三、**改错题**

1、下列程序中有三个错误，请改正错误（注意不要修改主函数），使程序的输出结果为：

The point is (0,1)

The point is (3,5)

源文件清单如下：

Line1: #include <iostream.h>

Line2: class Point {

Line3: public:

Line4: Point(int xx=0, int yy) : x(xx), y(yy) { }

Line5: void Move(int xOff, int yOff) const{

Line6: x+=xOff; y+=yOff;

Line7: }

Line8: void Print() const

Line9: { cout<<"The point is ("<<x<<', '<<y<<')'<<endl; }

Line10: private:

Line11: int x,y=0;

Line12: };

Line13: int main() {

Line14: Point p1,p2(2,1);

Line15: p1.Print();

Line16: p2.Move(1,4);

Line17: p2.Print();

Line18: return 0;

Line19: }

Line4: **Point(int xx=0, int yy=1) : x(xx), y(yy) { }**

Line5: **void Move(int xOff, int yOff){**

Line11: **int x,y;**

2、**下列程序中有三个错误，请改正错误（注意不要修改主函数），使程序的输出结果为：**

Kelly is 10 years old.

Patrick is 35 years old.

源文件清单如下：

Line1: #include <iostream.h>

Line2: #include <string.h>

Line3: class Person {

Line4: public:

Line5: Person(char \*str, int \_age=10) : age(\_age)

Line6: { name=new char[strlen(str+1)]; strcpy(name,str); }

Line7: void ~Person() { delete[ ] name; }

Line8: void SetAge(int num){ age=num; }

Line9: void Print() const { cout<<name<<" is "<<age<<" years

old. "<<endl; }

Line10: private:

Line11: char \*name;

Line12: int age=10;

Line13: };

Line14: int main() {

Line15: Person p1("Kelly"),p2("Patrick",25);

Line16: p1.Print();

Line17: p2.SetAge(35);

Line18: p2.Print();

Line19: return 0;

Line20: }

Line5:**：Person(char \*str,int \_age=10) : age(\_age).**

Line7:**：~Person() { delete[ ] name; }**

Line12**：int age;**

3、下列程序中有三个错误，请改正错误（注意不要修改主函数），使程序的输出结果为：

Constructor

The value is 10

Destructor

源文件清单如下：

Line1: #include <iostream.h>

Line2: class MyClass

Line3: {

Line4: public:

Line5: MyClass(int x):value(x) { cout<<"Constructor"<<endl; }

Line6: void ~MyClass() { cout<<"Destructor"<<endl; }

Line7: void Print() const;

Line8: private:

Line9: int value=0;

Line10: };

Line11: void MyClass::Print()

Line12 {

Line13 cout<<"The value is "<<value<<endl;

Line14: }

Line15: int main()

Line16: {

Line17: const MyClass object(10);

Line18: object.Print();

Line19: return 0;

Line20: }

Line6: **~MyClass() { cout<<"Destructor"<<endl; }**

Line9: **int value;**

Line11: **void MyClass::Print() const**

4、下列程序中有三个错误，请改正错误，使程序的输出结果为：

Now we have 2 points.

The point 1 is (0,0)

The point 2 is (10,10)

源文件清单如下：

Line1: #include<iostream.h>

Line2: class Point{

Line3: public:

Line4: void Point(int\_x=0,int\_y=0)

:x(\_x),y(\_y){++count;}

Line5: void print() const

{cout<<”(”<<x<<”,”<<y<<”)”;}

Line6. Static int getNumber(){return count;}

Line7. Private:

Line8. static int count;

Line9. int x,y;

Line10. };

Line11. satic int Point::count=0；

Line12. int main(){

Line13. Point p1(0,0),p2(10,10);

Line14. cout<<”Now we have”<<getNumber()

<<”points.”<<endl;

答案：

Line4: Point(int\_x=0,int\_y=0)

:x(\_x),y(\_y){++count;}

Line11. int Point::count=0；

Line14. cout<<”Now we have”<<Point::getNumber()

<<”points.”<<endl;

5、下列程序中有三个错误，请改正错误，使程序的输出结果为：

Now we have 2 points.

The point 1 is (0,0)

The point 2 is (10,10)

源文件清单如下：

Line1: #include<iostream.h>

Line2: class Point{

Line3: public:

Line4: Point(int \_x=0,int \_y=0):x(\_x),y(\_y){++count;}

Line5: void print() const { cout<<"("<<x<<","<<y<<")";}

Line6: static int getNumber(){return count;}

Line7: private:

Line8: static int count;

Line9: int x,y;

Line10: };

Line11: int Point::count=0;

Line12: int main(){

Line13: Point p1(0,0),p2(10,10);

Line14: cout<<"Now we have "<< getNumber()<<" points. "<<endl;

Line15: cout<<"The point 1 is";

Line16: p1.print();

Line17: cout<<endl<<"The point 2 is ";

Line18: p2.print();

Line19: return 0;

Line20: }

Line4: **void Point(int \_x=0,int \_y=0):x(\_x),y(\_y){++count;}**

Line11: **static int Point::count=0;**

Line14: **cout<<"Now we have "<< Point::getNumber()<<" points. "<<endl;**

**四、阅读程序题**

1、请写出下列程序的输出结果。

#include <iostream.h>

class Test{

public:

