

1.下面哪些属于使用"常引用"的原因?

- A 提高程序的效率
- B 节省存储空间
- C 保护传递给函数的数据不在函数中被改变
- D 以上都不正确

2.找出其中的错处

```
class _student {  
public:  
    int a;  
    char b;  
protected:  
    int c;  
}
```

_student st;

st.a='a';

st.b="a";

st.c=1;

A st.a='a'

B st.b="a"

C st.c=1

D 没有错误

4.如下程序的输出结果是

```
#include<iostream>
```

```
using namespace std;
```

```
long fun(int n)
```

```
{
```

```
    if (n > 2)
```

```
        return (fun(n - 1) + fun(n - 2));
```

```
    else
```

```

        return (n);
    }
int main()
{
    int i;
    cout << "Needs an integer: ";
    cin >> i;
    cout << endl;
    cout << fun(i) << endl;
}

```

当分别输入 4 2 时，输出结果分别是？

- A 5, 1 B 4, 2 C 5, 2 D 6, 2

5.如果 A,B,C 为布尔型变量，“^”和“v”分别代表布尔类型的“与”和“或”，下面那一项是正确的（）

I. $A \wedge (B \vee C) = (A \wedge B) \vee (A \wedge C)$

II. $A \vee (B \wedge C) = (A \vee B) \wedge (A \vee C)$

III. $(B \wedge A) \vee C = C \vee (A \wedge B)$

- A 只有 I B 只有 II C 只有 I 和 II D I,II,III;

6.构造异质链表的意义是（）。

- A 用数组组织类对象 B 用链表组织类对象
C 用抽象类指针指向派生类对象 D 用抽象类指针构造派生类对象链表

7.请将下列构造函数补充完整，使得程序的运行结果是 5

```

#include<iostream>

using namespace std;

class Sample{

```

```

public:
    Sample(int x){
        _____
    }
    ~Sample(){
        if(p) delete p;
    }
    int show(){
        return *p;
    }
private:
    int*p;
};

int main(){
    Sample S(5);
    cout<<S.show()<<endl;
    return 0;
}

```

A *p=x; B p=new int(x); C *p=new int(x); D p=&x;

8. 下列选项中，正确的 C++ 标识符是

A 6_group B group~6 C age+3 D _group_6

9. 若 i=3, 则 cout<<-i++; 输出的值为 -4. 说法是否正确?

A 正确 B 错误

10. 在 C++ 语言中，若类 C 中定义了一个方法 int f(int a, int b)，那么方法()不能与该方法同时存

在于类 C 中。

A int f(int x, int y)

B int f(float a, int b)

C float f(int x, float y)

D int f(int x, float y)

11. ClassA *pclassa=new ClassA[5];

delete pclassa;

c++语言中，类 ClassA 的构造函数和析构函数的执行次数分别为()

A 5,1

B 1,1

C 5,5

D 1,5

12. 以下叙述中正确的是 ()

A 局部变量不能被说明为 static

B 只有全局变量才有自己的作用域，函数中的局部变量没有作用域

C 只有在函数内部定义的变量才是局部变量

D 只要是用户定义的标识符，都有一个有效的作用域

13. 关于 C++ 中的 new 和 C 语言中的 malloc，以下说法哪几个是正确的： ()

A 都是用来申请内存的

B 都是函数

C 都会调用构造函数

D 他们没有区别，都一样

14 关于代码输出正确的结果是() (Linux g++ 环境下编译运行)

```
int main(int argc, char *argv[])
```

```
{
```

```
    string a="hello world";
```

```
    string b=a;
```

```
    if (a.c_str()==b.c_str())
```

```
    {
```

```
        cout<<"true"<<endl;
```

```

    }

    else cout<<"false"<<endl;

    string c=b;

    c="";

    if (a.c_str()==b.c_str())

    {

        cout<<"true"<<endl;

    }

    else cout<<"false"<<endl;

    a="";

    if (a.c_str()==b.c_str())

    {

        cout<<"true"<<endl;

    }

    else cout<<"false"<<endl;

    return 0;

}

```

A false false false

B true false false

C true true true

D true true false

15.有关类的说法不正确的是()

A 类是一种用户自定义的数据类型

B 类中的成员函数可以存取类中的私有数据

C 在类中，如果不作特别说明，所有的数据均为私有类型

D 在类中，如果不作特别说明，所有的成员函数均为公有类型

16.关于 c++的 inline 关键字,以下说法正确的是()

- A 使用 `inline` 关键字的函数会被编译器在调用处展开
- B 头文件中可以包含 `inline` 函数的声明
- C 可以在同一个项目的不同源文件内定义函数名相同但实现不同的 `inline` 函数
- D 定义在 Class 声明内的成员函数默认是 `inline` 函数
- E 优先使用 Class 声明内定义的 `inline` 函数
- F 优先使用 Class 实现的 `inline` 函数的实现

17.在 32 位系统中，关于下面程序说法正确的是（ ） [不定项]

```
class Item
{
private:
char *desc;
int storage;
public:
Item()
{
desc = new char[51];
}

void setInfo(char *dsrc, int num)
{
strcpy(desc,dsrc);
storage = num;
}
};

int main()
{
```

```
Item stock;  
  
stock.setInfo("pen",20);  
  
return 0;  
  
}
```

- A main 函数定义了一个 stock 对象占 8 个字节
- B storage 和 desc 成员各占 4 个字节
- C 存在内存泄漏的问题
- D 存在编译错误

18.C++在函数体内定义的变量称全局变量，可以被程序中的所有函数引用。

- A 正确
- B 错误

19.写出程序的运行结果

```
#include<iostream>  
  
using namespace std;  
  
class Base  
{  
    int x;  
  
public:  
    Base(int b): x(b) {}  
    virtual void display()  
    {  
        cout << x;  
    };  
};  
  
class Derived: public Base  
{
```

```

        int y;
public:
    Derived(int d): Base(d), y(d) { } void display()
    {
        cout << y;
    }
};
int main()
{
    Base b(1);
    Derived d(2);
    Base *p = & d;
    b.display();
    d.display();
    p->display();
    return 0;
}

```

A 121

B 222

C 122

D 运行出错

20.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。

A 是

B 否

21.在 32 位环境下，以下代码输出的是（ ）

```
#include<stdio.h>
```

```
class A
```

```
{
```



```
public:
A() { printf("A"); }
~A() { printf("~A"); }

};

class B:public A
{
public;
B() { printf("B"); }
~B() { printf("~B"); }

};

int main()
{
A*c = new B[2];

delete[] c;

return 0;

}

A  ABAB`~A~A
C  ABAB`~B~A
B  ABAB`~B~A~B~A
D  ABAB`~A~B~A~B
```

22. 当一个类对象的生命周期结束后，关于调用析构函数的描述正确的是：（）

- A 如果派生类没有定义析构函数，则只调用基类的析构函数
- B 如果基类没有定义析构函数，则只调用派生类的析构函数
- C 先调用派生类的析构函数，后调用基类的析构函数
- D 先调用基类的析构函数，后调用派生类的析构函数

23. 下列程序的运行结果是 1，请将横线处缺失程序补充完整（）

```
class B;
```

```

class A
{
    friend class B;
public:
    void Display()
    {cout<<x<<endl;}
private:
    int x;
};

class B
{
    public:
    void Set(int i){    =i;}
    void Display(){a.Display();}
private:
    A a;
};

int main()
{
    B b;
    b.Set(1);
    b.Display();
    return 0;
}

```

A A::x

B A.x

C x

D x

24.关于以下代码，哪个说法是正确的？

```

myClass::foo(){
    delete this;
}

```

```
}
```

```
..
```

```
void func(){
```

```
    myClass *a = new myClass();
```

```
    a->foo();
```

```
}
```

A 它会引起栈溢出

B 都不正确

C 它不能编译

D 它会引起段错误

25.面向对象的基本特征有哪些 ()

A 封装

B 继承

C 多态

D 重载

26.当一个类 A 中没有声明任何成员变量与成员函数,这时 sizeof(A)的值是多少?

A 1

B 0

C 4

D 运行时错误

27

```
#include<iostream>
```

```
using namespace std;
```

```
class MyClass
```

```
{
```

```
public:
```

```
    MyClass(int i = 0)
```

```
    {
```

```
        cout << i;
```

```
    }
```

```
    MyClass(const MyClass &x)
```

```
    {
```

```

        cout << 2;
    }
    MyClass &operator=(const MyClass &x)
    {
        cout << 3;
        return *this;
    }
    ~MyClass()
    {
        cout << 4;
    }
};

int main()
{
    MyClass obj1(1), obj2(2);
    MyClass obj3 = obj1;
    return 0;
}

```

运行时的输出结果是（）

A 11214444

B 11314444

C 122444

D 123444

28.下列哪项不属于 C++的基本数据类型？

A wide

B char

C bool

29.某一基类有构造函数 A(), 析构函数 ~A(), 其派生类 B 的构造函数析构函数分别为 B(), ~B(), 程序使用 B 的对象 b 时, 下列说法正确的是 （）

A 开始时, 先调用 A(), 再调用 B(), 结束时, 先调用 ~A(), 再调用 ~B()

- B 开始时，先调用 A()，再调用 B()，结束时，先调用 ~B()，再调用 ~A()
- C 开始时，先调用 B()，再调用 A()，结束时，先调用 ~A()，再调用 ~B()
- D 开始时，先调用 B()，再调用 A()，结束时，先调用 ~B()，再调用 ~A()

30.下面哪种 C/C++ 分配内存的方法会将分配的空间初始化为 0

- A malloc()
- B calloc()
- C realloc()
- D new[]

31.在 C++里，堆中存储的数据是（）

- A 局部变量
- B 函数返回值
- C 函数参数
- D new 和 malloc 申请的内存

32.当一个类对象的生命周期结束后，关于调用析构函数的描述正确的是：（）

- A 如果派生类没有定义析构函数，则只调用基类的析构函数
- B 如果基类没有定义析构函数，则只调用派生类的析构函数
- C 先调用派生类的析构函数，后调用基类的析构函数
- D 先调用基类的析构函数，后调用派生类的析构函数

