

- 1、在定义类对象的语句执行时，系统在建立每个对象的过程中将自动调用该类的()使其初始化。
- 2、当一个类对象被撤消时将自动调用该类的()。
- 3、对基类数据成员的初始化是通过执行派生类构造函数中的()来实现的。
- 4、对一个类中的数据成员的初始化可以通过构造函数中的()实现，也可以通过构造函数中的()实现。
- 5、在一个派生类中，对基类成员、类对象成员和非类对象成员的初始化次序是先()，后()，最后为()。
- 6、当撤消一个含有基类和类对象成员的派生类对象时，将首先完成()的析构函数定义体的执行，接着完成()的析构函数定义体的执行，最后完成()的析构函数定义体的执行。
- 7、设 `px` 是指向一个类动态对象的指针变量，则执行 “`delete px;`”语句时，将自动调用该类的()。
- 8、当一个类对象离开它的作用域时，系统将自动调用该类的()。
- 9、假定一个类对象数组为 `a[n]`，当离开它的作用域时，系统自动调用该类析构函数的次数为()。
- 10、假定 `AB` 为一个类，则执行 “`AB a[10];`” 语句时，系统自动调用该类构造函数的次数为()。
- 11、假定用户没有给一个名为 `AB` 的类定义构造函数，则系统为其隐含定义的构造函数为()。
- 12、假定用户没有给一个名为 `AB` 的类定义析构函数，则系统为其隐含定义的析构函数为()。
- 13、若需要把一个函数 “`void f();`” 定义为一个类 `AB` 的友元函数，则应在类 `AB` 的定义中加入一条语句：()
- 14、若要把一个类 `AB` 定义为一个类 `CD` 的友元类，则应在类 `CD` 的定义中加入一条语句：()
- 15、假定类 `AB` 中有一个静态整形成员 `bb`，在类外为它进行定义初始化为 0 时，所使用的语句为 ()。
- 16、假定类 `AB` 中有一个公用属性的静态数据成员 `bb`，在类外不通过对象名访问该成员 `bb` 的写法为 ()
- 17、当类中一个字符指针成员指向具有 `n` 个字节的存储空间时，它所能存储字符串的最大长度为 ()
- 18、假定 `AB` 这一个类，则该类的拷贝构造函数的声明语句为 ()。
- 19、对类对象成员的初始化是通过执行构造函数中的 () 完成的。
- 20、对于类中定义的成员，其隐含访问的权限为 ()，对于结构中定义的成员，其隐含访问权限为 ()。
- 21、一个类的友元函数或友元类能够通过成员操作符访问该类的 ()。
- 22、假定要对类 `AB` 定义加号操作符重载成员函数，实现两个 `AB` 类对象的加法，并返回相加结果，则该成员函数的声明语句为：()。
- 23、在 C++ 流类库中，根基类为 ()。
- 24、在 C++ 流类库中，输入流类和输出流类的名称分别为 () 和 ()。
- 25、若要在程序文件中进行标准输入输出操作，则必须在开始的 `#include` 命令中使用 () 头文件。
- 26、若要在程序文件中进行文件输入输出操作，则必须在开始的 `#include` 命令中使用

() 头文件。

27、当从字符文件中读取回车和换行两个字符时，被系统看作为一个 ()。

28、当使用 `ifstream` 流类定义一个流对象并打开一个磁盘文件时，文件的隐含打开方式为 (),当使用 `ofstream` 流类定义一个流对象并打开一个磁盘文件时，文件的隐含打开方式为 ()。

29、当需要使用 `istream` 流类定义一个流对象并联系一个字符串时，应在文件开始使用 `#include` 命令，使之包含 ()。

1.

```
#include <iostream>
using namespace std;
```

```
//question 1
class A
{
    public:
        A() {a=b=0;}
        A( int aa, int bb){
            a = aa;
            b = bb;
            cout<<a<<" "<<b<<endl;
        }
    private:
        int a, b;
};
```

```
int main()
{
    A x;
    A y(6,3);
    A z(8,10);
    system("pause");
    return 0;
}
```

```
// question 2
```

```
class A
{
    private:
```

```

        int a;
        int b;
    public:
        A(int aa=0, int bb=0):a(aa), b(bb){
            cout<<"constructor"<<a+b<<endl;
        }
};

int main()
{
    A x;
    A y(2,5);
    A z(y);
    system("pause");
    return 0;
}

