复习资料  
1.1选择题  
1.在一个C++程序中，main函数的位置（ c ）。  
(a) 必须在程序的开头 (b) 必须在程序的后面  
( c ) 可以在程序的任何地方 (d) 必须在其它函数中间  
2.用C++语言编制的源程序要变为目标程序必须要经过（ d ）。  
(a) 解释 (b) 汇编 (c) 编辑 (d) 编译  
3.C++程序基本单位是（ c ）。  
(a) 数据 (b) 字符 (c) 函数 (d) 语句  
4. C++程序中的语句必须以（ b ）结束。  
(a) 冒号 (b) 分号 (c) 空格 (d)花括号  
5. 执行C++程序时出现的“溢出”错误属于（ c ）错误。  
(a) 编译 (b) 连接 (c) 运行 (d) 逻辑  
6.下列选项中，全部都是C++关键字的选项为（ c ）。  
(a) while IF static (b) break char go   
(c) sizeof case extern (d) switch float integer  
7. 按C++标识符的语法规定，合法的标识符是（ a ,c ）。  
(a) \_abc (b) new (c) int1 (d) “age”  
8．下列选项中，（ a ）不是分隔符。  
(a) ？ (b) ； (c) ： (d) （）  
9．下列正确的八进制整型常量表示是（ b ）。  
(a) 0a0 (b) 015 (c) 080 (d) 0x10  
10．下列正确的十六进制整型常量表示是（ a,b,d ）。  
(a) 0x11 (b) 0xaf (c) 0xg (d) 0x1f  
11．在下列选项中，全部都合法的浮点型数据的选项为（ a,b,d ），全部都不合法的浮点型数据选项是（ c ）。  
(a) -1e3 ， 15. ， 2e-4 (b) 12.34 ， -1e+5 ， 0.0  
(c) 0.2e-2.5 ， e-5 (d) 5.0e-4 ， 0.1 ， 8.e+2  
12．下列正确的字符常量为（ b,d ）。  
(a) " a " (b) '\0' (c) a (d) '\101'  
13．下列选项中，（ a,b,c ）能交换变量a和b的值。  
(a) t=b ；b=a ；a=t； (b) a=a+b ；b=a-b ；a=a-b；  
(c) t=a ；a=b ；b=t； (d) a=b ； b=a ；  
14．执行语句  
int i = 10，\*p = &i；  
后，下面描述错误的是（ a ）。  
(a) p的值为10 (b) p指向整型变量i  
(c) \*p表示变量i的值 (d) p的值是变量i的地址  
15．执行语句  
int a = 5，b = 10，c；int \*p1 = &a, \*p2 = &b；  
后，下面不正确的赋值语句是（ b ）。  
(a) \*p2 = b； (b) p1 = a；  
(c) p2 = p1； (d) c = \*p1 \*（\*p2）；  
16．执行语句  
int a = 10，b；int &pa = a，&pb = b；  
后，下列正确的语句是（ b ）。  
(a) &pb = a； (b) pb = pa；  
(c) &pb = &pa； (d) \*pb = \*pa；  
17．执行下面语句后，a和b的值分别为（ b ）。  
int a = 5，b = 3，t；  
int &ra = a；  
int &rb = b；  
t = ra；ra = rb；rb = t；  
(a) 3和3 (b) 3和5 (c) 5和3 (d) 5和5  
18. 在下列运算符中，（ d ）优先级最高。  
(a) <= （b）\*= （c）+ （d）\*  
19. 在下列运算符中，（ d ）优先级最低。  
(a) ！ （b）&& （c）！= （d）？：  
20．设i=1，j=2，则表达式i+++j的值为（ c ）。  
(a) 1 （b）2 （c）3 （d）4  
21．设i=1，j=2，则表达式 ++i+j的值为（ d ）。  
(a)1 （b）2 （c）3 （d）4  
22．在下列表达式选项中，（ c ）是正确。  
（a）++（a++） （b）a++b （c）a+++b （d）a++++b  
23．已知i=0，j=1，k=2，则逻辑表达式 ++i||--j&&++k的值为（ b ）。  
(a) 0 （b）1 （c）2 （d）3  
24. 执行下列语句后，x的值是( d )，y的值是( c )。  
int x , y ;  
x = y = 1; ++ x || ++ y ;  
(a) 不确定 (b) 0 (c) 1 (d) 2  
25.设X为整型变量，能正确表达数学关系1< X < 5的C++逻辑表达式是( b, c, d )。  
(a) 1<X<5 (b) X==2||X==3||X==4  
(c) 1<X&&X<5 (d) !(X<=1)&&!(X>=5)  
26. 已知x=5，则执行语句  
x += x -= x\*x ；  
后，x的值为（ c ）。  
(a)25 （b）40 （c）-40 （d）20  