33.关于类的成员函数，下面哪一种说法是错误的？

- A 一般来说，一个类的成员函数对该类中的数据成员进行读写计算
- B 如果一个数据成员希望是只读的，那么该成员应说明为私有的，而且用一个公有的 getXxx 成员函数来读取它的值
- C 一个类中的一组成员函数不能重名
- D 公有成员函数不应该返回本类的私有成员的指针或引用

34.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。

- A 是
- B 否

35.在下列描述中，错误的是（ ）。

- A 在基类中定义的 **public** 成员在公有继承的派生类中可见，也能在类外被访问
- B 在基类中定义的 **protected** 成员在私有继承的派生类中可见
- C 在基类中定义的公有静态成员在私有继承的派生类中可见
- D 访问声明可以在公有继承派生类中把基类的 **public** 成员声明为 **private** 成员

36.有如下程序段：

```
class A
{
    int _a;
public:
    A(int a): _a(a)
    {
    }
    friend int f1(A &);
    friend int f2(const A &);
    friend int f3(A);
    friend int f4(const A);
};
```

以下调用哪个是错误的：

- A f1(0)
- B f2(0)
- C f3(0)
- D f4(0)

37#include<iostream>

using namespace std;

class Point

{ public:

```
Point(int a=3,int b=5){ X=a;Y=b;}
```

```
int GetX(){return X;}
```

```
int GetY(){return Y;}
```

```
private:
```

```
int X,Y;};
```

现有语句 `Point *p=new Point[2];`则与`(*p).GetX()`等效的表达式是 ()

A `Point[0]. GetX()`

B `p[0].GetX()`

C `p->GetX()`

D `GetX()`

38.说明一个类的对象时，系统自动调用 ()。

A 成员函数

B 构造函数

C 析构函数

D 友元函数

39.如下程序的输出结果是

```
#include<iostream>
```

```
using namespace std;
```

```
long fun(int n)
```

```
{
```

```
    if (n > 2)
```

```
        return (fun(n - 1) + fun(n - 2));
```

```
    else
```

```
        return (n);
```

```
}
```

```
int main()
```

```
{
```

```
    int i;
```

```
    cout << "Needs an integer: ";
```

```
    cin >> i; cout << endl;
```

```
    cout << fun(i) << endl;  
}
```

当分别输入 4 2 时，输出结果分别是？

- A 5, 1 B 4, 2 C 5, 2 D 6, 2

40.下列关于一个类的静态成员的描述中，不正确的是

- A 该类的对象共享其静态成员变量的值
- B 静态成员变量可被该类的所有方法访问
- C 该类的静态方法能访问该类的静态成员变量
- D 该类的静态数据成员变量的值不可修改

41.关于类的成员函数，下面哪一种说法是错误的？

- A 一般来说，一个类的成员函数对该类中的数据成员进行读写计算
- B 如果一个数据成员希望是只读的，那么该成员应说明为私有的，而且用一个公有的 `getXxx` 成员函数来读取它的值
- C 一个类中的一组成员函数不能重名
- D 公有成员函数不应该返回本类的私有成员的指针或引用

42.有关类的说法不正确的是（ ）

- A 类是一种用户自定义的数据类型
- B 类中的成员函数可以存取类中的私有数据
- C 在类中，如果不作特别说明，所有的数据均为私有类型
- D 在类中，如果不作特别说明，所有的成员函数均为公有类型

43.C++中，关于 `new` 运算符的错误描述是？（ ）

- A 运算符 `new` 可以用来动态创建对象和对象数组
- B 使用运算符 `new` 创建对象时，会调用类的构造函数

C 使用 new 运算符创建的对象使用完成后会自动销毁并回收

D 使用运算符 new 创建对象数组时必须定义初始值

44. 下列运算符重载函数中，属于友元函数的是（ ）

A Base operator+(Base);

B Base operator--(Base);

C Base operator&&(Base, Base);

D Base operator++(Base,int);

45. 下列哪种情况会调用拷贝构造函数（ ）

A 用派生类的对象去初始化基类对象时

B 将类的一个对象赋值给该类的另一个对象时

C 函数的形参是类的对象，调用函数进行形参和实参结合时

D 函数的返回值是类的对象，函数执行返回调用时

46. 关于下面程序说法正确的是（ ）

```
class Item
{
private:
char *desc;
int storage;
public:
Item()
{
desc = new char[51];
}
void setInfo(char *dsrc, int num)
{
strcpy(desc,dsrc);
```

```

storage = num;

}

};

int main()

{

Item stock;

stock.setInfo("pen",20);

return 0;

}

```

- A main 函数定义了一个 stock 对象占 8 个字节 B storage 和 desc 成员各占 4 个字节
C 存在内存泄漏的问题 D 存在编译错误

47.下列关于一个类的静态成员的描述中，不正确的是

- A 该类的对象共享其静态成员变量的值
B 静态成员变量可被该类的所有方法访问
C 该类的静态方法能访问该类的静态成员变量
D 该类的静态数据成员变量的值不可修改

48.下面程序的输出结果是_____。

```

#include <iostream.h>

#define SQR(A) A*A

void main() {

    int x=6,y=3,z=2;

    x/=SQR(y+z)/SQR(y+z);

    cout<< x<< endl;

}

```

- A 5 B 6 C 1 D 0

49.在 C++语言中，以下说法正确的是（ ）

- A 多态是通过虚表实现的
- B 构造函数可以声明为虚函数
- C 析构函数不可以声明为虚函数
- D 抽象类中至少包含一个纯虚函数

50.在一个 cpp 文件里面,定义了一个 static 类型的全局变量,下面一个正确的描述是:

- A 只能在该 cpp 所在的编译模块中使用该变量
- B 该变量的值是不可改变的
- C 该变量不能在类的成员函数中引用
- D 这种变量只能是基本类型(如 int,char)不能是 C++类型

51.用于处理字符串流的是（ ）。

- A stringstream
- B ios
- C fstream
- D iostream

52.有关类的说法不正确的是（ ）

- A 类是一种用户自定义的数据类型
- B 类中的成员函数可以存取类中的私有数据
- C 在类中，如果不作特别说明，所有的数据均为私有类型
- D 在类中，如果不作特别说明，所有的成员函数均为公有类型

53.下列关于类中的静态成员的说法错误的是（ ）

- A 虽然静态成员不属于类的某个对象，但是我们仍然可以使用类的对象、引用或者指针来访问静态成员
- B 成员函数不用通过作用域运算符就能直接使用静态成员
- C 静态数据成员不是由类的构造函数初始化的
- D 静态成员不可以作为默认实参

54. 以下程序输出是_____。

```
#include <iostream>

using namespace std;

int main(void)
{
    const int a = 10;

    int * p = (int *)(&a);

    *p = 20;

    cout<<"a = "<<a<<", *p = "<<*p<<endl;

    return 0;
}
```

编译阶段报错 运行阶段报错

A a = 10, *p = 10

B a = 20, *p = 20

C a = 10, *p = 20

D a = 20, *p = 10

55. 对于默认构造函数，下面哪一种说法是错误的？

A 一个无参构造函数是默认构造函数

B 只有当类中没有显式定义任何构造函数时，编译器才自动生成一个公有的默认构造函数

C 默认构造函数一定是一个无参构造函数

D 一个类中最多只能有一个默认构造函数

56. 用 C++ 语言编写的源程序要成为目标程序必须要经过（ ）。

A 解释

B 汇编

C 编辑

D 编译

57. 在下列关于 C++ 函数的叙述中，正确的是（ ）

A 每个函数至少要有有一个参数

B 每个函数都必须返回一个值

C 函数在被调用之前必须先声明

D 函数不能自己调用自己

58.如果有一个类是 myClass，关于下面代码正确描述的是:

```
myClass::~~myClass(){  
    delete this;  
    this = NULL;  
}
```

- A 正确，我们避免了内存泄漏
- B 它会导致栈溢出
- C 无法编译通过
- D 这是不正确的，它没有释放任何成员变量。

59.面向对象的基本特征是（）

- A 封装
- B 继承
- C 重载
- D 多态

60.

```
ClassA *pclassa=new ClassA[5];  
  
delete pclassa;
```

c++语言中，类 ClassA 的构造函数和析构函数的执行次数分别为()

- A 5,1
- B 1,1
- C 5,5
- D 1,5

61 以下关于面向对象的描述错误的是:

- A 面向对象的基本特性是封装，继承和多态
- B 构造函数不可以是虚函数，析构函数可以是虚函数
- C 子类重新定义父类虚函数的方法叫做重载
- D 多态是为了接口重用，封装和继承是为了代码重用

62.基类的公有成员是其私有派生类的（ ）成员

- A 不可见
- B 私有
- C 保护
- D 公有

63.C++ 中，关于 new 运算符的错误描述是？

- A 使用 new 运算符创建的对象使用完成后会自动销毁并回收
- B 使用运算符 new 创建对象数组时必须定义初始值
- C 使用运算符 new 创建对象时，会调用类的构造函数
- D 运算符 new 可以用来动态创建对象和对象数组

64.请将下列构造函数补充完整，使得程序的运行结果是 5

```
#include<iostream>
```

```
using namespace std;
```

```
class Sample{
```

```
    public:
```

```
        Sample(int x){
```

```
            _____
```

```
        }
```

```
        ~Sample(){
```

```
            if(p) delete p;
```

```
        }
```

```
        int show(){
```

```
            return *p;
```

```
        }
```

```
    private:
```

```
        int*p;
```

```
};
```

```
int main(){
```

```
    Sample S(5);
```

```
    cout<<S.show()<<endl;
```

```
    return 0;
```

}

- A. *p=x; B. p=new int(x); C. *p=new int(x); D. p=&x;

65. 请说出 const 与#define 相比,有何优点 ()

- A.宏常量有数据类型,而 const 常量没有数据类型
B.有些集成化的调试工具可以对 const 常量进行调试,但是不能对宏常量进行调试
C.编译器可以对 const 进行类型安全检查。而对#define 只进行字符替换,没有类型安全检查,并且在字符替换可能会产生意料不到的错误。

66. 对使用关键字 new 所开辟的动态存储空间,释放时必须使用 ()

