# 一、选择题(40)

1. C++对C语言作了很多改进，下列描述中（D　）使得C语言发生了质变，从面向过程变成了面向对象。

A、增加了一些新的运算符；

B、允许函数重载，并允许设置缺省参数；

C、规定函数说明必须用原型；

D、引进了类和对象的概念；

1. 下列描述中，（C　）是错误的。

A、内联函数主要解决程序的运行效率问题；

B、内联函数的定义必须出现在内联函数第一次被调用之前；

C、内联函数中可以包括各种语句；

D、对内联函数不可以进行异常接口声明；

1. 在C++中，关于下列设置缺省参数值的描述中，（B　）是正确的。

A、不允许设置缺省参数值；

B、在指定了缺省值的参数右边，不能出现没有指定缺省值的参数；

C、只能在函数的定义性声明中指定参数的缺省值；

D、设置缺省参数值时，必须全部都设置；

1. （D　）不是构造函数的特征。

　A、构造函数的函数名与类名相同；

　B、构造函数可以重载；

　C、构造函数可以设置缺省参数；

　D、构造函数必须指定类型说明。

1. （D　）是析构函数的特征。

　A、析构函数可以有一个或多个参数；

　B、析构函数名与类名不同；

　C、析构函数的定义只能在类体内；

　D、一个类中只能定义一个析构函数；

1. 下列静态数据成员的特性中，（C　）是错误的。

　A、说明静态数据成员时前边要加修饰符static；

　B、静态数据成员要在类体外进行初始化；

　C、静态数据成员不是所有对象所共用的；

　D、引用静态数据成员时，要在其名称前加<类名>和作用域运算符；

1. 友元的作用是（A　）。

　A、提高程序的运用效率；

　B、加强类的封装性；

　C、实现数据的隐藏性；

　D、增加成员函数的种类；

1. 关于new运算符的下列描述中，（D　）是错误的。

　A、它可以用来动态创建对象和对象数组；

　B、使用它创建的对象或对象数组可以使用运算符delete删除；

　C、使用它创建对象时要调用构造函数；

　D、使用它创建对象数组时必须指定初始值；

1. 关于delete运算符的下列描述中，（C　）是错误的。

　A、它必须用于new返回的指针；

　B、使用它删除对象时要调用析构函数；

　C、对一个指针可以使用多次该运算符；

D、指针名前只有一对方括号符号，不管所删除数组的维数。

1. const int \*p说明不能修改（B　）。

　A、p指针；

　B、p指针指向的变量；

　C、p指针指向的数据类型；

D、上述A、B、C三者；

1. 派生类的对象对它的基类成员中（A　）是可以访问的。

　A、公有继承的公有成员；

　B、公有继承的私有成员；

　C、公有继承的保护成员；

D、私有继承的公有成员；

1. 下列描述中，正确的是（C　）

　 A、虚函数是一个static 类型的成员函数；

　B、虚函数是一个非成员函数；

　C、抽象类是指具有纯虚函数的类；

D、抽象类可以说明其对象；

1. 下列描述中，（D　）是不正确的。

A、面向对象程序设计可重用性较好；

B、对象之间通过消息进行信息传递；

C、多态性体现了行为共享；

D、面向对象方法的核心是继承；

1. 执行下列语句后的输出结果是（C　）。

int hot=100;

int &rad=hot;

hot+=100;

cout<<rad<<endl;