27. 设a=1，b=2，c=3，d=4，则条件表达式a<b？a：c<d？c：d的值为( a )。  
(a) 1 （b）2 （c）3 （d）4  
28. 逗号表达式“(x=4\*5，x\*5)，x+25的值为（ d ）。   
(a) 25 （b）20 （c）100 （d）45  
  
1．已知 int i，x，y；在下列选项中错误的是（ c ）。  
(a) if（x == y）i++； (b) if（x = y）i--；  
(c) if（ xy ）i--； (d) if（ x+y ）i++；  
2．设有函数关系为y= ，下面选项中能正确表示上述关系为（ c ）。  
(a) y = 1； (b) y = －1；  
if（ x>=0 ） if（ x！=0）  
if（ x==0 ）y=0； if（ x>0 ）y = 1；  
else y = -1； else y = 0  
(c) if（ x<=0 ） (d) y = -1；  
if（ x<0 ）y = -1； if（ x<=0 ）  
else y = 0； if( x<0 )y = -1;  
else y = 1； else y = 1；  
3．假设i=2，执行下列语句后i的值为（ b ）。  
switch（i）  
{ case 1：i++；  
case 2：i--；  
case 3：++i；break；  
case 4：--i；  
default：i++；  
}  
(a) 1 (b) 2 (c) 3 (d) 4  
4．已知int i=0，x=0；下面while语句执行时循环次数为（ d ）。  
while（!x && i<3 ）{ x++；i++；}  
(a) 4 (b) 3 (c) 2 (d) 1  
5．已知int i=3；下面do\_while 语句执行时循环次数为（ b ）。  
do{ i--； cout<<i<<endl；}while（ i!= 1 ）；  
(a) 1 (b) 2 (c) 3 (d) 无限  
6．下面for语句执行时循环次数为（ b ）。  
for （ int i=0，j=5；i=j；）  
{ cout << i << j << endl；  
i++；j--；  
}  
(a) 0 (b) 5 (c) 10 (d) 无限  
7．以下死循环的程序段是（ b ）。  
(a) for（int x=0；x<3 ；）{ x++；}；  
(b) int k=0；  
do { ++k；} while（ k>=0 ）；  
(c) int a=5；while（a）{ a--；}；  
(d) int i=3；for（；i；i--）；  
  
1．以下正确的函数原型为( d )。  
(a) f( int x; int y ); (b) void f( x, y );  
(c) void f( int x, y ); (d) void f( int, int );  
2．有函数原型 void fun1( int ); 下面选项中，不正确的调用是( c )。  
(a) double x = 2.17 ; fun1( x );  
(b) int a = 15 ; fun1( a\*3.14 ) ;   
(c) int b = 100 ; fun1( & b );  
(d) fun1( 256 );  
3．有函数原型 void fun2( int \* ); 下面选项中，正确的调用是( c )。  
(a) double x = 2.17 ; fun2( &x );  
(b) int a = 15 ; fun2( a\*3.14 );   
(c) int b = 100 ; fun2( &b );  
(d) fun2( 256 );  
4．有函数原型 void fun3( int & ); 下面选项中，正确的调用是( c )。  
(a) int x = 2.17; fun3( &x );  
(b) int a = 15; fun3( a\*3.14 );  
(c) int b = 100; fun3( b );  
(d) fun3( 256 ) ;  
5．有声明 int fun4( int ); int (\*pf)(int) = fun4;   
下面选项中，正确的调用是( c )。  
(a) int a = 15 ;int n = fun4( &a ); (b) int a = 15; cout<<pf(a\*3.14);  
(c) cout<<(\*pf)( 256 ); (d) cout << \*pf( 256 );  
注意：选项(b)也可以调用函数fun4，但由于实参为浮点型表达式，VC6编译器将出现与形参类型不匹配的警告。  
6．在VC中，若定义一个函数的返回类型为void，以下叙述正确的是( a,c )。  
(a) 用语句调用函数 (b) 用表达式调用函数  
(c) 没有返回值 (d) 通过return语句可以返回指定值  
7．函数参数的默认值不允许为( c )。  
(a) 全局常量 (b) 全局变量  
(c) 局部变量 (d) 函数调用  
8．使用重载函数编程序的目的是( a )。  
(a) 使用相同的函数名调用功能相似的函数 (b) 共享程序代码  
