**一、选择题**

**1**. 下列符号中能够作为C++标识符的是（**D.\_256**）

2. 为了解决程序中函数调用的效率问题，可以将一些函数体代码不是很大，但被频繁调用的函数定义为（B.内联函数）

**3解决程序中代码量不大，但却被频繁调用的函数的调用效率问题的是**（B.内联函数）

4. 抽象类至少包含一个（**C.纯虚函数**）

5. C++中重载的运算符>>是（**B.用于输入操作的非成员函数**）

6. 已知函数f的原型是void f(int &x, int y);，变量v1、v2的定义是：int v1, v2;，下列调用语句中，正确的是（B**.f(v1,&v2);**）

7. 下列有关继承和派生的描述中，错误的是（**C. 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**）

8. 下列语句中，会产生编译错误的是（**B.const int size;**）

9. 不论派生类以何种方式继承基类，都不能使用基类的（**C.私有成员**）

10. 下列关于类和对象的描述中，错误的是（**B.一个类只能有一个对象**）

11. 下列关于类和对象的描述中，错误的是（**B. 类对象可以访问类的全部成员**）

12. **如果运算符'-'作为友元函数重载，在使用显式函数调用时，表达式x-y可以表示为**（A.x.operator-(y)）

**13.如果运算符'-'作为友元函数重载，在使用显式函数调用时，表达式x-y可表示为**（B.operator-(x,y)）

14. 已知函数原型如下：int Fun(int x, int y=2, int z=3)，则下列函数调用错误的是（**D.Fun("1");**）

15. 下列关于友元的描述中，错误的是（**D.友元函数也是成员函数**）

**下列关于友元的描述中，错误的是**（B.友元函数一定是成员函数）

16. 下列关于友元的描述中，错误的是（**B.如果类A是类B的友元，那么类B也是类A的友元**）

**17.下列关于类和纯虚函数的描述中，正确的是**（C.抽象类中包含纯虚函数）

18.在C++中，要实现动态联编，调用虚函数时必须使用（**B.基类指针**）

19.在C++中，能够被派生类继承的函数是（D.成员函数）

20. 在下列函数原型中，可以作为类A构造函数的是（**D.A(int);**）

**21. 下列关于常类型的描述中，正确的是** （**C.必须使用成员初始化列表显式初始化常数据成员**）

22. 释放一个类的对象时，系统自动调用（**A.析构函数**）

**23.为了使用setw设置输出宽度，必须在程序中包含头文件**（A.iomainp）

**24.下列选项中，不属于C++程序开发步骤的是**（B.解释）

**25.下列选项中，能够作为C++标识符的是**（B.welcome）

26. 当一个派生类继承自一个基类时，基类中的所有公有成员都成为派生类的（**D.保护成员**）

27. 下列描述中，作为重载函数调用时选择依据的是 （**C.函数名字**）

28. 下列运算符中，可以重载的是（**D.++**）

29.下列运算符中，可以重载的是（D. !=）

30. 已知show()函数是一个类的常成员函数，无返回值，下列表示中正确的是（**B.void show() const**）

31. 下列符号中不能作为C++标识符的是（**B.5x**）

32. C++流中重载<<的运算符 （**C.用于输出操作的非成员函数**）

33. 已知函数f的原型是void f(int \*x, int &y);，变量v1、v2的定义是：int v1, v2;，下列调用语句中，正确的是（**C.f(&v1,v2);**）