A、0；

B、100；

C、200；

D、300；

1. 在C++中，关于下列函数参数的描述中，（B　）是正确的。

A、C++语言中，实参是按照自左至右的顺序求值的；

B、若一个参数设置了默认值，则其右边的所有参数都具有默认值；

C、函数参数的默认值不允许是表达式；

D、设置参数默认值时，应该全部参数都设置；

1. 定义析构函数时，应该注意（C　）。

A、其函数名与类名完全相同；

B、函数返回类型是void类型；

C、无形参，也不可重载；

D、函数体中必须有delete语句；

1. 下列关于构造函数的描述中，（A　）是正确的。

A、构造函数可以设置缺省参数；

B、构造函数可以被继承；

C、构造函数可以对静态数据成员进行初始化；

D、构造函数可以说明为虚函数；

1. 友元的作用是（A　）。

A、提高程序的运用效率；

B、加强类的封装性；

C、实现数据的隐藏性；

D、增加成员函数的种类；

1. 关于堆运算符的下列描述中，（C　）是错误的。

A、使用堆运算符可以动态建立或删除对象；

B、new运算符创建对象时要调用构造函数；

C、可以在使用new[]创建数组时，为每个数组元素指定初始值；

D、delete运算符可以作用于空指针；

1. 关于继承的描述中，（B　）是正确的。

A、派生类是基类的子集；

B、析构函数不能被继承；

C、子类型关系是可逆的；

D、派生类是基类的子类型；

1. 下列关于类型转换的描述中，（A　）是错误的。

A、在不同类型操作数组成的表达式中，其表达式类型一定是最高类型double型；

B、逗号表达式的类型是最后一个表达式的类型；

C、赋值表达式的类型是左值的类型；

D、由低向高的类型转换是保值映射。

1. （D　）不是构造函数的特征。

A、构造函数的函数名与类名相同；

B、构造函数可以重载；

C、构造函数可以设置缺省参数；

D、构造函数必须指定类型说明。

1. 在C++中，关于下列设置参数默认值的描述中，（C　）是正确的。

A、不允许设置参数的默认值；

B、设置参数默认值只能在定义函数时设置；

C、设置参数默认值时，应该是先设置右边的，再设置左边的；

D、设置参数默认值时，应该全部参数都设置。

1. 关于new运算符的下列描述中，（D　）是错误的。

A、它可以用来动态创建对象和对象数组；

B、使用它创建的对象或对象数组可以使用运算符delete删除；

C、使用它创建对象时要调用构造函数；

D、使用它创建对象数组时必须指定初始值。

1. 下列描述中，（D　）是抽象类的特性。

A、可以说明虚函数；

B、可以进行构造函数重载；

C、可以定义友元函数；

D、不能说明其对象。

1. （A　）是析构函数的特征。

A、一个类中只能定义一个析构函数；

B、析构函数名与类名不同；

C、析构函数的定义只能在类体内；

D、析构函数可以有一个或多个参数；

1. 关于子类型的描述中，（A　）是错误的。

A、子类型就是指派生类是基类的子类型；

B、一种类型当它至少提供了另一种类型的行为，则这种类型是另一种类型的子类型；

C、在公有继承下，派生类是基类的子类型；

D、子类型关系是不可逆的。

1. 在下列关键字中，用于说明类中公有成员的是（ A ）。

A．public；

B．private；

C．protected；

D．friend

1. 在下列的各类函数中，（ C ）不是类的成员函数。

A．构造函数；

B．析构函数；

C．友元函数；

D．拷贝初始化构造函数

1. 作用域运算符的功能是（ D ）。

A．标识作用域的级别的；

B. 指出作用域的范围的；

C .给定作用域的大小的；

D. 标识某个成员是属于哪个类的。

1. （ B ）是不可以作为该类的成员的。

A．自身类对象的指针；

B. 自身类的对象；

C. 自身类对象的引用；

D. 另一个类的对象。

1. （ D ）不是构造函数的特征。

A．构造函数的函数名与类名相同；

B. 构造函数可以重载；

C. 构造函数可以设置缺省参数；

D. 构造函数必须指定类型说明。

1. （ A ）是析构函数的特征。

A．一个类中只能定义一个析构函数；

B. 析构函数名与类名不同；

C. 析构函数的定义只能在类体内；

D. 析构函数可以有一个或多个参数。

1. 通常的拷贝初始化构造函数的参数是（ C ）。

A．某个对象名；

B. 某个对象的成员名；

C. 某个对象的引用名；

D. 某个对象的指针名。

1. 关于成员函数特征的下述描述中，（ A ）是错误的。

A．成员函数一定是内联函数；

B. 成员函数可以重载；

C. 成员函数可以设置参数的缺省值；

D. 成员函数可以是静态的。

1. 下述静态数据成员的特性中，（ D ）是错误的。

A．说明静态数据成员时前边要加修饰符static；

B．静态数据成员要在类体外进行初始化；

C．引用静态数据成员时，要在静态数据成员名前加<类名>和作用域运算符；

D．静态数据成员不是所有对象所共用的。

1. 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

1. 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

1. 关于this指针的说法错误的是（A）

A) this指针必须显示说明 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) 静态成员函数不拥有this指针