(c) 提高程序的运行速度 (d) 节省存贮空间  
9．下列的描述中( b )是错误的。  
(a) 使用全局变量可以从被调用函数中获取多个操作结果  
(b) 局部变量可以初始化，若不初始化，则系统默认它的值为0  
(c) 当函数调用完后，静态局部变量的值不会消失  
(d) 全局变量若不初始化，则系统默认它的值为0  
10．下列选项中，( c ,d )的具有文件作用域。  
(a) 语句标号 (b) 局部变量   
(c) 全局变量 (d) 静态全局变量  
  
1．以下对一维数组 a 的正确定义是( c )。  
(a) int n = 5, a[n]; (b) int a(5);  
(c) const int n = 5; int a[n]; (d) int n; cin>>n; int a[n];   
2．下列数组定义语句中，不合法的是( a )。  
(a) int a[3] = { 0, 1, 2, 3 }; (b) int a[] = { 0, 1, 2 };  
(c) int a[3] = { 0, 1, 2 }; (d) int a[3] = { 0 };  
3．已知 int a[10] = { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 }, \*p = a ;则不能表示数组 a 中元素的式子是( c )。  
(a) \*a (b) \*p (c) a (d) a[ p-a ]  
4．已知 int a[] = { 0, 2, 4, 6, 8, 10 }, \*p = a ; 值不等于0的表达式是( b,d )。  
(a) \*(p++) (b) \*(++p) (c) \*(p-- ) (d) \*(--p)  
5．以下不能对二维数组a进行正确初始化的语句是( c )。  
(a) int a[2][3] = { 0 };  
(b) int a[][3] = { { 0, 1 }, { 0 } };  
(c) int a[2][3] = { { 0, 1 }, { 2, 3 }, { 4, 5 } };  
(d) int a[][3] = { 0, 1, 2, 3, 4, 5 };  
6．已知int a[][3] = { { 0, 1 }, { 2, 3, 4 }, { 5, 6 }, { 7 } } ;则 a[2][1]的值是( c )。  
(a) 0 (b) 2 (c) 6 (d) 7  
7．已知int a[3][3] = { 1, 2, 3, 4, 5, 6, 7, 8, 9 } ; 则不能表示数组元素a[2][1]的地址是( a,b )。  
(a) &[2][1] (b) \*(a[2]+1) (c) a[2]+1 (d) \*(a+2)+1  
8．已知char \*a[]={ "fortran", " basic", "pascal", "java", "c++" ; 则 cout<<a[3];的显示结果是( c )。  
(a) t (b) 一个地址值 (c) java (d) javac++  
9．若用数组名作为调用函数的实参,则传递给形参的是( a )。  
(a) 数组存贮首地址 (b) 数组的第一个元素值  
(c) 数组中全部元素的值 (d) 数组元素的个数  
10．在下列选项中，( b, d )是错误的。  
(a) gets和puts函数可以输入输出包含空格的字符串  
(b) cin不能输入包含空格的字符串  
(c) cout不能输出包含空格的字符串  
(d) 使用赋值运算符可以对字符数组整体赋值  
11．下列描述中，错误的是( c )。  
(a) 输出字符指针就是输出字符串  
(b) 输出字符指针的间接引用就是输出单个字符  
(c) 具有相同字符的两个字符串常量相等  
(d) 两个数组名的比较是地址的比较  
12．判断字符串s1和s2是否相等的表达式为( d )。  
(a) s1=s2 (b) s1==s2  
(c) strcpy(s1,s2)==0 (d) strcmp(s1,s2)==0  
13．判断字符串s1是否大于字符串s2的表达式为( c )。  
(a) s1>s2 (b) strcmp(s1,s2)==0  
(c) strcmp(s1,s2)>0 (d) strcmp(s2,s1)>0  
  
5．若有以下声明和定义，则下列错误的引用是( d )。  
struct worker  
{ int no ;  
char name[ 20 ] ;  
} w, \*p ＝ &w ;  
(a) w.no (b) p->no (c) (\*p).no (d) \*p.no  
6．若有以下声明和定义，则下列引用非法的是( d )。  
struct data  
{ int n;  
float score;  
data \*q ;  
};  
data a[3] = {1001,87,&a[1],1002,75,&a[2],1003,90,&a[0]};  
data \*p = a;  
(a) p->n++ (b) (\*p).n++ (c) ++p->n (d) \*p->n  
7．关于类和对象不正确的说法是( c )。  
