# C++程序面试题

## 选择题（可多选）

1. **下面哪些名字是非法的？ acd**
2. int double = 3.14159;
3. char \_;
4. bool catch-22;
5. char 1\_or\_2 ='1';
6. float Float = 3.14f;
7. **如果在某机器上short 类型占16 位，那么可以赋给short 类型的最大数是什么 c**

A）32767 B）32768

C）65535 D）65536

1. **下列程序段将会输出什么？ a**

int i = 100, sum = 0;

for (int i = 0; i != 10; ++i)

sum += i;

std::cout << i << " " << sum << std::endl;

A）10 45 B）11 55

C）100 45 D）100 55

1. **下列代码输出什么？ d**

int i, &ri = i;

i = 5; ri =10;

std::cout << i << " " << ri << std::endl;

A）5 5 B）5 10

C）10 5 D）10 10

1. **若有定义：int aa[8];则以下表达式中不能代表数组元素aa[1]的地址的是 a**

A）&aa[0]+1  B）&aa[1]

C）&aa[0]++  D）aa+1

1. **若有定义double r = 99; \*p = &r; \*p = r; 则一下正确的叙述为 a**
2. 以上两处的\*p含义相同，都说明给指针变量p赋值
3. 在“double r = 99; \*p = &r;”中，把r的地址赋值给了p所指的存储单元
4. 语句“\*p = r;”把变量r的赋值给指针变量p
5. 语句“\*p = r;”取变量r的值返回r中
6. **请阅读下面的程序，输出的结果是 c**

#include <iostream>

using namespace std;

void main()

{

float i = 8, \*p=0;

void\* q;

q = (void\*)&i;

p = (float\*)q;

cout << q;

cout << p;

}

1. 编译出错
2. 88
3. 输出的都是地址，而且相同
4. 输出的都是地址，而且不同
5. **下面说法正确的是 d**
6. 当异常产生时，可中断程序从头开始
7. 当异常产生时，必须恢复、继续运行程序
8. 具体程序处理异常的方法可不同
9. 在程序中嵌入错误处理语句就能防止程序异常的出现

template <class X> func(X val) {}

template <> func<double>(double val) {}

class obj

{

public:

callFunc() { func(4.5); }

private:

func(int val) {}

};

1. **阅读下面代码，当调用callFunc（）函数之后实际执行的是**
2. template <> func<double>(double val)
3. template <class X> func(X val)
4. func(int val)
5. **关于运算符重载说法不正确的是 c**
6. 重载不能改变运算符的优先级，但可以通过加括号的方式改变其计算顺序
7. 重载不能改变运算符的结合律和操作数的个数
8. 能够创建新的运算符，并非只有现有的运算符才能被重载
9. 运算符重载不能改变运算符用于内部类型对象时的含义。它只能和用户自定义的类型的对象一起作用，或者用于用户自定义类型的对象和内部类型的对象混合使用时

## 填空题

1. **请列出三种STL常用容器\_\_map\_\_\_\_，\_\_\_\_\_set\_\_，\_\_\_vector\_\_\_\_。**
2. **请列出4种你知道的设计模式\_\_\_工厂模式\_\_\_，\_\_\_单例模式\_\_\_\_\_，\_\_桥接模式\_\_\_\_\_\_，\_\_\_\_\_\_\_\_\_\_。**
3. **请认真阅读下面的代码，程序输出结果为 6**

#include <iostream>

using namespace std;

void func( int\* a, int b[] )

{

b[0] = \*a + 6;

}

void main()

{

int a, b[5];

a=0; b[0]=3;

func(&a, b);

cout << b[0];

}

1. **请认真阅读下面的代码，程序输出结果为 8 zhangsan**

#include <iostream>

#include <string.h>

using namespace std;

template<typename T>

T Max( T a, T b )

{

return ( a >= b ) ? a : b;

}

char\* Max( char\* a, char\* b )

{

return ( strcmp(a, b) >=0 ) ? a:b;

}

void main()

{

cout<< Max(5,8) << " " << Max("zhangsan","lisi");

}

1. **下列程序是将字符串逆序化，请填空：**

#include <iostream>

#include <string.h>

using namespace std;

void main()

{

char ch[] = "Hello world!";

char c;

int i=0, j=strlen(ch);

// 以下代码实现将字符串ch逆序化

int left=0,right=j-1;

While(left<right)

{ int tmp=ch[left];

ch[left]=ch[right];

ch[right]=tmp;

left++;

Roght--;

}

for(i=0;i<j;++i){

cout<<ch[i]<<” ”;

}

Return 0;

cout<< ch;

}

### 程序设计题(可在背面空白处作答)

1. 一列数的规则如下: 1、1、2、3、5、8、13、21、34...... 求第30位数是多少， 用递归算法实现
2. 请找出以下代码中的错误并修正。

|  |
| --- |
| #include <iostream>  using namespace std;  class TestClass  {  char\* m\_data;  int m\_size;  public:  TestClass(void)  {  m\_size = 0;  }  ~TestClass(void)  {  if(m\_data)  delete[] m\_data;  }  void copyData(const char\* data, int size)  {  m\_data = new char[size];  memcpy(m\_data, data, size);  m\_size = size;  }  int getDataSize(void) const  {  return m\_size;  }  };  void showSize(TestClass a)  {  cout << "size = " << a.getDataSize() << endl;  }  int main(void)  {  const char\* const szData = "ABC";    TestClass a;  a.copyData(szData, strlen(szData)+1);  showSize(a);  return 0;  } |

### 问答题(在背面空白处作答，以下题目任选3题作答，多答可加分)

1. 请针对游戏开发比较一下HTTP，UDP和TCP的异同？
2. 进程和线程的区别，请列出至少两种线程同步的机制，并比较他们的异同。
3. 请问你使用过哪些数据库？他们的优缺点主要是什么？