1. 声明一个类的对象时，系统自动调用（BC）函数，撤消对象时，系统自动调用（）函数

A) 成员函数 B) 构造函数

C) 析构函数 D) 普通函数

1. 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

1. 下面对析构函数的正确描述是（C）

A) 系统不能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

1. 对静态成员的不正确描述是（C）

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) 非静态成员函数也可以操作静态数据成员

8. 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) 缺省参数的构造函数方言

1. 下面对友元的错误描述是（D）

A) 关键字friend用于声明友元

B) 一个类的成员函数可以是另一个类的友元

C) 友元函数访问对象的成员不受访问特性影响

D) 友元函数通过this指针访问对象成员

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

1. 下列关于类的继承描述中，（A）是错误的

A) 派生类可以访问基类的所有数据成员，也能调用基类的所有成员函数

B) 派生类也是基类，但基类不一定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

1. 当一个派生类仅有继承一个基类时，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) 友元

1. 当一个派生类私有继承一个基类时，基类中的所有仅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) 友元

1. 不论派生类以何种方法继承基类，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

1. 下面描述中，错误的是（BC）（两个答案）

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，需要进行访问声明

1. 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

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

1. 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

1. 下面函数原型声明中，（B）声明了fun为虚函数

A) void fun()=0; B) virtual void fun()=0;

C) virtual void fun(); D) virtual void fun(){};

1. 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

1. 假设A为抽象类，下列声明（B）是正确的

A) A fun(int); B) A \*p; C) int fun(A); D) A Obj;

1. 关于函数模板，描述错误的是（A）

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，则这个类是类模板

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

1. 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（AD）（两个答案）

A) float x,y; float z; Max(x,y,z);

B) int x,y,z; Max(x,y,z);

C) int x,y; float z; Max(x,y,z);

D) float x; int y, z; Max(x,y,z);

1. 关于类模板，描述错误的是（D）

A) 一个普通基类不能派生类模板

B) 类模板从普通类派生，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) 函数的类模板参数须通过构造函数实例化

1. 建立类模板对象的实例化过程为（C）

A) 基类à派生类 B) 构造函数à对象

C) 模板类à对象 D) 模板类à模板函数

1. 下列类中（BD）（两个答案）不是输入/输出流类iostream的派生类

A) fstream B) ofstream C) strstream D)ostrstream

1. 在下列选项中（BCD）（三个答案）是ostream类的对象

A) cin B) cerr C) clog D) cout

1. read函数的功能是从输入流中读取（D）

A) 一个字符 B) 当前字符

C) 一行字符 D) 指定若干个字符

1. 下列选项中，用于清除基数格式位置以十六制输出的语句是（B）

A) cout<<setf(ios::dec, ios::basefield);

B) cout<<setf(ios::hex, ios::basefield);

C) cout<<setf(ios::oct, ios::basefield);

D) cin>>setf(ios::hex, ios::basefield);

1. 包含类fstream定义的头文件是（A）

A) fstream.h B) ofstream.h C) ifstream.h D) iostream.h

1. 要求打开文件 d:\file.dat，可写入数据，正确的语句是（D）

A) ifstream infile(“d:\file.dat”, ios::in);

B) ifstream infile(“d:\\file.dat”, ios::in);

C) ofstream infile(“d:\file.dat”, ios::out);

D) fstream infile(“d:\\file.dat”, ios::in| ios::out);