```

//question 3

```

class A
{
    public:
        A(int aa=0)
        {
            a = new int(aa);
            cout<<"constructor!"<<*a<<endl;
        }
    private:
        int *a;
};

```

```

int main()
{
    A x[2];
    A *p = new A(5);
    delete p;
    system("pause");
    return 0;
}

```

//question 4

```

class A
{
    public:

```

```

        A(int aa=0):a(aa){}
        ~A()
        {
            cout<<"destructor!"<<a<<endl;
        }
    private:
        int a;
};

```

```

int main()
{
    A x(5);
    A *p = new A(10);
    delete p;
    system("pause");
    return 0;
}

```

//question 5

```

class A
{
    public:
        A(int x)
        {
            a = new int(x);
            cout<<"constructor!"<<*a<<endl;
        }
        ~A()
        {
            delete a;
            cout<<"destructor!"<<endl;
        }
    private:
        int *a;
};

```

```

int main()
{
    A x(9), *p;
    p = new A(12);
    delete p;
    system("pause");
    return 0;
}

```

```
//question 6
class A
{
    public:
        A(int aa = 0):a(aa)
        {
            cout<<"constructor A!"<<a<<endl;
        }
    private:
        int a;
};

class B:public A
{
    public:
        B(int aa, int bb):A(aa), b(bb)
        {
            cout<<"constructor B!"<<b<<endl;
        }
    private:
        int b;
};

int main()
{
    B x(2,3);
    B y(4,5);
    system("pause");
    return 0;
}
```

```
//question 7
class A
{
    public:
        A(int aa = 0):a(aa) {a = aa;}
        ~A(){cout<<"deconstructor A!"<<a<<endl;
    private:
        int a;
};

class B:public A
{
    public:
        B(int aa=0, int bb = 0):A(aa){b=bb;}
}
```

```

        ~B(){cout<<"destructor B!"<<b<<endl;}
    private:
        int b;
};

int main()
{
    B x(5);
    B y(6,7);
    system("pause");
    return 0;
}

//question 8
class A
{
    public:
        A(int aa, int bb, char ch)
        {
            a = aa;
            b = bb;
            op = ch;
        }
        int Comp()
        {
            switch(op)
            {
                case '+': return a+b;
                case '-': return a-b;
                case '*': return a*b;
                case '/':
                    if(b!=0)
                        return a/b;
                    else
                        exit(1);
                case '%':
                    if(b!=0)
                        return a%b;
                    else
                        exit(1);
                default: exit(1);
            }
        }
        void SetA(int aa, int bb, char ch)

```

```

        {
            a = aa;
            b = bb;
            op = ch;
        }
private:
    int a, b;
    char op;
};

```

```

int main()
{
    A x(3, 5, '*');
    int a = x.Comp();
    x.SetA(4, 9, '+');
    a+= x.Comp();
    x.SetA(13,8,'%');
    a+=x.Comp();
    cout<<"a="<<a<<endl;
}

```

//question 9 *copy

```

class A
{
    public:
        A(){a=b=0;}
        A(int aa,int bb)
        { a=aa; b=bb; }
        A operator+(A &x)
        {
            A r;
            r.a=a+x.a;
            r.b=b+x.b;
            return r;
        }
        A operator-(A &x)
        {
            A r;
            r.a=a-x.a;
            r.b=b-x.b;
            return r;
        }
        void OutA(){ cout<<a<<" "<<b<<endl;}
    private:

```

```
        int a,b;
    };
```

```
int main( )
{
    A x(6,5),y(13,3),z1,z2;
    z1=x+y;
    z2=x-y;
    z1.OutA();
    z2.OutA();
    system("pause");
    return 0;
}
```

```
//question 10  *copy
template <class TT>
class FF
{
    public:
        FF(TT b1, TT b2, TT b3){a1 = b1; a2 = b2; a3 = b3;}
        TT Sum(){ return a1+a2+a3;}
    private:
        TT a1, a2, a3;
};

int main()
{
    FF<int> x(8,3,4), y(5,9,11);
    cout<<x.Sum()<<" "<<y.Sum()<<endl;
    system("pause");
    return 0;
}
```