34. 下列关于静态成员的描述中，正确的是（**A.静态成员不属于对象，是类的共享成员**）

35. 下列关于类和对象的描述中，正确的是（**C. 类和对象的关系是一种数据类型与变量的关系**）

36. 下列对常类型的定义中，错误的是（**C.const int &ref=10;**）

37. 下列对常类型的定义中，错误的是（**D.void fun() const;**）

38. 下列描述中，不是面向对象系统包含的要素的是（**D.函数**）

39. 下列关于类定义的描述中，错误的是 （**D.成员函数必须在类体内声明**）

**40.下列关于类定义的描述中，错误的是**（C.可以在类体内队数据成员进行初始化）

**41.下列关于类定义的描述中，正确是**（A.类定义中包含数据成员和成员函数的声明）

42.下列关于默认构造函数中，正确的是（D.默认构造函数是指调用时不需要提供参数的构造函数。）

43.下列关于静态成员的描述中，错误的是（B.j静态数据成员要在类外定义和初始化。）

44.下列关于静态成员的描述中，正确的是（A.静态成员不属于对象，是类的共享成员）

45.下列对析构函数的描述中，正确的是（**C.析构函数没有参数**）

46. 已知函数原型为int test(int,int,int);，则下列重载形式中正确的是（**B.double test(int,int,double);**）

47. 有如下声明：MyClass \* const ptr;，下列描述中，正确的是（**B.ptr是一个常量指针**）

48. C++中，能够被派生类继承的函数是（**D.成员函数**）

49.当派生类从基类私有继承时，基类中的公有成员和保护成员成为派生类的（**C.私有成员**）

50. **C++中，实现封装性需借助于**（**B.类**）

51.**下列选项中，属于C++关键字的是**（B.static）

52.**为了取代C中带参数的宏，在C++中使用了**（C.内联函数）

**53.设变量x和y为整数变量，若有函数调用为”fun(&x,&y)”,则下列选项中，能54.够作为函数fun原型声明的是**（D.void fun(int\*a,int\*b）;)

55. 若有函数调用fun(x+y, 3, min(a,b))，则fun的实参个数为（**A.3**）

56. 下列选项中，不是类成员函数的是（**D.友元函数**）

57. 下列运算符中，不能被重载的是 （**C.::**）

58. 下列关于常类型的描述中，错误的是（**B.通过常对象可以调用一般成员函数**）

59. 下列选项中，istream类对象是（**A.cin**）

60.下列选项中，与实现运行时多态性无关的是（C.重载函数）

61. 下列格式控制符中，可以设置输出项域宽的是（**A.setw**）

62. 已知函数f的原型是void f(int x, int \*y);，变量v1、v2的定义是：int v1, v2;，63.下列调用语句中，正确的是（**A.f(v1,&v2);** ）

64.下列关于继承和派生的描述中，错误的是（**C.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**）

65.在C++的面向对象程序设计框架中，程序的基本组成单元是（A.类）

66.下列关于引用的描述中，错误的是（A.创建引用时必须为其分配存储空间）

67.下列函数参数默认值设置错误的是（B. Fun(int x=0,int y)）

68.下列关于new运算符的描述中，错误的是（C.new运算符创建类对象时要调用类构造函数）

69.若有声明“MyClass\*const ptr;”,则下列描述中，正确的是（B.ptr是一个常量指针）

70.下列关于多态性的描述中，正确的是（A.多态性是指相同的接口可以为不同的对象提供服务）

71. 下列选项中，能体现面向对象思想主要特征的是（**D. 封装**）

72. 下列关于函数原型声明的描述中，正确的是（**D. 在函数原型中不必声明形参名称**）

73. 下列选项中，不是C++关键字的是（**B.final**）

74.下列关于函数原型的叙述中，错误的是（**可以在函数原型中为形参指定默认值**）

75.下已知函数原型为“int test(int,int,int);”，则下来重载形式中正确的是（ **B.double test(int,int,double);** ）

76.下列关于类成员函数特性的描述中，错误的是（**C.成员函数都是内联函数** ）

77.声明一个类的对象时，系统自动调用（**构造函数**）

78.下列关于new运算符的的描述中，错误的是（**new运算符创建数组时要为数组元素指定初值**）

79.在一个派生类对象结束其生命周期时，下列描述中正确的是（**先调用派生类的析构函数，后调用基类的析构函数**）

80.下列函数中，对文件进行写操作的是（**put**）

81.下列选项中，不能作为重载函数调用时选择依据的是（**函数名称**）

82.若MyClass为一个类，执行语句“MyClass a[4],\*p[5];”时自动调用该类构造函数的次数是（**4**）

83. 执行下列语句序列后的输出结果为（**C. 110**）

int count=10, &ref=count;

count+=100; cout<<ref;