1. 假定已定义浮点型变量data，以二进制方式把data的值写入输出文件流对象outfile中去，正确的语句是（C）

A) outfile.write((float \*) &data, sizeof(float));

B) outfile.write((float \*) &data, data);

C) outfile.write((char \*) &data, sizeof(float));

D) outfile.write((char \*) &data, data);

1. 下列对派生类的描述中，（ D ）是错的。

A．一个派生类可以作另一个派生类的基类；

B．派生类至少有一个基类；

C．派生类的成员除了它自已的成员外，还包含了它的基类的成员；

D．派生类中继承的基类成员的访问权限到派生类保持不变。

1. 派生类的对象对它的基类成员中（ A ）是可以访问的。

A．公有继承的公有成员； B. 公有继承的私有成员；

C. 公有继承的保护成员； D. 私有继承的公有成员。

1. 对基类和派生类的关系描述中，（ B ）是错的。

A．派生类是基类的具体化； B. 派生类是基类的子集；

C. 派生类是基类定义的延续； D. 派生类是基类的组合。

1. 派生类的构造函数的成员初始化列中，不能包含（ C ）。

A．基类的构造函数； B. 派生类中子对象的初始化；

C. 基类的子对象初始化； D. 派生类中一般数据成员的初始化。

1. 关于子类型的描述中，（ A ）是错的。

A．子类型就是指派生类是基类的子类型；

B．一种类型当它至少提供了另一种类型的行为，则这种类型是另一种类型的子类型；

C．在公有继承下，派生类是基类的子类型；

D．子类型关系是不可逆的。

1. 关于多继承二义性的描述中，（ C ）是错的。

A．一个派生类的两个基类中都有某个同名成员，在派生类中对这个成员的访问可能出现二义性；

B．解决二义性的最常用的方法是对成员名的限定法；

C．基类和派生类中同时出现的同名函数，也存在二义性问题；

D．一个派生类是从两个基类派生来的，而这两个基类又有一个共同的基类，对该基类成员进行访问时，也可能出现二义性。

1. 设置虚基类的目的是（ B ）。

A．简化程序； B. 消除二义性；

C. 提高运行效率； D. 减少目标代码。

1. 带有虚基类的多层派生类构造函数的成员初始化列表中都要列出虚基类的构造函数，这样将对虚基类的子对象初始化（ D ）。

A．与虚基类下面的派生类个数有关； B. 多次；

C. 二次； D. 一次。

1. 按照标识符的要求，（ A ）符号不能组成标识符。

A、连接符 B、下划线 C、大小写字母 D、数字字母

1. 若有说明和语句：int a=5，b=6；b\*=a+1；则b的值为（ D ）。

　A、5 B、6 C、31 D、36

1. 表达式!(x>0&&y>0)等价于（ D ）。

A、!(x>0) && !(y>0) B、!x>0 || !y>0

C、!x>0 && !y>0 D、!(x>0) || !(y>0)

1. 若变量已正确定义并且指针p已经指向某个变量x，则(\*p)++相当于（ B ）。

A、p++ B、x++ C、\*(p++) D、&x++

1. 面向对象程序设计将数据与（ A ）放在一起，作为一个相互依存、不可分割的整体来处理。

A、对数据的操作。 B、信息。 C、数据隐藏。 D、数据抽象。

1. 下列关于构造函数与析构函数的描述中，正确的是（ A ）。

A、一个类一定有构造函数和析构函数。

B、构造函数不能有默认参数。

C、析构函数可以重载。

D、可以对一个对象显式地调用构造函数和析构函数。

1. 静态数据成员的初始化必须在（ B ）。

A、类内。 B、类外。 C、在构造函数内。 D、静态成员函数内。

1. 在C++类中定义和实现的函数称为（ C ）。

A、静态成员函数。 B、常成员函数。 C、内联函数。 D、一般函数。

1. 下列关于成员访问权限的描述中，正确的是（ B ）。

A、类的私有成员只可被类的成员函数访问。

B、类的私有成员可被类的成员函数和友元函数访问。

C、类的保护成员可通过类的对象进行访问。

D、类的数据成员只可被类的成员函数访问。

1. 若已定义f，g为double类型，则表达式：f=1，g=f+5/4的值是（ A ）。

A、2.0 B、2.25 C、2.1 D、1.5

1. 表示关系12<=x<=y的C++语言表达式为（ B ）。

A、（12<=x）&（x<=y） B、（12<=x）&&（x<=y）

C、（12<=x）|（x<=y） D、（12<=x）||（x<=y）

1. 设有以下程序段：

int a[5]={0}, \*p, \*q;

p=a;

q=a;

