

C++程序设计试题

一、单项选择题

1. 对 C++ 语言和 C 语言的兼容性,描述正确的是 (A)
A. C++ 兼容 C B. C++ 部分兼容 C
C. C++ 不兼容 C D. C 兼容 C++
2. 在 C++ 中使用流进行输入输出,其中用于屏幕输出的对象是 (C)
A. cerr B. cin C. cout D. cfile
cerr:标准错误输出; cin:标准输入
3. 对使用关键字 new 所开辟的动态存储空间,释放时必须使用 (C)
A. free B. create C. delete D. release
4. 如没有使用 private 关键字定义类的数据成员,则默认为 (A)
A. Private B. public C. protected D. friend
5. 使用值传递方式将实参传给形参,下列说法正确的是 (A)
A. 形参是实参的备份 B. 实参是形参的备份
C. 形参和实参是同一对象 D. 形参和实参无联系
6. 在函数调用时,如某一默认参数要指明一个特定值,则有 (A)
A. 其之前所有参数都必须赋值 B. 其之后所有参数都必须赋

值

C. 其前、后所有参数都必须赋值 D. 其前、后所有参数都不必赋值

7. 设存在函数 `int max(int, int)` 返回两参数中较大值，若求 22, 59, 70 三者中最大值，下列表达式不正确的是 (C)

A. `int m = max(22, max(59, 70));` B. `int m = max(max(22, 59), 70);`

C. `int m = max(22, 59, 70);` D. `int m = max(59, max(22, 70));`

8. 下列哪个类型函数不适合声明为内联函数 (A)

A. 函数体语句较多 B. 函数体语句较少

C. 函数执行时间较短 D. 函数执行时间过长

9. `int Func(int, int);` 不可与下列哪个函数构成重载 (B)

A. `int Func(int, int, int);` B. `double Func(int, int);`

C. `double Func(double, double);` D. `double Func(int, double);`

说明：不能以形参名来区分函数；不能以函数返回值来区分函数。

必须是形参的个数或者类型不同才能构成函数重载。

10. 对类中声明的变量，下列描述中正确的是 (C)

A. 属于全局变量 B. 只属于该类

C. 属于该类，某些情况下也可被该类不同实例所共享

D. 任何情况下都可被该类所有实例共享

11. 类的私有成员可在何处访问 (D)

A. 通过子类的对象访问 B. 本类及子类的成员函数中

C. 通过该类对象访问 D. 本类的成员函数中

12. 如果没有为一个类定义任何构造函数的情况下，下列描述正确的是（ A ）

A. 编译器总是自动创建一个不带参数的构造函数

B. 这个类没有构造函数

C. 这个类不需要构造函数

D. 该类不能通过编译

13. 一个类可包含析构函数的个数是（ B ）

A. 0 个 B. 1 个 C. 至少一个 D. 0 个或多个

只有一个，多了编译器都不知道调用哪个好。而构造函数可以有多个，因为构造函数调用时编译器可以根据变量的类型和数量来确定调用哪个构造函数，而析构函数是系统调用的，不能做到这种效果。

14. this 指针存在的目的是（ B ）

A. 保证基类公有成员在子类中可以被访问

B. 保证每个对象拥有自己的数据成员，但共享处理这些数据成员的代码

C. 保证基类保护成员在子类中可以被访问

D. 保证基类私有成员在子类中可以被访问

15. 下列关于类的权限的描述错误的是（ A ）

A. 类本身的成员函数只能访问自身的私有成员

B. 类的对象只能访问该类的公有成员

C. 普通函数不能直接访问类的公有成员，必须通过对象访问

D. 一个类可以将另一个类的对象作为成员

16. 在编译指令中，宏定义使用哪个指令（ B ）

A. #include B. #define C. #if D. #else

17. 设类 A 将其它类对象作为成员，则建立 A 类对象时，下列描述正确的是（ B ）

A. A 类构造函数先执行 B. 成员构造函数先执行

C. 两者并行执行 D. 不能确定

类的组合，描述的是一个类内嵌其他类的对象作为成员的情况，它们之间的关系是一种包含与被包含的关系。

在声明一个组合类的对象时，不仅它自身的构造函数将被调用，而且还将调用其内嵌对象的构造函数。这时构造函数的调用顺序是：(1)按照内嵌对象在组合类的声明中出现的次序，依次调用内嵌对象的构造函数；(2)执行本类构造函数的函数体，析构函数的调用执行顺序与构造函数刚好相反。