- A.free B.create C.delete D.realse

67. 在 linux+gcc 下,关于以下代码,正确的是()

```
std::string& test_str()
{
    std::string str="test";
    return str;
}

int main()
{
    std::string& str_ref=test_str();
    std::cout<<str_ref<<std::endl;
    return 0;
}
```

- A.编译警告
B.返回局部变量的引用,运行时出现未知错误
C.正常编译且运行

D.把代码里的&都去掉之后,程序可以正常运行

68. 在 C++中，以下定义和初始化的两数组 a1 和 a2，那么下列说法正确的是()。

```
char a1[] = "program";
```

```
char a2[] = {'p','r','o','g','r','a','m'}
```

A.a1 和 a2 完全相同

B.a1 和 a2 不同，a1 是指针

C.a1 和 a2 存储单元的数目相同

D.a1 和 a2 不同，a1 的存储单元数目多

69. 有如下类的定义：

```
int F1();
```

```
static int F2();
```

```
class TestClsaa
```

```
{
```

```
public:
```

```
int F3();
```

```
static int F4();
```

```
}
```

在所描述的函数中，具有隐含 this 指针的是哪个 ()

A.F1

B.F2

C.F3

D.F4

70. 哪种说法是错误的 ()

A.构造函数可以有形参

B.析构函数可以有形参

C.构造函数无任何函数类型

D.析构函数无任何函数类型

71. 有以下程序

```
#include<iostream>
```

```
#include<stdio.h>
```



```
using namespace std;

int main(){

    int m=0123, n = 123;

    printf("%o %o\n", m, n);

    return 0;

}
```

程序运行后的输出结果是()

- | | |
|-------------|------------|
| A.0123 0173 | B.0123 173 |
| C.123 173 | D.173 173 |

72. 下面程序的输出结果是 ()

```
char *p1= "123", *p2 = "ABC", str [50] = "xyz";

strcpy (str+2, strcat (p1,p2));

cout << str;
```

- | | |
|-------------|-----------|
| A.xyz123ABC | B.z123ABC |
| C.xy123ABC | D.出错 |

73. c/c++前提下，下列关于构造函数说法正确的是 ()

- | | |
|------------------|--------------|
| A.构造函数的返回值为 void | B.构造函数不可以被重载 |
| C.构造函数可以是内联函数 | D.构造函数可以没有参数 |

74. 下面说法错误的是()

- A.在组合时,为了保证成员对象被正确清除,在组合类的析构函数中需要显式调用其成员对象的析构函数
- B.在类的继承层次中,可以自动进行向上和向下类型转换.而且都是安全的
- C.构造函数可以重载,析构函数不能重载
- D.C++的派生类如果要覆盖一个继承到的成员函数,在基类中需要将该函数声明为 virtual

75. 对于析构函数，下面哪一种说法是错误的？

- A. 一个对象的生命周期始于构造函数执行，终于析构函数执行
- B. 果类中没有显式定义析构函数，编译器就自动生成一个公有的析构函数。
- C. 一个类中可重载定义多个析构函数
- D. 假设 A 是一个类，那么语句“AasC41; ”将执行析构函数 4 次。

76. 以下程序的输出结果是（ ）

```
void fut(int**s,int p[2][3]) {  
    **s=p[1][1];  
}  
  
void main( ) {  
    int a[2][3]={ 1,3,5,7,9,11 },*p;  
    p=(int*)malloc(sizeof(int));  
    fut(&p,a);  
    cout<< *p;  
}
```

- A.7 B.9 C.1 D.11

77. 下列程序输入：2 空格 0，其运行结果是（ ）

```
#include<iostream>  
  
using namespace std;  
  
int main()  
{  
    int i=0,j=0,n;  
    cin>>n;  
    for(;n!=0; )
```

```

{
    j++;
    cin>>n;
}
if(n=0) i++;
    cout<<i <<" "<<j<<endl;
return 0;
}

```

A.1 0

B.0 0

C.1 1

D.0 1

78. 假设已经定义好了一个类 `student`,现在要定义类 `derived`,它是从 `student` 私有派生的, 定义类 `derived` 的正确写法是()

A.class `derived`: `student private`{...};

B.class `derived`: `protected public`{...};

C.class `derived`: `private student`{...};

D.class `derived`: `public student`{...};

79. 在 C++中, 若有以下定义 `int a,b; float x` , 则正确的赋值语句是 ()。

A.a=1, b=2

B.b++

C.a=b

D.b=int(x)

80. “ () ”处应填写 ()。

```
#include <iostream>
```

```
using namespace std;
```

```
class A
```

```
{
```

```
    public:
```

```
        int m;
```

```
        int* p;
```

```
};
```

```

int main()
{
    A s;

    s.m = 10;

    cout<<s.m<<endl;

    s.p = &s.m;

    (    ) = 5;

    cout<<s.m<<endl;

    return 0;
}

```

A.s.p

B.s->p

C.s.*p

D.*s.p

81. 对象成员的表示方法与结构体变量成员的表示方法相同。该说法 ()

A.正确

B.错误

82.

```

#include<iostream>

using namespace std;

void f(int *p,int *q)
{
    p++;

    *q=*q+1;
}

int main() {

    int m=1,n=2,*r=&m;

    f(r,&n);

    cout<<m<<" "<<n;
}

```

```
    return 0;  
}
```

程序运行后的输出结果是()

A.2,3 B.1,4 C.1,3 D.1,2

83.

```
Class A;
```

```
Class B;
```

```
void F() {
```

```
    A a;
```

```
    B b;
```

```
}
```

在函数 F 中，本地变量 a 和 b 的构造函数(constructor)和析构函数(destructor)的调用顺序是()

A.b 构造 a 构造 a 析构 b 析构

B.a 构造 a 析构 b 构造 b 析构

C.b 构造 a 构造 b 析构 a 析构

D.a 构造 b 构造 b 析构 a 析构

84.

```
#include<iostream>
```

```
using namespace std;
```

```
class TestClass{
```

```
    char x;
```

```
public:
```

```
    TestClass() { cout << 'A'; }
```

```
    TestClass(char c) { cout << c; }
```

```
    ~TestClass() { cout << 'B'; }
```

```
};
```

```
int main() {
```

```

TestClass p1, *p2;

p2 = new TestClass('X');

delete p2;

return 0;
}

```

执行这个程序的结果是 ()

- A.XB B.AXBB C.AAXB D.AAXB BB

85. 下面 C++程序的输出是 ()

```

void f(char * x)
{
    x++;
    *x='a';
}

int main()
{
    char str [sizeof("hello")];
    strcpy(str, "hello");
    f(str);
    cout<<str;
    return 0;
}

```

- A.hello B.hallo C.allo D.以上都不是

86 在 C++语言中, 下面关于数组的描述错误的是 ()

- A.数组的名字就是指向该数组第一个元素的指针
 B.长度为 n 的数组, 下标的范围是 0-n-1

C.数组的大小必须在编译时确定

D.数组可通过值参数、地址指针、引用参数三种方式传递给函数

87. 若有 `char st1[12]= "Yes", st2[5]= "No", st3[7]= "Or";`

则执行下列语句，输出结果是（ ）

```
cout<< strcat(strcat(st1,st3), st2);
```

A.YesOrNo

B.Or No

C.No

D.Or

88.实例对象都是通过调用类的方法生成的，该说法（ ）。

A.正确

B.错误

89. 下面程序的输出结果是（ ）

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

```
void main(){
```

```
    int n[][3]={ 10,20,30,40,50,60};
```

```
    int (*p)[3];
```

```
    p=n;
```

```
    cout<<p[0][0]<<","<<*(p[0]+1)<<","<<(*p)[2]<<endl;
```

```
}
```

A.10,30,60

B.10,30,50

C.10,20,30

D.20,40,60

90. 写出下面代码的运行结果是（ ）

```
class AA {
```

```
public:
```

```
    int a;
```

```
    static int b;
```

```
    AA()
```

```

{
    a = 1;
    add2();
}
int add1()
{
    a = a + 1;
    return a;
}
int add2()
{
    b = b + 1;
    return b;
}
};
int AA::b = 1;
int main()
{
    AA a1;
    a1.add1();
    a1.add2();
    assert(a1.a < a1.b);
    printf("%d,%d", a1.a, a1.b);
    return 0;
}

```

A.2, 3

B.2, 2

C.2, 1

D.断言错误

91. 下面有关 C++ 的类和 C 里面的 struct 的描述，正确的有？

A. 在 C++ 中，来自 class 的继承默认按照 private 继承处理，来自 struct 的继承默认按照 public 继承处理

B. class 的成员默认是 private 权限，struct 默认是 public 权限

C. c 里面的 struct 只是变量的聚合体，struct 不能有函数

D. c++ 的 struct 可有构造和析构函数

92. 在下列选项中，全部都是 C++ 关键字的选项为（ ）。

A. while IF Static

B. break char go

C. sizeof case extern

D. switch float integer

93. c++ 中，声明 const int i, 是在（ ）阶段做到 i 只可读的。

A. 编译

B. 链接

C. 运行

D. 以上都不对

94. 以下程序的输出结果为（ ）

```
using namespace std;
```

```
void print(char **str)
```

```
{
```

```
++str;
```

```
cout<<*str<<endl;
```

```
}
```

```
int main()
```

```
{
```

```
static char *arr[]={ "hello", "world", "c++" };
```

```
char **ptr;
```

```
ptr=arr;
```

```
print(ptr);
```

```
return 0;
```

```
}
```

A.hello

B.world

C.字符 w 的起始地址

D.字符 e

96.分析一下这段程序的输出 ()

```
#include<iostream>
```

```
using namespace std;
```

```
class B
```

```
{
```

```
public:
```

```
    B()
```

```
{
```

```
    cout << "default constructor" << " ";
```

```
}
```

```
    ~B()
```

```
{
```

```
    cout << "destructed" << " ";
```

```
}
```

```
    B(int i): data(i)
```

```
{
```

```
    cout << "constructed by parameter" << data << " ";
```

```
}
```

```
private: int data;
```

```
};
```

```
B Play( B b)
```

```

{
    return b;
}

int main(int argc, char *argv[])
{
    B temp = Play(5);
    return 0;
}

```

- A.constructed by parameter5 destructed destructed
- B.constructed by parameter5 destructed
- C.default constructor" constructed by parameter5 destructed
- D.default constructor" constructed by parameter5 destructed destructed