则下列选项中，合法的运算为（ D ）。

A、p+q。 B、p%q。 C、p\*q。 D、p-q。

1. 下列关于类和对象的说法，错误的是（ B ）。

A、对象是类的一个实例。

B、一个类只能有一个对象。

C、任何一个对象只能属于一个具体的类。

D、类与对象的关系和数据类型与变量的关系相似。

1. 下列关于友元函数的描述中，正确的是（ C ）。

A、友元函数的实现必须在类的内部定义。

B、友元函数是类的成员函数。

C、友元函数破坏了类的封装性和隐蔽性。

D、友元函数不能访问类的私有成员。

1. 下列关于静态数据成员的描述中，正确的是（ A ）。

A、静态数据成员是类的所有对象共享的数据。

B、类的每个对象都有自己的静态数据成员。

C、类的不同对象有不同的静态数据成员值。

D、静态数据成员不能通过类的对象调用。

1. 不能作为函数重载的判断依据是（ B ）。

A、const。 B、返回类型。 C、参数个数。 D、参数类型。

1. 类的析构函数的作用是（ D ）。

A、一般成员函数的初始化。 B、类的初始化。

C、对象的初始化。 D、删除类创建的对象。

1. 18．下列关于运算符重载的描述中，错误的是（ C ）。

A、运算符重载不能改变操作数的个数、运算符的优先级、运算符的结合性和运算符的语法结构。

B、不是所有的运算符都可以重载。

C、运算符函数的调用必须使用关键字operator。

D、在C++语言中不可以通过运算符重载创造出新的运算符。

1. 若定义类A：

class A {

public:

A();

A(int x);

…

};

执行语句：A a(4), b[3], \*p[2]; 则自动调用该类构造函数的次数是（ B ）。

A、6 B、4 C、3 D、1

1. 假定AB为一个类，则该类的拷贝构造函数的声明语句为（ D ）。

A、AB&(AB x); B、AB(AB x); C、AB(AB \*x); D、AB(AB &);

# 二、判断题(40)

1. C++中标识符内的大小写字母是没有区别的( )。（×　）
2. C++是一种以编译方式实现的高级语言( )。（√　）
3. 计算函数参数顺序引起的二义性是由不同的编译系统决定的( )。（√　）
4. 返回值类型、参数个数和类型都相同的函数也可以重载( )。（×　）
5. 使用关键字class定义的类中缺省的访问权限是私有（private）的( )。（√　）
6. 类的私有成员只能被类中的成员函数访问，任何类以外的函数对它们的访问都是非法的( )。（×）
7. 多继承情况下，派生类的构造函数的执行顺序取决于成员初始化列表中的顺序( )。（×　）
8. 在公有继承中，基类中的公有成员和私有成员在派生类中都是可见的( )。（×　）
9. 在私有继承中，基类中所有成员对派生类都是不可见的( )。（×　）
10. 在保护继承中，对于垂直访问等同于公有继承，对于水平访问等同于私有继承( )。（√　）
11. 构造函数可以声明为虚函数( )。（×　）
12. 在析构函数中调用虚函数时，采用动态束定( )。（×　）
13. 在一个成员函数内调用一个虚函数时，对该虚函数的调用进行动态束定( )。（√　）
14. 公有继承可以实现子类型( )。（√　）
15. 构造函数可以声明为纯虚函数( )。（×　）
16. 使用关键字class定义的类中缺省的访问权限是私有（private）的( )。√
17. 作用域运算符（：：）只能用来限定成员函数所属的类( )。√
18. 析构函数是一种函数体为空的成员函数( )。×
19. 构造函数和析构函数都不能重载( )。×
20. 说明或定义对象时，类名前面不需要加class关键字( )。√

