1~5:B,C,C,B,D

6~10:A,B,C,D,D

11~15:C,C,D,A,D

16:->.

17:this 指针

18:友元

19:初始化列表

20:delete []pa;

21:不是

22:纯虚函数，一个或多个纯虚函数，抽象类

23:先构造的后析构，后构造的先析构

24:重载多态，参数多态，包含多态

25:不可以，可以

26:虚基类，作用域分辨符

27:public，protected，private，::

28:在程序中自行抛出异常，测试系统稳定性

29:1，当函数的参数是类的对象时；2，当函数的返回值是类的对象时；3，当一个对象需要另一个对象初始化时。

30:编译多态性也称为静态多态性，它是通过函数重载和运算符重载实现的；

运行多态性也称为动态多态性，它是通过继承和虚函数来实现的。

31:优点:可读性高，由于有继承，在维护的时候只是在局部，更方便。有封装性。有继承，封装，多态的特性

缺点:虽然功能丰富，但同时也更复杂，搞不清楚

32:输出结果：20

20

30

20

33:输出结果：Instance 1 is constructed.

Instance 1 is constructed.

Instance 2 is constructed.

34:输出结果：i：10

J：12

35:输出结果：

B's constructor called.

B's default constructor called.

C's constructor called.

5

6

C's destructor called.

B's destructor called.

B's destructor called.

36:错误1：类B中不存在默认构造函数

错误2：初始化列表有问题

37:[1]virtual;[2]3.14\*x\*x.

38: class Complex{

double real;

double image;

public:

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

Complex operator+( const Complex& cc )const

{

return Complex(real+cc.real,image+cc.image);

}

Complex operator-( const Complex& cc )const

{

return Complex(real-cc.real,image-cc.image);

}

Complex operator\*( const Complex& cc )const

{

double r=real\*cc.real-image\*cc.image;

double i=image\*cc.real+real\*cc.image;

return Complex(r,i);

}

friend ostream& operator

};

39:

LineString::LineString(Point\*pnts,int num):m\_num(num){

for(int i=0;i<m\_num;i++){

m\_data[i].x=pnts[i].x;

m\_data[i].y=pnts[i].y;

}

}

LineString::Point&operator[](int index){

Return m\_data[index];

}