(a) 类是一种类型，它封装了数据和操作 (b) 对象是类的实例  
(c) 一个类的对象只有一个 (d) 一个对象必属于某个类  
8．在类定义的外部，可以被访问的成员有( c )。  
(a) 所有类成员 (b) private或protected的类成员  
(c) public的类成员 (d) public或private的类成员  
9．关于this指针的说法错误的是( a,d )。  
(a) this指针必须显式说明  
(b) 当创建一个对象后，this指针就指向该对象  
(c) 成员函数拥有this指针  
(d) 静态成员函数拥有this指针  
10．声明一个类的对象时，系统自动调用( b,d )函数；撤消对象时，系统自动调用( c )函数。  
(a) 成员函数 (b) 构造函数 (c) 析构函数 (d) 复制构造函数  
11．下面对构造函数的不正确描述是( b )。  
(a) 系统可以提供默认的构造函数   
(b) 构造函数可以有参数，所以也可以有返回值  
(c) 构造函数可以重载  
(d) 构造函数可以设置默认参数  
12．下面对析构函数的正确描述是( a,c )。  
(a) 系统可以提供默认的析构函数 (b) 析构函数必须由用户定义  
(c) 析构函数没有参数 (d) 析构函数可以设置默认参数  
13．对静态成员的不正确描述是( c,d )。  
(a) 静态成员不属于对象，是类的共享成员  
(b) 静态数据成员要在类外定义和初始化  
(c) 调用静态成员函数时要通过类或对象激活，所以静态成员函数拥有this指针  
(d) 只有静态成员函数可以操作静态数据成员  
14．下面选项中，不是类的成员函数为( c )。  
(a) 构造函数 (b) 析构函数 (c) 友元函数 (d) 复制构造函数  
15．下面对友员的错误描述是( d )。  
(a) 关键字friend用于声明友员  
(b) 一个类中的成员函数可以是另一个类的友员  
(c) 友员函数访问对象的成员不受访问特性影响  
(d) 友员函数通过this指针访问对象成员  
  
1.在下列运算符中，能重载的是( a,c,d )。  
(a) ! (b) sizeof (c) new (d) delete  
2. 在下列运算符中，不能重载的是( c )。  
(a) <= (b) >> (c) && (d) &=  
3.下列关于运算符重载的描述中，( d )是正确的。  
(a) 可以改变参与运算的操作数个数 (b) 可以改变运算符原来的优先级  
(c) 可以改变运算符原来的结合性 (d) 不能改变原运算符的语义  
4.下列函数中，能重载运算符的函数是( b,c )。  
(a) 成员函数 (b) 构造函数 (c) 析构函数 (d) 友员函数  
5.不能用友员函数重载的是( a )。  
(a) = (b) == (c) += (d) !=  
6.下面描述中，错误的是( b )。  
(a) 只有系统预先定义的运算符才可能被重载  
(b) 使用类型转换函数不能把一个类转换为另一个类  
(c) 使用类型转换函数可以把类转换为基本类型  
(d) 类型转换函数只能定义为一个类的成员函数，不能定义为类的友员函数  
1．在c++中，类与类之间的继承关系具有( c )。  
(a) 自反性 (b) 对称性 (c) 传递性 (d) 反对称性  
2．下列关于类的继承描述中，( a,b )是错误的。  
(a) 派生类可以访问基类的所有数据成员，调用基类的所有成员函数  
(b) 派生类也是基类，所以基类具有派生类的全部属性和方法  
(c) 继承描述类的层次关系，派生类可以具有与基类相同的属性和方法  
(d) 一个基类可以有多个派生类，一个派生类可以有多个基类  
3．当一个派生类公有继承一个基类时，基类中的所有公有成员成为派生类的( a )。  
(a) public 成员 (b)private成员  
(c) protected成员 (d)友员  
4．当一个派生类私有继承一个基类时，基类中的所有公有成员和保护成员成为派生类的( b )。  
(a) public 成员 (b)private成员  
(c) protected成员 (d)友员  
5．当一个派生类保护继承一个基类时，基类中的所有公有成员和保护成员成为派生类的( c )。  
(a) public 成员 (b)private成员  
(c) protected成员 (d)友员  
6．不论派生类以何种方式继承基类，都不能使用基类的( b )。  
(a) public 成员 (b)private成员  
(c) protected成员 (d)public 成员和protected成员  
7．下面描述中，错误的是( b, c )。  
(a) 在基类定义的public成员在公有继承的派生类中可见，也能在类外被访问  
(b) 在基类定义的public和protected成员在私有继承的派生类中可见，在类外可以被访问  
(c) 在基类定义的public和protected成员在保护继承的派生类中不可见  
(d) 在派生类中不可见的成员要变成可访问的需进行访问声明  
8．在c++中，不能被派生类继承的函数是( b,c )。  