==============================================================================================================================================================

1. 对象成员的表示与结构变量成员表示相同，使用运算符.或-> ( )。√
2. 所谓私有成员是指只有类中所提供的成员函数才能直接使用它们，任何类以外的函数对它们的访问都是非法的( )。×
3. 某类中的友元类的所有成员函数可以存取或修改该类中的私有成员( )。√
4. 可以在类的构造函数中对静态数据成员进行初始化( )。×
5. 如果一个成员函数只存取一个类的静态数据成员，则可将该成员函数说明为静态成员函数( )。√
6. C++语言中，既允许单继承，又允许多继承( )。√
7. 派生类是从基类派生出来，它不能再生成新的派生类( )。×
8. 派生类的继承方式有两种：公有继承和私有继承( )。×
9. 在公有继承中，基类中的公有成员和私有成员在派生类中都是可见的( )。×
10. 在公有继承中，基类中只有公有成员对派生类对象是可见的( )。√
11. 在私有继承中，基类中只有公有成员对派生类是可见的( )。×
12. 在私有继承中，基类中所有成员对派生类的对象都是不可见的( )。√
13. 在保护继承中，对于垂直访问同于公有继承，而对于水平访问同于私有继承( )。√
14. 派生类是它的基类的组合( )。√
15. 构造函数可以被继承( )。×
16. 析构函数不能被继承( )。√
17. 子类型是不可逆的( )。√
18. 只要是类M继承了类N，就可以说类M是类N的子类型( )。×
19. 如果A类型是B类型的子类型，则A类型必然适应于B类型( )。√
20. 多继承情况下，派生类的构造函数的执行顺序取决于定义派生类时所指定的各基类的顺序( )。√
21. 单继承情况下，派生类中对基类成员的访问也会出现二义性( )。×
22. C++程序一般应当含有main函数，它是C++程序执行的入口( )。√
23. C++语言的标识符是不区分大小写的( )。×
24. 在编译C++程序时，计算机根本不会去识别各个变量名的具体含义，因此命名规范对编写C++程序而言不是必要的，而且有可能会降低编写C++程序的效率( )。×
25. 结构化程序是一种单入口／单出口程序，已经证明了只用三种基本控制结构就可实现这种程序，这三种基本控制结构是：顺序、选择和循环( )。√
26. C++面向对象程序设计建立在结构化程序设计的基础上，以类为构造程序的基本单位( )。√
27. C++语言引入类主要起两个作用：数据封装和类型定义( )。√
28. 假定 AB 为一个类，则执行 AB x; 语句时将自动调用该类的无参构造函数( )。√
29. C++中只能使用注释符号（//），不能使用注释符号（/\*…\*/）( )。×
30. C++中定义常量，不能使用宏定义#define的方法，只能用const ( )。×
31. C++提供的标准库函数都被声明在它所对应的头文件中( )。√
32. C++中数组元素的下标从0开始，它们连续存储在内存单元中( )。√
33. 引用是某个变量的别名，对引用的操作就是对被引用变量的操作( )。√
34. x = 2.5、a = 7、y = 4.7，表达式x + a % 3 \* (int)(x + y) / 24 的值是2.6 ( )。×
35. 常量87.363的数据类型是float ( )。×
36. switch语句中可以没有default子句( )。√
37. 一个函数功能不太复杂，但被频繁调用，可以声明为内联函数( )。√
38. 已知枚举类型：enum Token { NAME,NUMBER,PLUS=5,MINUS,PRINT=10 }; 则枚举常量NAME的值为1 ( )。×
39. 构造函数用来初始化类的新对象，构造函数与类同名，返回类型只能为void ( )。×
40. C++的标准输入对象是cin，标准输出对象有三个，分别是cout、cerr和clog ( )。√
41. 用new动态分配的内存是在堆中分配的，而不是在栈中分配的( )。√

