print(str);  
  
    print(ustr);  
  
    return 0;  
  
}
```

- A. The value is TrendMicro[char]
The string is TrendMicro[unsigned char]
- B. The value is TrendMicro[char]
The value is TrendMicro[unsigned char]
- C. The string is TrendMicro[char]
The string is TrendMicro[unsigned char]
- D. The string is TrendMicro[char]
The value is TrendMicro[unsigned char]

38. 以下哪项不属于 STL container? ()

- A. stack B. queue C. multimap D. string

39. 模板的使用是为了 ?

- A. 提高代码的可重用性 B. 提高代码的运行效率
- C. 加强类的封装性
- D. 实现多态性

40. 关于类模板的说法正确的是（）

A. 类模板的主要作用是生成抽象类

B. 类模板实例化时，编译器根据给出的模板实参生成一个类

C. 在类模板中的数据成员具有同样类型

D. 类模板中的成员函数没有返回值