Test(int x=0):val(x) { cout<<"Cons "<<val<<endl; }

Test(const Test &p) { val=p.val; cout<<"Copy "<<val<<endl; }

friend Test operator + (Test &t1,Test &t2);

private:

int val;

};

Test operator + (Test &t1,Test &t2) {

Test temp(t1.val+t2.val);

return temp;

}

int main() {

Test A(1),B(3);

Test C;

C=A+B;

return 0;

}

结果：

Cons 1

Cons 3

Cons 0

Cons 4

Copy 4

2、请写出下列程序的输出结果 。

#include <iostream.h>

class A {

public:

A(int i):r1(i) { cout<<r1<<endl; }

~A()　　　　 { cout<<'~'<<r1<<endl; }

void print() {cout<<"Empty:"<<r1<<endl;}

void print() const {cout<<"Const:"<<r1<<endl;}

void print(int x) {cout<<"Param:"<<x\*x<<endl;}

private:

int r1;

};

int main() {

A a1(1),a2(2);

a1.print();

a2.print(3);

return 0;

}

结果：

1

2

Empty:1

Param:9

~2

~1

3、请写出下列程序的输出结果。

#include <iostream.h>

class Point{

public:

Point() { number++; }

~Point() { number--; cout<<number<<endl; }

static int GetNumber() { return number; }

private:

static int number;

};

int Point::number=0;

int main() {

Point A,B;

Point \*ptr=new Point[3];

cout<<Point::GetNumber()<<endl;

delete[] ptr;

return 0;

}

结果：

5

4

3

2

1

0

4、请写出下列程序的输出结果。

#include <iostream.h>

class MyClass {

public:

MyClass(int x):val(x) { cout<<val<<endl; }

~MyClass() { cout<<'~'<<val<<endl; }

int Add() { val=2\*val; return val; }

int Add(int x) { val=val+x; return val; }

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

private:

int val;

};

int main() {

MyClass obj1(10),obj2(30);

obj1.Add();

obj2.Add(10);

obj1.Print();

obj2.Print();

return 0;

}

结果：

10

30

20

40

~40

~20

8、请写出下列程序的输出结果

#include<iostream.h>

Class Base{

public:

virtual void Show() {cout<<”Base”<<endl;}

};

class Derived: public Base {

public:

virtual void Show() { cout<<”Derived”<<endl;}

};

void Print(Base&obj) {obj.Show();}

int main() {

Base b;

Derived d;

cout<<”Section 1:”; Print(b);

cout<<”Section 2:”; Print(d);

return 0;

}

结果： Section 1: Base Section2: Derived

5、请写出下列程序的输出结果。

#include <iostream.h>

class A {

public:

A(int i):r1(i) { cout<<r1<<endl; }

~A()　　　　 { cout<<'~'<<r1<<endl; }

void print() {cout<<"Empty:"<<r1<<endl;}

void print() const {cout<<"Const:"<<r1<<endl;}

void print(int x) {cout<<"Param:"<<x\*x<<endl;}

private:

int r1;

};

int main() {

A a1(1);

const A a2(2);

a1.print();

a2.print();

return 0;

}

结果：

1

2

Empty:1

Const:2

~2

~1

6、请写出下列程序的输出结果。

#include <iostream.h>

class Point{

public:

Point() { number++; cout<<number<<endl; }

~Point() { number--; }

static int GetNumber() { return number; }

private:

static int number;

};

int Point::number=0;

int main() {

Point A,B;

Point \*ptr=new Point[3];

delete[] ptr;

cout<<Point::GetNumber()<<endl;

return 0;

}

结果：

1

2

3

4

5

2

7、请写出下列程序的输出结果

#include<iostream.h>

class A{

public:

A(int i):rl(i) {cout<<rl<<endl;}

~A() {cout<<’~’<<rl<<endl;}

void print() {cout<<”Empty:”<<rl<<endl;}

void print() const {cout<<”Const:”<<rl<<endl;}

void print(int x) {cout<<”Param:”<<x\*x<<endl;}

private:

int rl;

};

int main() {

A al (1);

const A a2(2);

a1.print(3);

a2.print();

return();

}

结果：1

2

Param:9

Const:2

~2

~1

9、 请写出下列程序的输出结果。

#include <iostream.h>

class Base1 {

public:

Base1() { cout<<"Base1"<<endl; }

~Base1() { cout<<"~Base1"<<endl; }

};

class Base2 {

public:

Base2() { cout<<"Base2"<<endl; }

~Base2() { cout<<"~Base2"<<endl; }

};

class Derived : public Base1, public Base2 {

public:

Derived() { cout<<"Derived "<<endl; }

~Derived() { cout<<"~Derived"<<endl; }

private:

Base1 b1;

Base2 b2;

};

int main() {

Derived d;

return 0;

}

结果：

Base1

Base2

Base1

Base2

Derived

~Derived

~Base2

~Base1

~Base2

~Base1

10、请写出下列程序的输出结果。

#include <iostream.h>

class Base{

public:

Base() { cout<<"Base"<<endl; }

~Base() { cout<<"~Base"<<endl; }

};

class Base1 : virtual public Base{

public:

Base1() { cout<<"Base1"<<endl; }

~Base1() { cout<<"~Base1"<<endl; }

};

class Base2 : virtual public Base{

public:

Base2() { cout<<"Base2"<<endl; }

~Base2() { cout<<"~Base2"<<endl; }

};

class Derived : public Base1, public Base2 {

public:

Derived() { cout<<"Derived "<<endl; }

~Derived() { cout<<"~Derived"<<endl; }

private:

Base b;

};

int main() {

Derived d;

return 0;

}

结果：

Base

Base1

Base2

Base

Derived

~Derived

~Base

~Base2

~Base1

~Base

**11、请写出下列程序的输出结果**

#include <iostream.h>

class TV{

public:

TV(int s=41):size(s) {cout<<'/'<<"Cons";}

~TV() {cout<<'/'<<"Des"<<'/'<<size;}

void Print() { cout<<'/'<<'N'<<'/'<<size;}

void Print() const {cout<<'/'<<'C'<<'/'<<size;}

void Print(int s) { cout<<'/'<<'S'<<'/'<<size;}

private:

int size;

};

int main(){

TV room1;

const TV room2(54);

room1.Print();

room2.Print();

return 0;

}

结果：/Cons/Cons/N/41/C/54/Des/54/Des/41

**12、请写出下列程序的输出结果**

#include <iostream.h>

class TV{

public:

TV(int s=41):size(s) {cout<<'/'<<"Cons";}

~TV() {cout<<'/'<<"Des"<<'/'<<size;}

void Print() {cout<<'/'<<'N'<<'/'<<size;}

void Print() const {cout<<'/'<<'C'<<'/'<<size;}

void Print(int s) { cout<<'/'<<'S'<<'/'<<size;}

private:

int size;

};

int main(){

TV room1,room2;

room1.Print();

room2.Print(54);

return 0;

}

结果：/Cons/Cons/N/41/S/41/Des/41/Des/41

**13、请写出下列程序的输出结果。**

#include <iostream.h>

class One {

public:

One() { cout<<"One"<<endl; }

~One() { cout<<"~One"<<endl; }

};

class Two {

public:

Two() { cout<<"Two"<<endl; }

~Two() { cout<<"~Two"<<endl; }

};

class Base1 {

public:

Base1() { cout<<"Base1"<<endl; }

~Base1() { cout<<"~Base1"<<endl; }

};

class Base2 {

public:

Base2() { cout<<"Base2"<<endl; }

~Base2() { cout<<"~Base2"<<endl; }

};

class Derived : public Base2, public Base1 {

public:

Derived() { cout<<"Derived "<<endl; }

~Derived() { cout<<"~Derived"<<endl; }

private:

One o1;

Two t1;

};

int main() {

Derived obj;

return 0;

}

结果：

Base2

Base1

One

Two

Derived

~Derived

~Two

~One

~Base1

~Base2

**14、 请写出下列程序的输出结果。**

#include <iostream.h>

class One {

public:

One() { cout<<"One"<<endl; }

~One() { cout<<"~One"<<endl; }

};

class Two {

public:

Two() { cout<<"Two"<<endl; }

~Two() { cout<<"~Two"<<endl; }

};

class Base1 {

public:

Base1() { cout<<"Base1"<<endl; }

~Base1() { cout<<"~Base1"<<endl; }

};

class Base2 {

public:

Base2() { cout<<"Base2"<<endl; }

~Base2() { cout<<"~Base2"<<endl; }

};

class Derived : public Base1, public Base2 {

public:

Derived() { cout<<"Derived "<<endl; }

~Derived() { cout<<"~Derived"<<endl; }

private:

One o1;

Two t1;

};

int main() {

Derived obj;

return 0;

}

结果：

Base1

Base2

One

Two

Derived

~Derived

~Two

~One

~Base2

~Base1

**15、请写出下列程序的输出结果。**

#include <iostream.h>

class Base {

public:

void f() { cout<<"fB"<<endl; }

virtual void g() { cout<<"gB"<<endl; }

};

class Derived : public Base {

public:

void f() { cout<<"fD"<<endl; }

virtual void g() { cout<<"gD"<<endl; }

};

int main() {

Derived d;

Base \*p=&d;

Cout<<”Section 1:” ; p->f();

Cout<<”Section 2:” ; p->g();;

return 0;

}

结果：

Section 1:**fB**

Section 2:**gD**

**16、请写出下列程序的输出结果。**

#include <iostream.h>

class Base {

public:

virtual void f() { cout<<"fB"<<endl; }

virtual g() { cout<<"gB"<<endl; }

};

class Derived : public Base {

public:

virtual void f() { cout<<"fD"<<endl; }

void g() { cout<<"gD"<<endl; }

};

int main() {

Derived d;

Base \*p=&d;

Cout<<”Section 1:”; p->f();

Cout<<”Section 2:”; p->g();;

return 0;

}

结果：

Section 1:fD

Section 2:gB