==============================================================================================================================================================

1. 字符数组中每一个元素可存放一个字符,而字符指针变量存放字符串首地址( ).（√）
2. 设int a[][4]={1,2,3,4,5,6,7,8,9};则数组a的第一维的大小是4 ( ) (×)
3. 不能对字符数组整体进行赋值,只能给各个元素赋值,而字符指针变量可以直接赋值.答案( )：√
4. 字符串连接函数是strcat( ) ( )√
5. int a [10]={0, 1, 2, 3, 4, 5};a[6]~ a[9]均为0 ( ).√
6. 若有 int ⅰ=6,j=10; 则执行完语句switch(i){case 4: j++;case 5: j-- ;case 6: j++;case... ( )×
7. C++语言中只能逐个引用数组元素而不能一次引用整个数组( ).√
8. 保护继承方式下基类的保护成员在派生类中仍是保护成员( ).√
9. 若有int i,\*p;则语句\*p=1000是合法的( ).×
10. 对于函数原型:int fun(int a, int b=300,int c=200,int d=100);可以使用如下方式调用:fun(1...：( )×
11. C++语言提供的文件操作包含文本文件和十进制文件.答案( )：×
12. 定义一个对象时,系统只为该对象的数据成员开辟内存空间,而成员函数是同类对象共享的( ).√
13. 从用户的使用角度,函数可以分为以下两种:即系统库函数和用户自定义函数( ).√
14. 同一程序中局部变量和全局变量不可以同名( ).×
15. 在说明一个结构变量后,不能整体引用结构体变量,需要逐一引用( ).√
16. 没用参数的两个函数是不能重载的( ).√
17. 析构函数是一个函数体为空的成员函数( ).×
18. 在for语句中,循环次数是由变量来控制的( ).√
19. 若有 #define S(a,b) a\*b 则语句 area=S(3,2); area的值为6 ( ).√
20. C++语言允许嵌套定义函数,也允许嵌套调用函数( ).×