18. 下列描述错误的是（ A ）

A. 在创建对象前，静态成员不存在 B. 静态成员是类的成员

C. 静态成员不能是虚函数 D. 静态成员函数不能直接访问非静态成员

说明：静态成员函数可以直接访问该类的静态数据和函数成员，而访问非静态成员，必须通过参数传递方式得到对象名，然后通过对象名来访问。

19. 对于友元描述正确的是 (B)

- A. 友元是本类的成员函数 B. 友元不是本类的成员函数
C. 友元不是函数 D. 友元不能访问本类私有成员

20. 在哪种派生方式中, 派生类可以访问基类中的 `protected` 成员 (B)

- A. `public` 和 `private` B. `public` 和 `protected`
C. `protected` 和 `private` D. 仅 `protected`

二、填空题(本大题共 20 小题, 每小题 1 分, 共 20 分)

请在每小题的空格中填上正确答案。错填、不填均无分。

21. 设要把一个文件输出流对象 `myFile` 与文件 “f: \myText.txt” 相关联, 所用的 C++ 语句是: `myFile.open(“f:\\myText.txt”);`_____。

22. C++ 中 `ostream` 类的直接基类是 `ios` 类_____。

23. 运算符 `[]` 只能用_____成员 _____运算符来重载, 不能用友元运算符来重载。

24. 定义虚函数所用的关键字是 `virtual` _____。

25. `vector` 类中用于返回向量中第 1 个对象的方法是 `front()` _____。

26. 在 C++ 中, 利用向量类模板定义一个具有 20 个 `char` 的向量 `E`, 其元素均被置为字符 ‘t’, 实现此操作

的语句是 `vector<char>E(20, ‘t’)`_____。

27. 类的继承是指子类继承基类的__**数据成员**____和成员函数。

28. 不同对象可以调用相同名称的函数，但可导致完全不同的行为的现象称为__**多态性**或(类的多态性)____。

29. __**#include** ____指令指示编译器将一个源文件嵌入到带该指令的源文件之中。

30. 设类 A 有成员函数

```
void Fun(void);
```

若要定义一个指向类成员函数的指针变量 pafn 来指向 Fun，该指针变量的声明语句是：__**void(A::*pafn)(void);**____。

31. 设在程序中使用如下语句申请了一个对象数组：

```
Point * ptr = new Point[2];
```

则在需要释放 ptr 指向的动态数组对象时，所使用的语句是__**delete[]ptr;**____。

32. 在保护派生中，基类权限为 **Private** 的成员在派生类中__不可访问____。

33. 类 A 有如下成员函数

```
int A::fun(double x){return (int) x / 2; }
```

```
int A::fun(int x){return x*2; }
```

设 a 为类 A 的对象，在主函数中有 `int s=a. fun(6.0)+a.fun(2)`，则执行该语句后，s 的值为__**7**__。

34. 对象传送的消息一般由 3 部分组成：接收对象名、调用操作

名和____必要的参数____。

35. 将指向对象的指针作为函数参数，形参是对象指针，实参是对象的____地址值 ____。

36. 在使用 `string` 类的 `find` 成员函数来检索主串中是否含有指定的子串时，若在主串中不含指定的子串，`find` 函数的返回值是____ -1____。

37. 在 C++ 中，声明布尔类型变量所用的关键字是____ `bool` ____。

38. 执行下列代码

```
int a=29, b=100;
```

```
cout <<setw (3) << a << b << endl;
```

程序的输出结果是：____ 29100__。（2 前面有个空格）

39. 执行下列代码

```
cout <<" Hex: " << hex << 255;
```

程序的输出结果为____ `Hex:ff` ____。

40. C++ 语言中可以实现输出一个换行符并刷新流功能的操控符是____ `endl` ____。

三、改错题(本大题共 5 小题，每小题 2 分，共 10 分)

下面的类定义中有一处错误，请用下横线标出错误所在行并给出修改意见。

41. `#include<iostream. h>`

```
void main( ) {
```

```
int x=5, y=6;
```

```
const int * p=&x;
```

```
*p=y;
```

```
cout<<*p<<endl;
```

```
} *p=y;      指针 p 所指向的为一常量，不能进行左值操作
```

```
42. #include<iostream. h>
```

```
class f{
```

```
private: int x, y;
```

```
public: f1( ){x=0; y=0; }
```

```
print( ){cout<<x<<" "<<y<<endl; }
```

```
};
```

```
void main( ){
```

```
f a;
```

```
a.f1(1,1);
```

```
a.print( );
```

```
} a.f1(1,1); f1( )调用时实参与形参数量不一致
```

```
43. #include<iostream.h>
```

```
class f{
```

```
private: int x=0, y=0;
```

```
public: void f1(int a, int b){x=a; y=b; }
```

```
void get( ){cout<<x<<' '<<y<<endl; }
```

```
};
```



```

void main( ){
    f a;

    a.f1(1,3);

    a.get ( );

    } private:int x=0,y=0; 数据成员声明时不允许直接赋值