84.有如下语句序列：

int x=100,&r=x;

cout<<&x<<’\_’<<r<<endl;

已知其中变量x的地址为0012FF7C，则执行该语句列的输出结果为（C.0012FF7C-100）

85.有如下语句序列：

int x=100,&r=x;

cout<<x<<’\_’<<r<<endl;

已知其中变量x的地址为0012FF7C，则执行该语句列的输出结果为（D.100-100）

86. 下列函数中，具有隐含this指针的是（**A.①**）

class MyClass{

public:

MyClass(); //①

friend void fun2(); //②

static int fun3(); //③

private:

int count; //④

};

87. 下列语句中，会产生编译错误的是（**B.②**）

int i=0, j=1;

int &r=i; //①

&r=j; //②

r=r+5; //③

int \*p=&i; //④

88. 下列函数中，具有隐含的this指针的是（**A.** **①** ）

class MyClass{

public:

int fun1(); //①

friend int fun3(); //②

static void fun2(); //③

static int count; //④

};

89. 有如下类定义，obj是类D的对象，下列语句中不违反访问控制权限的是（**A.obj.fun1();**）

class B{

public: void fun1();

private: void fun2();

protected: void fun3();

};

class D : public B {

protected: void fun4();

};

90. 下列程序段中，横线处应填入的内容是（**D.int MyClass::**）

class MyClass{

public:

MyClass() { count++; }

private:

static int count;

};

　　　　　　count=10;

91.如果下类定义，obj是类D的对象，下列选项中的语句，不违反访问控制权

限的是A．obj.fun1();

class B{

public:void fun1();

private: void fun2();

protected: void fun3();

};

class D:public B{

protected: void fun4();

}

D obj;

二、**填空题**

1. 类成员的访问权限分为(公有继承)、（私有继承）和（保护继承）三类。

2. 为了解决多继承产生的(二义性)问题，在C++中引入了虚基类。

3. 派生新类的类称为(父类)，而派生出的新类称(子类)。

4. **运算符函数的函数名是由运算符前加关键字(operator)**构成的。

5. 所有在类内部定义的成员函数都是（内联）函数。

6. 已知一个函数的原型是double fun(double x);，若要以4.25为实参调用该函数，应使用表达式（fun (4.25) ）。

7. 将一个函数声明为一个类的友元函数必须使用关键字（friend ）。

8.为了取代C中带参数的宏，在C++中使用了（内联）函数。

9.设置虚基类的目的是为了解决多继承产生的（二义）性问题。

**10.若有函数声明“void fun(int& x);”，则函数fun是按**（引用)方式传递参数的。

**11.(this指针)是一个隐含于类的每一个非静态成员函数中的特殊指针。**

12.如果积累Alpha继承自类Beta，则类Beta称为(基)类。

**13.在C++中，（函数重载）是指同一个函数名可以对应多个函数的实现。**

**14.当一个派生类继承自一个基类的每一个非静态成员函数中的特殊指针。**

15.. **有两个类M和C，其中类C定义如下：class C { public: M m;} ;。若建立类C的对象object，则对象m和对象object中后被初始化的是对象**（**object**），先被初始化的是对象（**m）**。

**16.运算符函数的函数名是由运算符前加关键字（operator）构成的。**

17.形状类Shape中包含（一个纯虚函数）Print，它没有形参和返回值，则Print函数在Shape类中的声明格式为（virtual void Print（）=0；）

18.下列语句序列执行后输出10，请将划线处的语句补充完整。

class MyClass {

public:

MyClass(int x):val(x) { }

void Print();

private :

int val ;

};

void　**MyClass::**　　Print() { cout<<val<<endl; }

int main() {

MyClass obj(10);

obj.Print();

return 0;

}

19. 下列语句序列执行后输出Hello，请将划线处的语句补充完整。

class MyClass {

public:

void Print() const { cout<<"Hello"; }

};

int main() {

MyClass\* p = new MyClass();

**p->**　　　Print();

return 0;