# 三、填空(20)

1. 对象的核心概念就是通常所说的继承、封装和多态性。
2. 函数的声明、定义和调用称为函数三步曲。
3. Cin后面的符号是：>>，cout后面的符号是：<<。
4. 面向对象的英文缩写是：OOP。
5. 已知X=2,y=3,x<y的值是：true。
6. C语言中printf命令在c++中改用：cout。
7. C语言中scanf命令在c++中改用：cin 。
8. #include<iostream.h>命令中，include的意义是：包含头文件。
9. C++语言中的每条基本语句以 ； 作为结束符，每条复合语句以 {} 作为结束符。
10. 执行cout<<char('A'+2)<<endl；语句后,得到的输出结果为 C 。
11. 假定一个一维数组的定义为char a[8]；则该数组所含元素的个数为 8 ，所占存储空间的字节总数为 8 。
12. 变量分为 全局和局部 两种，变量没有赋初值时，其值是不确定的。
13. 对一个类中的数据成员的初始化，可以通过构造函数来实现，也可以通过成员函数来实现。
14. 假定AB为一个类，则执行AB a（10）；语句时，系统自动调用该类的构造函数的次数为 1 。
15. 假定AB为一个类，则执行AB a[10]；语句时，系统自动调用该类的构造函数的次数为 10 。
16. 假定AB为一个类，有类AB的两个对象a1,a2；则执行语句AB \*a[10]={&a1,&a2}；时系统自动调用该类的构造函数的次数为 0 。
17. 假定类AB中有一个公用属性的静态数据成员static int bb；在类外不通过对象名给该成员bb赋值为10的写法： int AB::bb=10;。
18. 在面向对象程序设计中，具有相同属性和服务的一组对象的集合称为 类 。
19. 在面向对象程序设计中，对象由一组 属性（数据） 和一组 服务（操作） 构成。
20. 类成员的三种访问控制属性是： 公有（public） 、 保护（protected） 、私有（private） 。
21. 在类内部定义的public数据可以在类外进行存取，但protected数据和private数据不能在类外直接存取。
22. 在OOP中，当类的继承方式为公有继承时，基类的 公有成员 和 保护成员 的访问属性在派生类中不变，基类的 私有成员 不可访问。
23. 一个C++源程序按结构一般划分为三种文件，分别是： 类说明文件 、 类实现文件和类的使用文件。
24. 在已经定义了整型指针ip后，为了动态得到一个包括10个整数的数组并由ip所指向，应使用语句 ip=new int[10] 。
25. 26.在一个类中可以定义 多个 构造函数， 1个 析构函数。
26. 在用class定义一个类时，数据成员和成员函数的默认访问权限是 私有 。
27. 在删除一个动态对象时，将自动调用该动态对象所属类的 析构 函数。
28. 为了在对象生存期结束时释放其指针成员所指向的动态存储空间，通常需要为该类定义 析构函数 。
29. 当建立一个新对象时，程序自动调用 构造函数 。
30. 类的私有成员只能被它的成员函数和 友元函数 访问。
31. 假定用户没有给一个名为AB的类定义析构函数，则系统自动给出的缺省析构函数定义为 ~AB(){ } 。
32. 一个抽象类的派生类可以实例化的必要条件是 实现了所有的纯虚函数 。
33. 在C++中，虚函数帮助实现了类的 多态 性。
34. 35.通过C++语言中类的 继承（派生） 机制，可以扩充和完善已有类以适应新的需求。
35. C++类族中，不能被派生类继承的有 构造函数、析构函数 。
36. 定义重载函数时，应保证 参数个数或参数类型 有所不同。
37. 运算符重载仍然保持其原来的 优先级、结合性和语法结构 。
38. 使用指向对象的指针来标识类的成员，则必须使用的运算符是 ->。
39. 拷贝构造函数通常使用 对象的引用 来初始化创建中的对象。
40. 在OOP中多态是指 同样的消息被不同对象接收时可产生不同的行为 ；它通过 重载 和 虚函数 等技术来实现。
41. 含有纯虚函数的类称为 抽象类 。
42. 定义模板的关键字是 template 。
43. 模板定义中标识类型的参数必须用关键字 class(或typename) 修饰。
44. 类模板可以使用两种形式的参数，分别是： 类型参数 和 普通参数 。
45. 函数模板既可以与 函数模板 重载，也可以与 普通函数 重载。
46. 从现存类中构建其子类可以通过C++语言中的 继承（派生）机制 实现。
47. C++的I/O系统通过 流 的机制实现文件和控制台的输入输出操作。
48. cin是 istream 的一个对象，处理标准输入；cout是 ostream 的一个对象，处理标准输出。
49. 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：\_\_ifstream myFile(“f:\\myText.txt”)；\_。

C++中ostream类的直接基类是\_\_\_ios类\_\_\_\_\_。

1. 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。
2. 定义虚函数所用的关键字是\_\_\_virtual\_\_。
3. 类的继承是指子类继承基类的\_\_\_数据成员\_\_\_和成员函数。
4. 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)\_\_。
5. \_\_\_#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。
6. 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]；
7. 当要释放ptr指向的动态数组对象时，所使用的语句是\_\_ delete[ ]ptr；\_。
8. 在保护派生中，基类权限为Private的成员在派生类中\_\_\_不可访问\_\_\_\_\_。
9. 类A有如下成员函数：
   * 1. int A::fun(double x){return (int) x／2；}
     2. int A::fun(int x){return x\*2；}
10. 设a为类A的对象，在主函数中执行语句int s=a．fun(6.0)+a.fun(2)后的s值为\_7\_\_。
11. 类A有如下成员函数：
    * 1. int A::fun(double x){return (int) x／2；}
      2. int A::fun(int x){return x\*2；}
12. 设a为类A的对象，在主函数中执行语句int s=a．fun(6)+a.fun(2.0)后的s值为\_\_13\_\_。
13. 在C++中，声明布尔类型变量所用的关键字是\_\_\_\_bool\_\_\_\_。
14. 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\_\_\_\_。