```

44. #include<iostream.h>

```

class point{private: float x;

    public: void f(float a){x=a;}

        void f( ){x=0;}

        friend float max(point& a, point& b);

    };

    float max(point& a, point& b)
{return(a.x>b.x)? a.x: b.x; }

void main( ){

    point a, b;

    a.f(2.2); b.f(3.3);

    cout<<a.max(a,b);

    } cout<<a.max(a,b); max()不是类的成员函数，不能用 a.max( )

```

方式调用

45. #include<iostream.h>

```

    template<class T>

class f{

```

```

private: T x, y;

public: void f1(T a, T b){x=a; y=b; }

T max( ){return(x>y)?x:y;}

};

void main( ){

f a;

a.f1(1.5,3.8);

cout<<a.x<<a.y<<a.max( )<<endl;

} cout<<a.x<<a.y<<a.max()<<endl; x,y 是私有类型，不能在主函数
中直接使用

```

四、完成程序题(本大题共 5 小题，每小题 4 分，共 20 分)

46. 完成下面类中的成员函数的定义。

```

class point
{
private:
int m, n;

public:
point(int, int);
point(point&);

};

point::point(int a, int b)
{

```

```

        m=a;
_____ n ____=b;
}

    point::point(_____point&t _____)
    {
        m=t.m;
        n=t.n;
    }

```

47. 下面是一个输入半径，输出其面积和周长的 C++ 程序，在下划线处填上正确的语句。

```

#include<iostream>

using namespace std;

_____ const double _____pi=3.14159;

void main( )
{
    double r;

    cout<<" r=" ;

    _____ cin>>r _____ ;

    double l=2.0*pi*r;

    double s=pi*r*r;

    cout<<" \n The long is: " <<l<<endl;

    cout<<" The area is: " <<s<<endl;

```

```
}
```

48. 在下划线处填上缺少的部分。

```
#include<iostream>
```

```
#include<fstream>
```

```
using namespace std;
```

```
class complex
```

```
{
```

```
public:
```

```
int real;
```

```
int imag;
```

```
complex(int r=0, int i=0)
```

```
{
```

```
    real=r;
```

```
    imag=i;
```

```
}
```

```
};
```

```
complex operator+(____ complex&a _____, complex& b)
```

```
{
```

```
int r=a.real+b.real;
```

```
int i=a.imag+b.imag;
```

```
return_____ complex(r,i) _____;
```

```
}
```

```

void main( )
{
    complex x(1, 2), y(3, 4), z;

    z=x+y;

    cout<<z.real<<" + " <<z.imag<<" i" <<endl;

}

```

49. 程序的输出结果如下：

1, 9

50, 30

请根据输出数据在下面程序中的下划线处填写正确的语句。

源程序如下：

```

#include<iostream>

using namespace std;

class base
{
    private:

    int m;

    public:

    base( ) { };

    base(int a):m(a){}

    int get( ){return m;}

    void set(int a){m=a;}

```

```

};

void main()

{

base*ptr=new base[2];

ptr->set(30);

ptr= ____ ptr+1; ____;

ptr->set(50);

base a[2]={1, 9};

cout<<a[0].get( )<<" , " <<a[1].get( )<<endl;

cout<<ptr->get( )<<" , " ;

ptr=ptr-1;

cout<<____ ptr->get( ) ____<<endl;

delete[ ]ptr;

}

```

50. 在下面横线处填上求两个浮点数之差的 cha 函数的原型声明、调用方法。

```

#include <iostream>

using namespace std;

void main( )

{

float a, b;

____ float cha(float,float); ____ / / 函数 cha 的原型声明

```

```

a=12.5;

b=6.5;

float c=_____ cha(a,b) _____;      // 调用函数 cha

    cout<<c<<endl;

}

float cha(float x, float y)

{

float w;

w=x-y;

return w;

}

```

五、程序分析题(本大题共 4 小题，每小题 5 分，共 20 分)

51. 给出下面程序的输出结果。

```

#include <iostream. h>

template <class T>

class Sample

{
T n;

public:

Sample(T i){n=i;}

int operator==(Sample &);   };

template <class T>

int Sample<T>::operator==(Sample &s)