}

20. 请在划线处填写正确内容，使类MyClass的复制构造函数的声明完整。

class MyClass {

public:

MyClass(const \_\_\_\_\_**MyClass &**\_\_\_\_\_ obj);

};

21. 下列语句序列的输出结果是\_\_**100**\_\_\_\_\_。

class MyClass {

public:

MyClass(int x) { cout<<x; }

~MyClass() { cout<<0; }

};

int main() {

MyClass obj1(1),obj2(obj1),&ptr=obj1;

return 0;

}

22.**下列语句序列的输出结果是**\_\_\_\_ **120000**\_\_\_\_\_\_\_。

class MyClass {

public:

MyClass(int x=0) { cout<<x; }

~MyClass() { cout<<0; }

};

int main() {

MyClass arr[3]={MyClass(1),MyClass(2)};

return 0;

}

23. 下列语句序列执行后输出10，请将划线处的语句补充完整。

class MyClass {

public:

MyClass(int x) : \_**val(x)**\_\_ \_{ }

void Print( ) { cout<<val<<endl; }

private:

int val;

};

int main() {

MyClass obj(10);

**obj.**　　Print();

return 0;

}

24. 请在划线处填写正确内容，使类Test的复制构造函数的声明完整。

class Test{

public:

Test(const \_\_\_**Test&**\_\_\_\_obj);

};

25. **请在划线处写出MyClass类的析构函数声明**。

class MyClass {

public:

\_\_\_\_\_**~MyClass();\_**\_\_\_\_\_\_

};

26. 请在划线处填写构造函数的声明，使得语句Point p1;创建的p1对象为原点。

class Point {

public:

\_\_\_ **Point() : x(0), y(0) { };\_**\_\_\_\_\_\_\_\_

private:

int x,y;

};

**27.请将划线处缺损部分补充完整，使得类Fruit成为一个抽象类。**

class Fruit{

public;

virtual double Price()\_\_\_=0\_\_\_;

};

28.下列程序运行后的输出结果是“2020”，则划线处的内容应为 **int b** 。

#include <iostream.h>

void fun(int &a, ){int c;c=a;a=b;b=c;}

int main(){

int x=14,y=20;

fun(x,y);

cout<<x<<y;

return 0;

}

29.下列语句序列重载了复数的加法操作，请将划线处的内容补充完整。

class Complex {

public:

Complex(double r=0.0,double i=0.0):real(r),imag(i){}

Friend Complex operator +(Complex &,Complex &);

private:

double real,imag;

};

**Complex** operator +(Complex &c1,Complex &c2){

return Complex(c1.real+c2.real,c1.imag+c2.imag);

}

30.**下列程序输出结果“0，11，12”，请将划线处的语句补充完整**。

#include<iostream.h>

class A{

public:

A(int aa=0):a(aa){}

void Print() const{cout<<a<<’,’;}

privat:

int a;

};

class B:public A{

public:

B(int aa): b(aa+1),\_\_c(aa+2)\_\_{}

void Print() const{A::Print();cout<<b<<’,’;c.Print();}

private:

int b;

A c;

};

31.下列程序执行后输出结果是\_\_\_10\_\_

#include <iostream.h>

int f1(int x,int y=5){return x+y;}

int f1(double x) {return x;}

int main(){

int a=3;

double b=2.5;

cout<<f1(a)+f1(b);

return 0;

}

32.下列程序运行后输出结果是\_\_123000\_\_。

# include <iostream.h>

class Myclass{

public:

Myclass(int x){cout<<x;}

~Myclass(){cout<<0;}

};

int main(){

Myclass arr[]={Myclass(1), Myclass(2), Myclass(3),};

Return 0;

}

33.请按照下列注释语句的提示补充完整类B的构造函数的定义。

class A{

public:

A(int aa=0):a(aa){}

private:

int a;

};

class B: public A{

public:

//用aa初始化基类A，用aa+1初始化成员对象c

B(int aa=0):\_\_A(aa)\_\_,\_\_c(aa+1)\_\_,b(aa+2){}

private:

int b;

A c;

};