97. 基类的（ ）在派生类内不能被访问。

- A.私有成员
- B.保护成员
- C.公有数据成员
- D.公有静态数据成员

98.函数 func 的定义如下：

```

void func(const int& v1, const int& v2)
{
    std::cout << v1 << ' ';
    std::cout << v2 << ' ';
}

```

以下代码在 vs 中输出结果为（ ）

```

int main (int argc, char* argv[])
{

```

```
int i=0;

func(++i,i++);

return 0;

}
```

A.0 1 B.1 2 C.2 1 D.2 0

99. 下面关于类定义的说法中，正确的是（ ）

- A.类定义中包括数据成员和函数成员的声明
- B.类成员的缺省访问权限是保护的
- C.数据成员必须被声明为私有的
- D.成员函数只能在类体外进行定义

100. 若有以下说明，则对 n 的正确访问语句是（ ）

```
class Y
{
//... ;

public:

static int n;

};

int Y::n;

Y objY;
```

A.n=1; B.Y::n=1; C.objY::n=1; D.Y>n

101. 下列选项中，正确的 C++标识符是（ ）

A.6_group B.group~6 C.age+3 D._group_6

102. 以下是 C++的不同数据类型值的比较语句，请问这些判断语句中作为条件部分的语句编写

有问题的有()

- A.如果变量 bVar 是布尔类型: if(false==bVar){doSomething();}
- B.如果变量 nVar 是 int 型: if(0==nVar){doSomething();}
- C.如果变量 fVar 为浮点型:if(0.02=fVar){doSomething();}
- D.如果变量 sVar 为字符串型: if (""==sVar){doSomething();}

103. C++对 C 语言做了很多改造, 下列描述中() 使得 C 语言发生了质变, 从面向过程编程了面向对象。

- 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

104. 假设 AA 是一个类, AA* abc () const 是该类的一个成员函数的原型。若该函数返回 this 值, 当用 x.abc ()调用该成员函数后, x 的值是 ()

- A 可能被改变
- B 已经被改变
- C 收到函数调用的影响
- D 不变

105. 若有以下说明, 则对 n 的正确访问语句是 ()。

```
class Y
{
//...;
public:
static int n;
};
int Y::n;
Y objY;
```

- A n=1;
- B Y::n=1;
- C objY::n=1;
- D Y>n

108. 阅读 c++代码输出 ()

```
class base1{
    private: int a,b;
    public:
    base1 ( int i ) : b(i+1),a(b){ }
    base1():b(0),a(b){ }
    int get_a(){return a;}
    int get_b(){return b;}
};

int main()
{
    base1 obj1(11);

    cout<<obj1.get_a()<<endl<<obj1.get_b()<<endl;

    return 0;
}
```

A 12 12

B 随机数 12

C 随机数 随机数

D 12 随机数

109 对于构造方法，下列叙述正确的是 ()。

A 构造方法的优先级一般比代码块低。

B 构造方法的返回类型只能是 void 型。

C 构造方法的主要作用是完成对类的对象的初始化工作。

D 一般在创建新对象时，系统会自动调用构造方法。

110. c/c++前提下，下列关于构造函数说法正确的是 ()

A 构造函数的返回值为 void

B 构造函数不可以被重载

C 构造函数可以是内联函数

D 构造函数可以没有参数

112.以下 () 不是合法的标识符?

A STRING

B x3x

C void

D deSf

113.下列程序输入：2 空格 0，其运行结果是 ()

```
#include<iostream>

using namespace std;

int main()
{
    int i=0,j=0,n;
    cin>>n;
    for(;n!=0; )
    {
        j++;
        cin>>n;
    }
    if(n=0) i++;
    cout<<i <<" "<<j<<endl;
    return 0;
}
```

A 1 0

B 0 0

C 1 1

D 0 1

114.下列哪项不属于 C++的基本数据类型?

A wide

B char

C bool

115.一个类的友元函数能够访问类的

A 私有成员

B 保护成员

C 公有成员

D 所有成员

116.在 C++语言中，下面关于数组的描述错误的是（）

A 数组的名字就是指向该数组第一个元素的指针

B 长度为 n 的数组，下标的范围是 0-n-1

C 数组的大小必须在编译时确定

D 数组可通过值参数、地址指针、引用参数三种方式传递给函数

117.对于子类的构造函数说明，下列叙述中错误的是（）。

A 子类不能继承父类的无参构造函数。

B 子类可以在自己的构造函数中使用 **super** 关键字来调用父类的含参数构造函数，但这个 C 调用语句必须是子类构造函数的第一个可执行语句。

C 在创建子类的对象时，若不含带参构造函数，将先执行父类的无参构造函数，然后再执行自己的无参构造函数。

D 子类不但可以继承父类的无参构造函数，也可以继承父类的有参构造函数。

118.#include<bits/stdc++.h>

using namespace std;

int main(){

int n = 5;

vector<int>a;

set<int>b;

for(int i = 1; i <= n; ++i){

a.push_back(i);

b.insert(i);

}

for(int i = 1; i <= n; ++i){


```

        a.push_back(i);

        b.insert(i);

    }

    cout<<a.size()<<b.size()<<endl;

    return 0;

}

```

程序的输出为()

A 105

B 510

C 1010

D 55

119.下列说法正确的有 ()

A 构造方法的方法名必须与类名相同

B 构造方法也没有返回值，但可以定义为 void

C 在子类构造方法中调用父类的构造方法，super () 必须写在子类构造方法的第一行，否则编译不通过

D 一个类可以定义多个构造方法，如果在定义类时没有定义构造方法，则编译系统会自动插入一个默认的构造方法，这个构造方法不执行任何代码

120.有以下程序

```

#include <iostream>

using namespace std;

class D{

    int d;

    public:

    D(int x=1):d(x){ }

    ~D(){cout<<"D";};

    int main(){

        D d[]={ _____ };
    }
}

```

```
D* p=new D[2];
```

```
delete[]p;
```

```
return 0; }
```

程序运行的结果是 DDDDD，请为横线处选择合适的程序（ ）

A 3,3,3

B D(3), D(3), D(3)

C 3,3,3,3

D D(3,3),D(3,3)

121.关于 C++中的友元函数说法正确的是（ ）

A 友元函数需要通过对象或指针调用

B 友元函数是不能被继承的

C 友元函数没有 this 指针

D 友元函数破坏了继承性机制

122.C++中以下关于函数调用的说法哪个是正确的？

A 传地址后实参和形参指向不同的对象

B 传引用后实参和形参是不同的对象

C 传值后对形参的修改会改变实参的值

D 其他三项都不对

123.当输入数值数据时,对于整型变量只能输入整型值;对于实型变量只能输入实型值。

以上叙述是否正确？

A 正确

B 错误

124.

```
Class A;
```

```
Class B;
```

```
void F() {
```

```
    A a;
```

```
    B b;
```

```
}
```

在函数 F 中，本地变量 a 和 b 的构造函数(constructor)和析构函数(destructor)的调用顺序是：

A b 构造 a 构造 a 析构 b 析构

B a 构造 a 析构 b 构造 b 析构

C b 构造 a 构造 b 析构 a 析构

D a 构造 b 构造 b 析构 a 析构

125.有下列程序

```
using namespace std;
```

```
class SC
```

```
{public:
```

```
SC(int r){R = _____;}
```

```
int Get(){return *R;};
```

```
private:
```

```
int *R;  };
```

```
int main()
```

```
{ SC C(10);
```

```
cout <<C.Get()<< endl;
```

```
return 0;}
```

请将构造函数补充完整，使得程序的运行结果是 10 ()

A new int R

B &r

C r

D *r

126.请将构造函数补充完整，使得程序的运行结果是 10 ()

A new int R

B &r

C r

D *r

127.以下代码是哪一句可能导致的编译错误?

```
#include <iostream>
```

```
using namespace std;
```

```
class Test
```

```
{
```

```
public:
```

```

    Test() { cout << "i like zk"; }

    Test(int) { cout << "i like wangyi"; }

    void exe() { cout << "this is my choice"; }

};

int main()
{
    Test a(1); //1
    a.exe(); //2
    Test b(); //3
    b.exe(); //4
    Test c; //5
    c.exe(); //6
    return 0;
}

```

A 1

B 2

C 3

D 4

E 5

F 6

128.有以下程序

```

#include<iostream>

using namespace std;

class complex
{ public:
    int real;
    int imag;
    complex(int r=0,int i=0)
    { real=r;
      imag=i;}

```

```
};

complex add(complex &a,complex &b)

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

int i=a.imag+b.imag;

return complex(r,i);}

int main()

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

z=add(x,y);

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

程序的输出结果为 4+6i，请为横线处选择合适的程序 ()

- A friend complex add(complex &a,complex &b);
- B friend complex add(complex &,complex &);
- C complex add(complex &a,complex &b);
- D complex add(complex ,complex);

129.面向对象编程包括以下特性

- A 封装
- B 继承
- C 多态
- D 动态类型

130.对静态成员的正确描述是 () ?

- A 静态数据成员要在类外定义和初始化
- B 非静态成员函数也可以操作静态数据成员
- C 全局变量、静态全局变量和静态局部变量都存放在内存的静态存储区域，局部变量存放在内存的栈区
- D 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

131.下列的常量定义中，错误的定义是 ()。

- A const int M=10;
- B const int M=20;

C const char ch;

D const bool mark=true;

132.有如下程序，执行后输出的结果是()

```
#include <iostream.h>

class cla{
    static int n;
public:
    cla(){n++;}
    ~cla(){n--;}
    static int get_n(){return n;}
};

int cla::n= 0;

int main()
{
    cla *p =new cla;
    delete p;
    cout<<"n="<<cla::get_n()<<endl;
    return 0;
}
```

A n=3

B n=4

C n=1

D n=0

133.下面程序的输出结果是_____。

```
#include <iostream.h>

#define SQR(A) A*A

void main() {
    int x=6,y=3,z=2;
    x/=SQR(y+z)/SQR(y+z);
}
```

```
        cout<<x<<endl;
    }
```

A 5

B 6

C 1

D 0

134. 下面程序的输出结果是()