```

```

    {if(n==s.n)    return 1;

    else    return 0; }

void main( )

{Sample<int>s1(2), s2(3);

    cout<<" s1 与 s2 的数据成员 " <<(s1 ==s2 ?" 相等" : " 不相
等" )<<endl;

    Sample<double>s3(2.5), s4(2.5);

    cout<<" s3 与 s4 的数据成员 " <<(s3==s4 ? " 相等" : " 不相
等" )<<endl;

    }          S1 与 S2 的数据成员不相等

                S3 与 S4 数据成员相等

```

52. 给出下面程序的输出结果。

```

#include<iostream>

using namespace std;

template<class T>

T max(T m1, T m2)

{return(m1>m2)?m1:m2;}

void main( ) {

    cout<<max(1, 7)<<" \t" <<max(2.0, 5.0)<<endl;

    cout<<max(' y' , ' b' <<" \t" <<max(" A" , " a" )<<endl;

}

```


7 5

y a

53. 给出下面程序的输出结果

```
#include <iostream>

using namespace std;

class A {
public:
    int x;

    A() {}

    A(int a){x=a;}

    int get(int a){return x+a;} };

void main( ) {

    A a(8);

    int(A::*p)(int);

    p=A::get;

    cout<<(a.*p)(5)<<endl;

    A*pi=&a;

    cout<<(pi->*P)(7)<<endl; }
```

13

15

54. 给出下面程序的输出结果。

```
include<iostream>

#include<string>

using namespace std;

class Book {

char*title;

char*author;

int numsold;

public:

    Book( ){ }

    Book(const char*str1, const char*str2, const int num) {

        int len=strlen(str1);

        title=new char[len+1];

        strcpy(title, str1);

        len=strlen(str2);

        author=new char[len+1];

        strcpy(author, str2);

        numsold=num;    }

    void setbook(const char*str1, const char*str2, const int num)

{

    int len=strlen(str1);

        title=new char[len+1];
```

```

strcpy(title, str1);

len=strlen(str2);

author=new char[len+1];

    strcpy(author, str2);

numsold=num;

}

~Book( ) {

    delete title;

    delete author;

}

void print(ostream&output) {

    output<<" 书名: " <<title<<endl;

    output<<" 作者: " <<author<<endl;

    output<<" 月销售量: " <<numsold<<endl;

}

};

void main( ) {

Book obj1(" 数据结构" , " 严蔚敏" , 200), obj2;

obj1.print(cout);

obj2.setbook(" C++语言程序设计" , " 李春葆" , 210);

obj2.print(cout);

}

```

月销售量：200

书名：c++程序设计

作者：李春葆

月销售量：210

六、程序设计题(本大题共 1 小题，共 10 分)

55. 在三角形类 TRI 实现两个函数，功能是输入三个顶点坐标判断是否构成三角形

```
#include<iostream.h>

#include<math.h>

class point{
private: float x, y;
public: point(float a, float b){x=a;y=b;}

    point( ){x=0; y=0; }    ,

    void set(float a, float b){x=a;y=b;}

    float getx( ){return x;}

    float gety( ){return y;}

};

class tri{
point x, y, z;

float s1, s2, s3;

public: ... settri(.... ); // 用于输入三个顶点坐标
```

```
.... test(.... ); // 用于判断是否构成三角形
```

}; 请写出两个函数的过程(如果需要形式参数, 请给出形参类型和数量, 以及返回值类型)

判断三条边能构成三角形的条件: 任意两边之和大于第三边或任意两边之差小于第 3 边。

```
void tri::settri(float x1,float y1,float x2,float y2,float x3,float y3)
```

```
{x.set(x1,y1); y.set(x2,y2); z.set(x3,y3); }
```

```
void tri::test(){
```

```
s1=sqrt((x.getx()-y.getx())*(x.getx()-y.getx())+(x.gety()-y.gety())*(x.gety()-y.gety()));
```

```
s2=sqrt((x.getx()-z.getx())*(x.getx()-z.getx())+(x.gety()-z.gety())*(x.gety()-z.gety()));
```

```
s3=sqrt((y.getx()-z.getx())*(y.getx()-z.getx())+(y.gety()-z.gety())*(y.gety()-z.gety()));
```

```
if(((s1+s2>s3)&&(s1+s3>s2)&&(s2+s3>s1))||
```

```
((abs(s1-s2)<s3)&&(abs(s1-s3)<s2)&&(abs(s3-s2)
```

```
<s1))) (
```

```
cout<< “三个顶点能构成三角形” ;  
else cout<< “三个顶点坐标不能构成三角形” ;  
void tri::settri(float x1,float y1,float x2,float y2,float x3,float y3)  
{x.set(x1,y1);   y.set(x2,y2);   z.set(x3,y3); }
```