(a) 成员函数 (b)构造函数  
(c) 析构函数 (d)静态成员函数  
9．在创建派生类对象时，构造函数的执行顺序是( d )。  
(a) 对象成员构造函数、基类构造函数、派生类本身的构造函数  
(b) 派生类本身的构造函数、基类构造函数、对象成员构造函数  
(c) 基类构造函数、派生类本身的构造函数、对象成员构造函数  
(d) 基类构造函数、对象成员构造函数、派生类本身的构造函数  
10．当不同的类具有相同的间接基类时，有特点( d )。  
(a) 各派生类对象将按继承路线产生自己的基类版本  
(b) 派生类对象无法产生自己的基类版本  
(c) 为了建立惟一的间接基类版本，应该必须改变类格  
(d) 为了建立惟一的间接基类版本，应该声明虚继承  
1．在C++中，要实现动态联编，必须使用( d )调用虚函数。  
(a) 类名 (b) 派生类指针 (c) 对象名 (d) 基类指针  
2．下列函数中，可以作为虚函数的是( c,d )。  
(a) 普通函数 (b) 构造函数 (c) 成员函数 (d) 析构函数  
3．在派生类中，重载一个虚函数时，要求函数名、参数的个数、参数的类型、参数的顺序和函数的返回值( b )。  
(a) 不同 (b) 相同 (c) 相容 (d) 部分相同  
4．下面函数原型声明中，( b )声明了fun为纯虚函数。  
(a) void fun()=0； (b)virtual void fun()=0；  
(c) virtual void fun()； (d)virtual void fun(){ }；  
5．若一个类中含有纯虚函数，则该类称为( d )。  
(a) 基类 (b) 纯基类 (c) 派生类 (d)　抽象类  
6．假设 Aclass为抽象类，下列声明( a,c,d )是错误的。  
(a) Aclass fun( int ) ; (b)Aclass \* p ;  
(c) int fun( Aclass ) ; (d)Aclass Obj ;  
7．下面描述中，正确的是( b,d )。  
(a) 虚函数是没有实现的函数 (b) 纯虚函数的实现在派生类定义  
(c) 抽象类是只有纯虚函数的类 (d) 抽象类指针可以指向不同的派生类  
  
使用虚函数编写程序求球体和圆柱体的体积及表面积。由于球体和圆柱体都可以看作由圆继承而来，所以可以定义圆类circle作为基类。在circle类中定义一个数据成员radius和两个虚函数area()和volume()。由circle类派生sphere类和column类。在派生类中对虚函数area()和volume()重新定义，分别求球体和圆柱体的体积及表面积。  
#include <iostream.h>const double PI=3.14159265;class circle{ public: circle(double r) { radius = r; } virtual double area() { return 0.0; } virtual double volume() { return 0.0; } protected: double radius;};class sphere:public circle{ public: sphere( double r ):circle( r ){ } double area() { return 4.0 \* PI \* radius \* radius; } double volume() { return 4.0 \* PI \* radius \* radius \* radius / 3.0; }};class column:public circle{ public: column( double r,double h ):circle( r ) { height = h; } double area() { return 2.0 \* PI \* radius \* ( height + radius ); } double volume() { return PI \* radius \* radius \* height; } private: double height;};void main(){ circle \*p; sphere sobj(2); p = &sobj; cout << "球体:" << endl; cout << "体积 = " << p->volume() << endl; cout << "表面积 = " << p->area() << endl; column cobj( 3,5 ); p = &cobj; cout << "圆柱体:" << endl; cout << "体积 = " << p->volume() << endl; cout << "表面积 = " << p->area() << endl;}  
  
2、定义一个Book（图书）类，在该类定义中包括  
数据成员： bookname（书名）、price（价格）和number（存书数量）；  
成员函数： display()显示图书的情况；borrow()将存书数量减1，并显示当前存书数量；restore()将存书数量加1，并显示当前存书数量。  