```
#include <iostream>

using namespace std;

int i = 0;

int fun( int n )
{
    static int a = 2;

    a++;

    return(a * n);
}

int main()
{
    int k = 5;

    {
        int i = 2;

        k += fun( i );
    }

    k += fun( i );

    cout << k;

    return(0);
}
```

A 11

B 12

C 13

D 14

135.类成员的访问权限有()

A public

B private

C protected

D static

136.下面程序的输出结果是 ()

```
#include<iostream>

using namespace std;

class ClassA
{
    friend long fun (ClassA a)
    {
        if (a.i < 2) return 1;
        return a.i * fun(ClassA(a.i-1));
    }
public:
    ClassA(long a) { i = a; }
private:
    long i;
};

void main()
{
    int sum = 0;
    for (int i=0; i<4; i++)
    {
        sum += fun(ClassA(i));
    }
    cout << sum;
}
```


A 10

B 12

C 16

D 34

137.下面 3 段程序代码的效果一样吗?

int b;

(1)const int *a = &b;

(2)int const *a = &b;

(3)int *const a = &b;

A (2)=(3)

B (1)=(3)

C (1)=(2)

D 都不一样

E 都一样

138.下列说法正确的是 ()

A 在类方法中可用 **this** 来调用本类的类方法

B 在类方法中调用本类的类方法时可直接调用

C 在类方法中只能调用本类中的类方法

D 在类方法中绝对不能调用实例方法

139.下列说法正确的有 ()

A 在类的一般方法中可用 **this** 来调用本类的属性

B 在类的静态方法中可以访问本类中的属性

C 在类方法中绝对不能调用实例方法

D 在类方法中只能调用本类中的类方法

140.对于子类的构造函数说明, 下列叙述中错误的是 ()。

A 子类不能继承父类的无参构造函数。

B 子类可以在自己的构造函数中使用 **super** 关键字来调用父类的含参数构造函数, 但这个调用语句必须是子类构造函数的第一个可执行语句。

C 在创建子类的对象时, 若不含带参构造函数, 将先执行父类的无参构造函数, 然后再执行自己的无参构造函数。

D 子类不但可以继承父类的无参构造函数，也可以继承父类的有参构造函数。

141.已知 int 占 4 个字节，bool 占 1 个字节。

```
unsigned int value = 1024;
```

```
bool condition = *((bool *)&value);
```

```
if (condition) value += 1; condition = *((bool *)&value);
```

```
if (condition) value += 1; condition = *((bool *)&value);
```

问 value, condition 的值为_____。

A 1026, 1

B 1025, 0

C 1025, 1

D 1024, 0

142.下列关于 C/C++的宏定义的说法中，不正确的是：

A 宏定义的常量更容易理解，如果可以使用宏定义常量的话，要避免使用 const 常量

B 宏的嵌套定义过多会影响程序的可读性，而且很容易出错

C 相对于函数调用，宏定义可以提高程序的运行效率

D 宏定义不检查参数正确性，这样会有安全隐患

143.静态初始化程序会在构造函数之前执行，这句话是否正确？

A 正确

B 错误

144.关于 C++中的友元函数说法正确的是（ ）

A 友元函数需要通过对象或指针调用

B 友函数是不能被继承的

C 友元函数没有 this 指针

D 友元函数破坏了继承性机制

145.有以下程序

```
#include <iostream>

using namespace std;

_____

void One (float one) { cout << "1" << endl; }

void Two (float two) { cout << "2" << endl; }

void Three(float three) { cout << "3" << endl; }

int main( ) {

    float i=1,j=2,k=3;

    function = One;

    function(i);

    function= Two;

    function(j);

    function = Three;

    function(k);

}
```

请为横线处选择合适的程序使得程序的运行结果是 123 ()

A void *function();

B void *function(float);

C void (*function)();

D void (*function)(float)

146.以下关于构造函数的描述错误的是 ()

- A.每个类有且只能有一个构造函数。
- B.构造函数是类的一种特殊函数，它的方法名必须与类名相同
- C.构造函数的主要作用是完成对类的对象的初始化工作
- D.一般在创建新对象时，系统会自动调用构造函数

147. 有以下程序

```
#include <stdio.h>

void fun ( int n ,int *s ) {

    int f;

    if ( n == 1 )

        *s = n + 1;

    else{

        fun( n-1, &f);

        *s = f;

    }

}

main(){

    int x =0;

    fun( 4,&x );

    printf("%d\n",x);

}
```

程序的输出结果是？

A.3

B.1

C.2

D.4

148. 对于以下变量定义，正确的赋值是()。

```
int *p[3], a[3];
```

A.p=a

B.*p=a [0]

C.p=&a [0]

D.p[0]=&a[0]

149. 有以下程序

```
#include<iostream>
```

```
using namespace std;
```

```
class complex{
```

```
public:
```

```
int real;
```

```
int imag;
```

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

```
real=r;
```

```
imag=i;
```

```
}
```

```
};
```

```
complex add(complex &a,complex & b) {
```

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

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

```

        return complex(r,i);

    }

int main() {

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

    z=add(x,y);

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

}

```

程序的输出结果为 4+6i，请为横线处选择合适的程序 ()

- A.friend complex add(complex &a,complex & b);
- B.friend complex add(complex &,complex &);
- C.complex add(complex &a,complex & b);
- D.complex add(complex ,complex);

150. struct 和 class 的区别?

- A.struct 的成员默认是公有的
- B.struct 的成员默认是私有的
- C.类的成员默认是私有的
- D.类的成员默认是公有的

151. 以下哪项描述是正确的

- A.父类的构造函数与析构函数不会自动被调用
- B.成员变量需要用 public protected private 修饰，在定义变量时不再需要 var 关键字
- C.父类中定义的静态成员，不可以在子类中直接调用
- D.包含抽象方法的类必须为抽象类，抽象类不能被实例化

152. 在 C++ 中，引用和指针的区别是（ ）

- A. 引用总是指向一个对象, 指针可能不指向对象
- B. 引用和指针都可以被重新赋值
- C. 引用不能用 `const` 修饰，而指针可以
- D. 引用创建时必须初始化，而指针则可以在任何时候被初始化

153. 定义析构函数时，应该注意：（ ）

- A. 其名与类名完全相同
- B. 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

154. C++ 中的拷贝构造函数在下面哪些情况下会被调用（ ）

- A. 对象创建的时候
- B. 使用一个类的对象去初始化该类的一个新对象
- C. 被调用函数的形参是类的对象
- D. 当函数的返回值是类的对象时，函数执行完成返回调用者

155. 哪些因素可能会影响到一个类的大小（多选）

- A. 成员个数
- B. 函数个数
- C. 是否有虚函数
- D. 虚函数个数
- E. 静态变量个数
- F. 对齐方式

156. 可以把任何一种数据类型的变量赋给 `Object` 类型的变量。

A.对

B.错

157. #include<iostream>

using namespace std;

void f(int *p,int *q) {

 p++;

 *q=*q+1;

}

int main() {

 int m=1,n=2,*r=&m;

 f(r,&n);

 cout<<m<<" "<<n;

 return 0;

}

程序运行后的输出结果是()

A.2,3

B.1,4

C.1,3

D.1,2

158. Test 执行后的输出是:

void Test {

 class B {

 public:

 B(void){ cout << "B\t"; }


```

        ~B( void ){ cout << "~B\t"; }

};

struct C {

    C( void ){ cout << "C\t"; }

    ~C( void ){ cout << "~C\t"; }

};

struct D : B {

    D { cout << "D\t"; }

    ~D { cout << "~D\t"; }

    private: C c;

};

D d;

}

A.  B C D ~D ~C ~B
B.  D C B ~B ~C ~D
C.  C D B ~B ~D ~C
D.  C ~C D ~D B ~B

```

159. 类所实现的接口以及修饰不可以是（）。

- A. void B. public C. abstract D. final

160. 以下 C 语言和 C++语言比较中错误的是（）

- A. C 语言绝大部分语法适用于 C++
- B. C 语言和 C++的 struct 意义和用法一致

- C. C 语言不能使用继承技术，但 C++ 可以
- D. C++ 是面向对象的语言，C 语言是面向过程的语言

161. 在（ ）情况下适宜采用 inline 定义内联函数

- A. 函数体含有循环语句
- B. 函数体含有递归语句
- C. 函数代码少、频繁调用
- D. 函数代码多，不常调用

162. 基类的公有成员是其私有派生类的（ ）成员

- A. 不可见
- B. 私有
- C. 保护
- D. 公有

163. 关于变量及其作用范围描述正确的是（ ）：

- A. 实例变量是类的成员变量
- B. 静态变量用关键字 static 声明
- C. 在方法体中定义的局部变量在该方法被执行时创建
- D. 局部变量在使用前可不用初始化

164. 下面中哪两个可以在 Demo 的子类中使用：（ ）

```
class Demo {  
  
    protected int m1 (int a, int b) { return 0; }  
  
}
```

- A. public int m1 (int a, int b) { return 0; }
- B. private int m1 (int a, int b) { return 0; }

- C. `private int m1 (int a, long b) { return 0; }`
- D. `public short m1 (int a, int b) { return 0; }`

165. 面向对象编程包括以下特性

- A. 封装
- B. 继承
- C. 多态
- D. 动态类型

166. 外部变量可以供其所在的程序文件中的任何函数使用。 ()

- A. 正确
- B. 错

167. 用友元数实现运算符重载，下面哪一种说法是错误的？

- A. 因为友元函数没有当前对象，因此要定义单目运算符，就需要单参函数，要定义双目运算符，就需要双参函数
- B. 定义后置“++”或后置“- -”运算是特例，它们是单目运算符，但需要两个形参，头一个形参是作用对象，后一个是 `int` 形参
- C. 用友元函数可以定义成员函数不能实现的运算，例如一些双目运算符，右操作数是本类对象，而左操作数不是本类对象
- D. 一个类说明的友元函数可以被派生类继承

168. 下面代码中，如果要输出 `NAME` 的值，该用哪个代码运行()

```
class Computer{  const NAME='dell'; }
```

- A. `echo Computer::NAME;`
- B. `Computer->NAME`
- C. 无法直接输出
- D. `echo Computer.NAME`

169. 在 C++中，一个函数功能不太复杂，但要求被频繁调用，则应该把它定义为（）

- A. 递归函数 B. 嵌套函数 C. 重载函数 D. 内联函数

170. c++中，声明 `const int i`是在哪个阶段做到 `i` 只可读的？

- A. 编译 B. 链接 C. 运行 D. 以上都不对

171. 基类的公有成员是其私有派生类的（ ）成员

- A. 不可见 B. 私有 C. 保护 D. 公有

172. 在创建派生类对象，构造函数的执行顺序（）

- A. 基类构造函数，派生类对象成员构造函数，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，派生类对象成员构造函数
D. 对象成员构造函数，基类构造函数，派生类本身的构造函数

173. 请问下列关于构造函数的说法中，哪一种不正确？

- A. 构造函数可以为 `const` 函数
B. 构造函数不能为虚函数
C. 默认构造函数不会初始化复合类型成员变量
D. 类的成员变量在构造函数中初始化

174. 对于派生类的构造函数，创建对象时构造函数执行的顺序为？（ ）

1. 成员对象构造函数

2. 基类的构造函数

3. 派生类本身的构造函数

A. 123

B. 231

C. 321

D. 213

175. 下面有关友元函数的描述中，正确的说法是（ ）

A. 友元函数是独立于当前类的外部函数

B. 一个友元函数不可以同时定义为两个类的友元函数

C. 友元函数必须在类的外部进行定义

D. 在类的外部定义友元函数时必须加上 `friend` 关键字

176. 成员函数（不包括析构函数）与一般的函数一样可以重载、内联和设置参数的默认值。
这句话是否正确

A. 正确

B. 错误

177. `Class demo{Const C = "const";}`, 以下调用 `demo` 类中的常量 `C` 正确的是:

A. `$Cls = new demo(); $Cls->C;`

B. `Demo::$C;`

C. `Demo::C;`

D. `Demo->$C;`

178. 友元函数说明在类体内，它是一种成员函数。

A. 正确

B. 错误

179. 在声明类时，下面的说法正确的是（ ）。

- A. 可以在类的声明中给数据成员赋初值（c++11 新标准）
- B. 数据成员的数据类型可以是 register
- C. private、public、protected 可以按任意顺序出现
- D. 没有用 private、public、protected 定义的数据成员是公有成员

180. 在下列关于 C++函数的叙述中，正确的是（ ）

- A. 每个函数至少要有一个参数
- B. 每个函数都必须返回一个值
- C. 函数在被调用之前必须先声明
- D. 函数不能自己调用自己

181. this 代表了()的对象引用,super 表示的是当前对象的()对象?

- A. 当前类 当前类
- B. 当前类的父类 当前类
- C. 当前类 当前类的父类
- D. 以上都不正确

182. 对象成员的表示方法与结构体变量成员的表示方法相同。说法是否正确?

- A. 正确
- B. 错误

183. 静态初始化程序会在构造函数之前执行，这句话是否正确?

- A. 正确
- B. 错误

184. 对于子类的构造函数说明，下列叙述中错误的是（ ）。

- A. 子类不能继承父类的无参构造函数。

- B. 子类可以在自己的构造函数中使用 `super` 关键字来调用父类的含参数构造函数，但这个调用语句必须是子类构造函数的第一个可执行语句。
- C. 在创建子类的对象时，若不含带参构造函数，将先执行父类的无参构造函数，然后再执行自己的无参构造函数。
- D. 子类不但可以继承父类的无参构造函数，也可以继承父类的有参构造函数。

185. 下面有关值类型和引用类型描述正确的是（ ）？

- A. 值类型的变量赋值只是进行数据复制，创建一个同值的新对象，而引用类型变量赋值，仅仅是把对象的引用的指针赋值给变量，使它们共用一个内存地址。
- B. 值类型数据是在栈上分配内存空间，它的变量直接包含变量的实例，使用效率相对较高。而引用类型数据是分配在堆上，引用类型的变量通常包含一个指向实例的指针，变量通过指针来引用实例。
- C. 引用类型一般都具有继承性，但是值类型一般都是封装的，因此值类型不能作为其他任何类型的基类。
- D. 值类型变量的作用域主要是在栈上分配内存空间内，而引用类型变量作用域主要在分配的堆上。

186. 面向对象的三个基本元素是什么？

- A. 多态
- B. 继承
- C. 封装
- D. 重载

187. 以下说法正确的是（ ）

- A. 内联（`inline`）函数改善了函数调用的执行效率
- B. 类 A 的友元(`friend`)函数可以访问类 A 的私有成员
- C. 类 A 的友元(`friend`)类 B 可以访问类 A 的私有成员

- D. 类 A 的静态数据成员为类 A 的所有对象所共享
- E. 类 A 的静态成员函数没有传递 this 指针作为参数

188. 下列 const 使用方法错误的是 ()

- A. const int Val = 10; Val = 20;
- B. class A { const int SIZE = 100; int array[SIZE]; };
- C. class A { protected: static int const Inity; };
- D. int a=7; const int *aPtr; aPtr = &a;

189. 下列程序的运行结果是 1*2 3*4, 请将横线处缺失程序补充完整 ()

```
class Test{  
  
    public:  
  
    Test(int a,int b) {x=a;y=b;}  
  
    ~Test() {}  
  
    void print() { cout<<x<<"*"<<y<<" "; }  
  
    private:  
  
    int x;  
  
    int y;  
  
};  
  
int main(){  
  
    Test t[2]={      };  
  
    for(int i=0;i<2;i++)
```



```
t[i].print();  
  
return 0;  
  
}
```

- A. 1,2,3,4
- B. (1,2),(3,4)
- C. Test(1,2),Test(3,4)
- D. 1234

190. 哪种说法是错误的 () ?

- A. 构造函数可以有形参
- B. 析构函数可以有形参
- C. 构造函数无任何函数类型
- D. 析构函数无任何函数类型

191. constructor 必须与 class 同名,但方法不能与 class 同名?下列说法正确的是()

- A. class 中的 constructor 不可省略
- B. constructor 必须与 class 同名,但方法不能与 class 同名
- C. constructor 在一个对象被 new 时执行
- D. 一个 class 只能定义一个 constructor

192. 以下代码的执行结果为()

```
class a{  
  
    protected $c;  
  
    public function a() { $this->c = 10; }
```

```

}

class b extends a{

    public function print_data(){ return $this->c; }

}

$b = new b();

echo $b->print_data();

```

A.9 B.10 C.11 D. null

193. (C++部分) 下列关于对象初始化的叙述中, 正确的是:

- A. 定义对象的时候不能对对象进行初始化
- B. 定义对象之后可以显式地调用构造函数进行初始化
- C. 定义对象时将自动调用构造函数进行初始化
- D. 在一个类中必须显式地定义构造函数实现初始化

194. 下面对对象概念描述正确的是 ()

- A. 对象间的通信靠消息传递
- B. 对象是名字和方法的封装体
- C. 任何对象必须有继承性
- D. 对象的多态性是指一个对象有多个操作

195. 关于构造函数的描述下列说法正确的是

- A. 构造函数不能被重载
- B. 构造函数不能被覆盖
- C. 一个构造函数可以返回一个私有的或一个对象的引用

D. 构造函数代码执行顺序时是从最远的祖先类到当前的类层级

196. 有以下程序

```
#include <iostream>

using namespace std;

class D{

    int d;

    public:

    D(int x=1):d(x){}

    ~D() { cout<<"D";}

};

int main(){

    D d[ ]={_____};

    D* p=new D[2];

    delete[]p;

    return 0;

}
```

程序运行的结果是 DDDDD，请为横线处选择合适的程序（ ）

A. 3,3,3

B. D(3), D(3), D(3)

C. 3,3,3,3

D. D(3,3),D(3,3)

197. 类成员的访问权限中，Private 只能被本类的成员函数和其友元函数访问（ ）

A. 对

B. 错

198. 在派生类的函数中 能够直接访问基类的 公有成员和保护成员。这句话是否正确？

A. 正确

B. 错误

199. 若有以下类 Z 说明，则函数 fStatic 中访问数据 a 错误的是（ ）。

```
class Z{  
  
    static int a;  
  
    public:  
  
    static void fStatic(Z&);  
  
};
```

```
int Z::a=0;
```

```
Z objZ;
```

A. void Z::fStatic() { objZ.a =1; }

B. void Z::fStatic() { a = 1; }

C. void Z::fStatic() { thisa = 0; }

D. void Z::fStatic() { Z::a = 0; }

200. 下列代码中的错误是（ ）

```
public class Test {  
  
    public static void main(String [] args) {  
  
        int i;
```

```

        i+=1;

    }

}

```

- A. 非法的表达式 `i+=1` B. 找不到符号 `i`
- C. 类不应为 `public` D. 尚未初始化变量 `i`

```

201. class Chinese{

    private static Chinese objref =new Chinese();

    private Chinese(){ }

    public static Chinese getInstance() { return objref; }

}

```

```

public class TestChinese {

    public static void main(String [] args) {

        Chinese obj1 = Chinese.getInstance();

        Chinese obj2 = Chinese.getInstance();

        System.out.println(obj1 == obj2);

    }

}

```

- A. true B. false C. TRUE D. FALSE

202. 以下 C++代码会输出什么?