在main函数中，要求创建某一种图书对象，并对该图书进行简单的显示、借阅和归还管理。  
#include <stdlib.h>#include <stdio.h>int main(){class Book{public:long number;float price;char \*bookname;void display(){printf("The name of this book is:%s\n",bookname);printf("The price of this book is:%fdolars\n",price);printf("The number of such book is:%d\n",number);}void restore(){number++;}void borrow(){number--;}};Book b;b.bookname="Harry Potter";b.price=18.00;b.number=100;b.display();b.borrow();b.display();b.restore();b.display();system("pause");return 0;}  
  
  
1. #include <iostream.h>  
void main()  
{ int a,b,c,d,x;  
a = c = 0; b = 1; d = 20;  
if( a ) d = d-10;  
else if( !b )  
if( !c )  
x = 15;  
else x = 25;  
cout << d << endl;  
}  
  
2. #include <iostream.h>  
void main()  
{ int a = 0, b = 1;  
switch( a )  
{ case 0: switch( b )  
{ case 0 : cout << "a=" << a << " b=" << b << endl; break;  
case 1 : cout << "a=" << a << " b=" << b << endl; break;  
}  
case 1: a++; b++; cout << "a=" << a << " b=" << b << endl;  
}  
}  
  
3. #include <iostream.h>  
void main()  
{ int i = 1;  
while( i<=10 )  
if( ++i % 3 != 1 )  
continue;  
else cout << i << endl;  
}  
4、#include < iostream.h >  
class T  
{ public :  
T( int x, int y )  
{ a = x ; b = y ;  
cout << "调用构造函数1." << endl ;  
cout << a << '\t' << b << endl ;  
}  
T( T &d )  
{ cout << "调用构造函数2." << endl ;  
cout << d.a << '\t' << d.b << endl ;  
}  
~T() { cout << "调用析构函数."<<endl; }  
int add( int x, int y = 10 ) { return x + y ; }  
private :  
int a, b ;  
};  
void main()  
{ T d1( 4, 8 ) ;  
T d2( d1 ) ;  
cout << d2.add( 10 ) << endl ;  
}  
答案：  
调用构造函数1.  
4 8  
调用构造函数2.  
4 8  
20  
调用析构函数.  
调用析构函数.  
5． #include < iostream.h >  
struct data  
{ int n ;  
float score ;  
} ;  
void main()  
{ data a[3] = { 1001,87,1002,72,1003,90 } , \*p = a ;  
cout << (p++)->n << endl ;  
cout << (p++)->n << endl ;  
cout << p->n++ << endl ;  
cout << (\*p).n++ << endl ;  
}  
6． #include < iostream.h >  
struct node  
{ char \* s ;  
node \* q ;  
} ;  
void main()  
{ node a[ ] = { { "Mary", a+1 }, { "Jack", a+2 }, { "Jim", a } } ;  
node \*p = a ;  
cout << p->s << endl ;  
cout << p->q->s << endl ;  
cout << p->q->q->s << endl ;  
cout << p->q->q->q->s << endl ;  
}