```

#include

```

```
using namespace std;
```

```
class A {
```

```
    A() {}
```

```
    ~A() {}
```

```
    static int a;
```

```
};
```

```
int main(){
```

```
    cout << sizeof(A) << endl
```

```
    return 0;
```

```
}
```

A. 0

B. 1

C. 4

D. 8

203 已知：类 A 中一个成员函数说明如下：void Set(A &a);其中，A &a 的含意是

A 指向类 A 的指针为 a;

B 将 a 的地址值赋给变量 Set;

C a 是类 A 的对象引用，用来作函数 Set（）的形参;

D 变量 A 与 a 按位相与作为函数 Set（）的参数。

204 下列哪条语句使用方法是正确的（ ）

```
public class Example{
```

```
    private float f=1.0f;
```

```
    int i=12;
```

```
    static int j=1;
```

```
    public static void main(String args[]){
```

```
        Example test=new Example();
```

```
    }
```

}

A test.f = 1.0

B this.j

C Example.i

D Example.j

205 下列哪种情况会调用拷贝构造函数（）

A 用派生类的对象去初始化基类对象时

B 将类的一个对象赋值给该类的另一个对象时

C 函数的形参是类的对象，调用函数进行形参和实参结合时

D 函数的返回值是类的对象，函数执行返回调用时

206 看以下代码：

```
class A
```

```
{
```

```
    public:
```

```
    ~A();
```

```
};
```

```
A::~~A()
```

```
{
```

```
    printf("delete A ");
```

```
}
```

```
class B : public A
```

```
{
```

```
    public:
```

```
    ~B();
```

```
};
```

```
B::~~B()
```

```
{
```

```
    printf("delete B ");
```

```
}
```

请问执行以下代码

A *pa = new B();

delete pa;

输出的串是 ()

A delete A

B delete B

C delete B delete A

D delete A delete B

207 创建对象时系统自动调用相应的构造函数为对象初始化.这句话是否正确

A 正确

B 错误

208 下列运算符重载函数中, 属于友元函数的是 ()

A Base operator+(Base);

B Base operator--(Base);

C Base operator&&(Base, Base);

D Base operator++(Base,int);

209 如果 MyClass 为一个类, 执行 " MyClass a[5], *b[6]" 语言会自动调用该类构造函数的次数是()

A 2 B 5 C 4 D 9

210 派生类只继承基类中的公有成员和保护成员,而不继承私有成员.说法是否正确?

A 正确 B 错误

211 关于类的成员函数, 下面哪一种说法是错误的?

A 一般来说, 一个类的成员函数对该类中的数据成员进行读写计算

B 如果一个数据成员希望是只读的, 那么该成员应说明为私有的, 而且用一个公有的 getXxx 成员函数来读取它的值

C 一个类中的一组成员函数不能重名

D 公有成员函数不应该返回本类的私有成员的指针或引用

有以下说明语句：

```
struct Worker
{
    int no;
    char name[20];
};

Worker w, *p=&w;
```

下列错误的引用是（ ）。

- | | |
|------------|----------|
| A. w.no | B. p->no |
| C. (*p).no | D. *p.no |

218. 若有以下说明，则在类外使用对象 objX 成员的正确语句是（ ）。

```
class X
{
    int a;
    void fun1();
public:
    void fun2();
};

X objX;
```

- A. objX.a=0;
- B. objX.fun1();
- C. objX.fun2();
- D. X::fun1();

219 下列运算符重载函数中，属于友元函数的是（ ）

- A. Base operator+(Base);
- B. Base operator--(Base);

C.Base operator&&(Base, Base);

D.Base operator++(Base,int);

220.若 MyClass 是一个类名，其有如下语句序列：

MyClass c1, *c2;

MyClass *c3=new MyClass;

MyClass &c4=c1;

上面的语句序列调用构造函数个数是()。

A.1 B.2 C.3 D.4

221 假设 A 为抽象类，下列声明 () 是正确的。

A.int fun(A);

B.A Obj;

C.A fun(int);

D.A *p;

222.在下面的类定义中，横线上应填入的内容是 ()

Class Fred

{public:

void print{ } {cout<<DATA<<endl;}

void setData(double d){data = d;}

static int cout;

private:

double data;};

_____count = 0;

}

A.static int Fred::

B.int Fred::

C.int

D.static int

223.用于包含 C++标准库 iostream 的预处理指令是 ()

A.#define<iostream>

B.#include<iostream>

C.#define'iostream'

D.#include'iostream'

224 下列有关 this 指针使用方法的叙述正确的是 ()

保证基类保护成员在子类中可以被访问

保证基类私有成员在子类中可以被访问

保证基类共有成员在子类中可以被访问

保证每个对象拥有自己的数据成员，但共享处理这些数据的代码

225.#include<iostream>

#include<string>

using namespace std;

class A

{

friend long fun(A s)

{

if (s.x<3) {

return 1;

}

return s.x+fun(A(s.x - 1));

}

public:

A(long a)

{

x = a--;

```

    }

private:

    long x;

};

int main()

{

    int sum=0;

    for( int i=0; i<5; i++) {

        sum += fun(A(i));

    }

    cout<<sum;

}

```

代码运行结果是？

- | | |
|------|------|
| A.21 | B.15 |
| C.9 | D.36 |

226.print()函数是一个类的常成员函数，它无返回值，下列表示中正确的是()

```

const void print();

void const print();

void print() const;

void print(const);

```

227.关于内联函数正确的是（）

- A.类的私有成员函数不能作为内联函数
- B.在所有类说明中内部定义的成员函数都是内联函数

C.类的保护成员函数不能作为内联函数

D.使用内联函数的地方会在运行阶段用内联函数体替换掉

228.类中的成员函数都是公有的,数据成员都是私有的.这句话是否正确?

A.正确

B.错误

229.对于 C/C++语言的函数，下列叙述中正确的是（ ）

A.函数的定义不能嵌套，但函数调用可以嵌套

B.函数的定义可以嵌套，但函数调用不能嵌套

C.函数的定义和调用都不能嵌套

D.函数的定义和调用都可以嵌套

230.关于引用与指针的区别，下面叙述错误的是：

A.引用必须被初始化，指针不必。

B.指针初始化以后不能被改变，引用可以改变所指的對象。

C.删除空指针是无害的，不能删除引用。

D.不存在指向空值的引用，但是存在指向空值的指针。

231.在 C++语言中，下列说法正确的是：

A. `inline` 函数表示编译器一定会将函数体直接插入到调用此函数的地方，这样可以加快程序的运行速度

B. 局部变量和全局变量不能重名

C. 静态局部变量内存是在运行期分配的，存储在堆栈区

D. C++不是类型安全的语言

232.下列 描述中，（ ） 是抽象类特有的

A. 不可以声明虚函数

B. 不可以定义友元函数

C. 不可以进行构造函数重载

D. 不可以实例化

233.关于类与对象，下面哪一种说法是错误的？

- A. 一个对象是某个类的一个实例
- B. 一个实例是某个类型经实例化所产生的一个实体
- C. 创建一个对象必须指定被实例化的一个类
- D. 一个类的多个对象之间不仅持有独立的数据成员，而且成员函数也是独立的

234.在声明类时,关键字 `private`、`public` 和 `protected` 出现 任意次数。说法是否正确?

- A. 正确
- B. 错误

235.下列哪条语句使用方法是正确的 ()

```
public class Example{  
  
    private float f=1.0f;  
  
    int i=12;  
  
    static int j=1;  
  
    public static void main(String args[]){  
  
        Example test=new Example();  
  
    }  
}
```

- | | |
|----------------|-------------|
| A.test.f = 1.0 | B.this.j |
| C.Example.i | D.Example.j |

236.有下列类定义

```
#include<iostream>  
  
using namespace std;  
  
class Point  
{ public:  
    Point(int a=3,int b=5){ X=a;Y=b;}  
  
    int GetX(){return X;}  
  
    int GetY(){return Y;}  
  
private:  
    int X,Y;};
```

现有语句 `Point *p=new Point[2];`则与`(*p).GetX()`等效的表达式是 ()

- A. `Point[0]. GetX()`
- B. `p[0].GetX()`
- C. `p->GetX()`
- D. `GetX()`

237.C++定义函数时，函数必须有返回值，`void` 不算返回值。

- A. 正确
- B. 错误

238.c/c++中，关于类的静态成员的不正确描述是 ()。

- A. 静态成员不属于对象，是类的共享成员
- B. c++11 之前，非 `const` 的静态数据成员要在类外定义和初始化
- C. 静态成员函数不拥有 `this` 指针，需要通过类参数访问对象成员
- D. 只有静态成员函数可以操作静态数据成员

239.下列程序的运行结果是 1，请将横线处缺失程序补充完整 ()

```
class C
{
public:
    C(int a=0);
    void Show();
private:
    int H;
}
void C::Show()
{
    cout<<H<<endl;
}
C::C(int a)
{
    _____;
}
int main()
{
    C m(1);
    m.Show();
}
```

- A. `a=H`
- B. `H=0`
- C. `H=a`
- D. `a=1`

240.下列说法正确的是:

- A. 内联函数在运行时是将该函数的目标代码插入每个调用该函数的地方
- B. 内联函数在编译时是将该函数的目标代码插入每个调用该函数的地方
- C. 类的内联函数必须在类体内定义
- D. 类的内联函数必须在类体外通过关键字 inline 定义

241.有如下程序:

```
#include <math.h>

#include <iostream>

using namespace std;

class Point
{
    friend double Distance( const Point & p1, const Point & p2 )    /* ① */
    {
        double dx  = p1.x_ - p2.x_;
        double dy  = p1.y_ - p2.y_;
        return(sqrt( dx * dx + dy * dy ));
    }
public:
    Point( int x, int y ) : x_( x ), y_( y )                        /* ② */
    {
    }
private:
    int    x_;
    int    y_;
};

int main( void )
{
    Point  p1( 3, 4 );
```

```

Point p2( 6, 9 );

cout << Distance( p1, p2 ) << endl;          /* ③ */

return(0);

}

```

下列叙述中正确的是

- A 程序编译正确
- B 程序编译时语句①出错
- C 程序编译时语句②出错
- D 程序编译时语句③出错

242、类 A 是类 B 的友元,类 C 是类 A 的公有派生类,忽略特殊情况则下列说法正确的是()

- A 、类 B 是类 A 的友元
- B 、类 C 不是类 B 的友元
- C 、类 C 是类 B 的友元
- D 、类 B 不是类 A 的友元

243、撤销对象时，系统自动调用（ ）。

- A、成员函数
- B、构造函数
- C、析构函数
- D、友元函数

244、（ ）面向对象程序设计语言不同于其他语言的主要特点。

- A、继承性
- B、消息传递
- C、多态性
- D、封装性

245 下面描述中，错误的是（ ）

- A、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、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、基类定义的 public 和 protected 成员在保护继承的派生类中不可见
- D、基类定义的 protected 成员在 protected 继承的派生类中可见，也能在类外被访问

246、用 class 关键字定义的类，其成员默认访问属性为（ ）

- A 、 private
- B、 protected

C、 public

D 、无定义

247、下面有关类的静态成员和非静态成员，说法错误的是？

A、静态成员存在于内存，非静态成员需要实例化才会分配内存

B、非静态成员可以直接访问类中静态的成员

C、静态成员能访问非静态的成员

D、非静态成员的生存期决定于该类的生存期，而静态成员生存期则与程序生命期相同

248、能实现删除文件功能的语句是（ ）。

A ofstream fs("date.dat", ios::trunc);

B ifstream fs("date.dat", ios::trunc);

C ofstream fs("date.dat", ios::out);

D ifstream fs("date.dat", ios::in);

249、在 c++中，下列描述错误的是（ ）

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

250.拷贝构造函数的特点是（ ）

A、该函数名同类名，也是一种构造函数，该函数返回自身引用

B、该函数只有一个参数，是对某个对象的引用

C、每个类都必须有一个拷贝初始化构造函数，如果类中没有说明拷贝构造函数，则编译器系统会自动生成一个缺省拷贝构造函数，作为该类的保护成员

D、拷贝初始化构造函数的作用是将一个已知对象的数据成员值拷贝给正在创建的另一个同类的对象

251、友元的作用之一是（ ）

A、提高程序的运行效率

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

252、关于 C++/JAVA 类中 static 成员和对象成员的说法正确的是？

A、static 成员变量在对象构造时生成

- B、static 成员函数在对象成员函数中无法调用
- C、虚成员函数不可能是 static 成员函数
- D、static 成员函数不能访问 static 成员变量

253、问以下说法，哪个是正确的：

- A、每个类都有一个无参数的构造函数。
- B、每个类都有一个拷贝构造的函数。
- C、每个类能有多个构造函数。
- D、每个类能有多个析构函数。

254、下列程序执行后输出的结果是

```
#include<iostream>

using namespace std;

int f(int a)
{
    int b = 0;

    static int c = 3;

    a = c++,b++;

    return (a);
}

int main()
{
    int a = 2, i, k;

    for (i = 0; i < 2; i++)

        k = f(a++);

    printf(" % d\n", k);

    return 0;
}
```

A 3 B 0 C 5 D 4

255、下列情况中，不会调用拷贝构造函数的是（ ）

- A 用一个对象去初始化同一个类的另一个新对象时

- B 将类的一个对象赋值给该类的另一个对象时
- C 函数的形参对象，调用函数进行形参和实参结合时
- D 函数的返回值是类的对象，函数执行返回调用时

256、重载（overload）和重写（override）的区别？

- A 重载：是指允许存在多个同名函数，而这些函数的参数表不同
- B 重写：是指子类重新定义复类虚函数的方法
- C 重写：是指允许存在多个同名函数，而这些函数的参数表不同
- D 重载：是指子类重新定义复类虚函数的方法

257、类成员的访问权限有

- A public
- B private
- C protected
- D static

258.下面对析构函数的正确描述是（）。

- A 系统在任何情况下都能正确析构对象
- B 用户必须定义类的析构函数
- C 析构函数没有参数，也没有返回值
- D 析构函数可以设置默认参数

259.如果一个类 `class A{}` 没有构造函数定义，那么当这个类的 Objects 被创建时会发生什么？

（ ）

- A 编译器产生错误
- B 运行时会产生错误
- C 编译器会为此类提供默认构造函数去创建 Objects
- D 以上都不对

260.对于任意一个类，析构函数的个数最多为（ ）

- A. 0
- B. 1
- C. 2
- D. 3

261. 下列关于构造函数的描述中，错误的是（ ）。

- A 构造函数可以重载
- B 构造函数名同类名
- C 带参数的构造函数具有类型转换作用
- D 构造函数是系统自动调用的

262. 以下哪些做法是不正确或者应该极力避免的：【多选】（ ）

- A 构造函数声明为虚函数
- B 派生关系中的基类析构函数声明为虚函数
- C 构造函数中调用虚函数
- D 析构函数中调用虚函数

263. c/c++语言中，关于类的静态成员的不正确描述是（ ）

- A 静态成员不属于对象，是类的共享成员
- B c++11 之前，非 const 的静态成员要在类外定义和初始化
- C 静态成员函数不拥有 this 指针，需要通过类参数访问对象成员
- D 只有静态成员函数可以操作静态数据成员

264. 假定一个类的构造函数为 B (int x, int y) {a=x--; b=x*y--; }。则执行 B (3,5); 语句后，a 和 b 的值分别是（ ）

- A 3 和 10
- B 2 和 10
- C 3 和 15
- D 2 和 15

265. 下列程序的运行结果是（ ）

```
#include <iostream>
#include <cmath>
using namespace std;
class Point
{public:
    Point(int X=0, int Y=0);
    Point(Point &p);
    int GetX() {return X;}
```

```

int GetY() {return Y;}

static int countP;

static void GetC(Point A,Point B)

{int z;

z=sqrt((B.X-A.X)*(B.X-A.X)+(B.Y-A.Y)*(B.Y-A.Y));

cout<<z<<endl;}

private:

int X,Y;    };

Point::Point(int X, int Y)

{this->X=X;this->Y=Y;

countP++;}

Point::Point(Point &p)

{ X=p.X; Y=p.Y; countP++;}

int Point::countP=0;

int main()

{ Point D(3,4),*p;

Point E(D);

p=&E;

void (*q)(Point,Point)=Point::GetC;

(*q)(D,E);

return 0;

}

```

A. 2 B. 0 C. 3 D. 4

266. 下面的程序输出可能是什么?

```

class Printer{

    public:

        Printer(std::string name) {std::cout << name;}

};

class Container{

```

```

public:
    Container() : b("b"), a("a") {}

    Printer a;

    Printer b;

};

int main(){
    Container c;

    return 0;

}

```

- A 可能是 "ab" 或 "ba"。 依赖于具体的实现
- B 一直都是 "ba"
- C 一直都是 "ab"

267.以下代码是否完全正确，执行可能得到的结果是_____。

```

class A{
    int i;
};

class B{
    A *p;
public:
    B(){p=new A;}
    ~B(){delete p;}
};

void sayHello(B b){
}

int main(){
    B b;

    sayHello(b);

}

```


- A 程序正常运行 B 程序编译错误
- C 程序崩溃 D 程序死循环

268.下面对析构函数的正确描述是（）。

- A 系统在任何情况下都能正确析构对象 B 用户必须定义类的析构函数
- C 析构函数没有参数，也没有返回值 D 析构函数可以设置默认参数

269.下面程序段包含 4 个函数,其中具有隐含 this 指针的是()

```
int f1();

class T
{
    public:static int f2();
    private:friend int f3();
    protect:int f4();
};
```

- A. f1 B. f2 C. f3 D. f4

270. 若有以下类 T 说明，则函数 fFriend 的错误定义是（ ）。

```
class T
{
    int i;

    friend void fFriend( T&, int );
};
```

- A void fFriend(T &objT, int k) { objT.i = k; }
- B void fFriend(T &objT, int k) { k = objT.i; }
- C void T::fFriend(T &objT, int k) { k += objT.i; }
- D void fFriend(T &objT, int k) { objT.i += k; }

271.在 c/c++语言中，关于类的静态成员的不正确描述是（）

- A 静态成员不属于对象，是类的共享成员
- B c++11 之前，非 `const` 的静态数据成员要在类外定义和初始化
- C 静态成员函数不拥有 `this` 指针，需要通过类参数访问对象成员
- D 只有静态成员函数可以操作静态数据成员

272.C++中空类默认产生哪些类成员函数（）

- A 默认构造函数
- B 析构函数
- C 拷贝构造函数
- D 赋值函数

273.下面关于垃圾收集的说法正确的是

- A 一旦一个对象成为垃圾，就立刻被收集掉。
- B 对象空间被收集掉之后，会执行该对象的 `finalize` 方法
- C `finalize` 方法和 C++的析构函数是完全一回事
- D 一个对象成为垃圾是因为不再有引用指着它，但是线程并非如此

274.关于 `delete` 运算符的下列描述中，（ ）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

275.若 `class B` 中定义了一个 `class A` 的类成员 `A a`，则关于类成员的正确描述是（ ）。

- A 在类 `B` 的成员函数中可以访问 `A` 类的私有数据成员
- B 在类 `B` 的成员函数中可以访问 `A` 类的保护数据成员
- C 类 `B` 的构造函数可以调用类 `A` 的构造函数做数据成员初始化
- D 类 `A` 的构造函数可以调用类 `B` 的构造函数做数据成员初始化

276.有以下类定义

```
#include<iostream>

using namespace std;
```

```

class Clock

{public:

Clock(int NewH=0, int NewM=0, int NewS=0)

{Hour=NewH; Minute=NewM;Second=NewS;}

void ShowTime()

{cout<<Hour<<":"<<Minute<<":"<<Second;}

Clock  operator ++();

private:

int Hour, Minute, Second;};

Clock Clock::operator ++()

{ Second++;

if(Second>=60)

{   Second=Second-60;

Minute++;

if(Minute>=60)

{

Minute=Minute-60;

Hour++;

Hour=Hour%24;}}

return _____;}

int main()

{   Clock c1;

(++c1).ShowTime();

return 0;}

```

请为横线处选择合适的程序使得程序的运行结果是 0:0:1 （ ）

- | | |
|------------|------------|
| A c1 | B *this |
| C Clock | D this |

277.调用以下函数，将返回

```

public static int func()
{
    try
    {
        return 1;
    }
    catch (Exception e)
    {
        return 2;
    } finally
    {
        return 3;
    }
}

```

A 1 B 2 C 3

278.在 c++中的结构体是否可以有构造函数？

A 不可以，结构类型不支持成员函数

B 可以有

C 不可以，只有类允许有构造函数

279.下面哪种面向对象的方法可以让你变得富有()

A 继承

B 封装

C 多态

D 抽象

280.在 C++中，定义结构体类型 struct 时不能有成员函数，只有 class 可以有成员函数。

A 正确

B 错误

281.设已经有 A,B,C,D4 个类的定义，程序中 A,B,C,D 析构函数调用顺序为？

C c;

```
void main()
{
    A*pa=new A();
    B b;
    static D d;
    delete pa;
}
```

A. A B C D
C. A C D B

B. A B D C
D. A C B D

