**二、关于友元专题**

1. 选择与简答题

1、什么是友元机制？友元有哪些分类？

2、下面友员的错误描述是（ ）【对错误项进行描述分析】

A 关键字friend用于声明友员

B 一个类中的成员函数可以是另一个类的友员

C 友员函数访问对象的成员不受访问特性影响

D 友员函数通过this指针访问对象成员

（二）程序填空题

1. 分析并填充程序

#include <iostream>

using namespace std;

class Time

{ public:

Time(int,int,int);

void prin();

private:

int hour;

int minute;

int sec;

};

Time::Time(int h,int m,int s)

{ hour=h;

minute=m;

sec=s;

}

Time::prin() { cout<<hour<<minute<<sec; }

void display( )

{ cout<<t.hour<<":"<<t.minute<<":"<<t.sec<<endl;} //直接输出对象的hour、minute和sec

int main()

{ Time t1(10,13,56);

display( );

t1.prin(); //成员函数的调用

return 0;

}

（1）请分析：display()和prin()函数的区别

（2）请分析：display()函数的参数和函数体的书写特点

2、在没有描述运算符重载之前，实现相关操作，比如加的操作，是通过定义成员函数或友元来实现。请完善代码

#include <iostream>

using namespace std;

class Complex

{ private:

double real;

double im;

public:

Complex(double r, double i) { real=r;

im=i;

}

Complex(Complex &t) { real=t.real;

im=t.im;

cout<<"HAHA ";

}

**\*\*\* Add(\*\*\*\*\*\*\*\*) //要求完善代码，实现复数的加**

**{ \*\*\*\*\*\***

**}**

**void display() { cout<<real<<":"<<im; }**

};

**\*\*\*Add2(\*\*\*\*\*\*) //要求完善代码，实现复数的加**

**{ \*\*\*\*\*\***

**}**

int main() {

Complex p1(11,12),p2(13,14);

Complex c1,c2;

c1=p1.Add(p2);

c2=Add2(p1,p2);

c1.display();

c2.display();

return 1;

}

（1）根据add和add2位置特性，完善代码；

（2）在main中添加语句：c2= p1+p2; 程序的执行情况反馈

（3）对main函数中下列语句，重点分析执行过程

c1=p1.Add(p2);

c2=Add2(p1,p2);

（4）结合（2）和（3）可以看出，上述代码在功能上实现了两个复数p1和p2的“加”，但是无法满足用户对“p1+p2”表述书写的需求，该如何做？

（5）请描述程序中复制构造函数调用了几次？分别是在什么时候调用的？

（三）改错题

1. #include <iostream>
2. class Animal;
3. void setValue(Animal&,int);
4. void setValue(Animal&,int,int);
5. class Animal {
   1. pulbic:
   2. friend void Setvalue(Animal &,int);
   3. protected:
   4. int itsWeight;
   5. int itsAge;
6. };
7. void Setvalue(Animal& ta,int tw) {
   1. ta.itsWeight=tw;
8. }
9. void SetValue(Animal& ta,int tw,int tn) {
   1. ta.itsWeight=tw;
   2. ta.itsAge=tn;
10. }
11. int main() {
    1. Animal peppy;
    2. SetValue(peppy,5);
    3. SetValue(peppy,7,9）;
    4. return 0;
12. }

要求：

1. 对错误的代码做修改并标注
2. 取消程序中的友元，采用普通函数完成函数功能【通过增加访问类中保护数据的const成员函数】

（四）代码题：计算二维坐标系中，两点之间的距离。

类的声明：

class Point {

double x,y;

public:

Point(double xx,double yy): //参数初始化列表

friend double Distance( ); //计算两点距离的普通函数

double NewDistance( ); //计算两点距离的成员函数

};

(1) 完善类的声明

(2) 在类外定义计算两点之间距离的成员函数、计算两点之间距离的普通函数

(3) 完善主函数，比较Distance和NewDistance的应用上的区别

(4) 思考：由于将Distance描述成友元，那么它可以访问对象的私有成员，实现两点间距离的计算。现在若要求在不引入友员前提下，仍实现两点之间距离的计算，该如何实现？（思路：可增加public的成员函数，成员函数来访问私有成员；普通函数访问public的成员函数）

三、**静态成员专题**

（一）简答与选择题

1、静态成员的分类？

2、静态数据成员有什么特性？如何声明、定义和初始化、使用？

3、静态成员函数有什么特性？如何声明、定义和调用？

4、下面对静态数据成员的描述中，正确的选项（ ）【对错误的选项进行分析】

A 静态数据成员可以在类体内进行初始化

B 静态数据成员不可以被类的对象调用

C 静态数据成员年不受private控制符作用

D 静态数据成员可以直接用类名调用

（二）程序阅读题：分析程序的运行结果，思考静态成员的使用

|  |  |
| --- | --- |
| 1、画出y空间的结构示意图，分析结果  #include <iostream>  class Sample {  private:  int x;  static int y; //语句1  public:  Sample(int a) {  x=a;  x++;  y++; }  void print()  { cout<<”x=”<<x<<”,y=”<<endl; }  };  **int Sample::y=10;**  void main() {  Sample s1(20);  Sample s2(30);  s1.print(); s2.print();  }  2、在main中，直接添加： cout<<Sample::y; 请问程序的运行结果如何？并解释原因。  3、若将语句1调整至public:下，main函数中只保留并执行这语句：cout<<Sample::y; 请问程序的运行结果如何？并解释原因。 | 1、分析static成员函数，找出错误分析原因；  2、用文字总结static成员函数特性  #include <iostream>  class Sample {  private:  int x;  static int y;  public:  Sample(int a) {  x=a;  y+=x; }  static void print(Sample s) {  cout<<”x=”<<x<<”,y=”<<y<<endl;  }  };  int Sample::y=10;  void main() {  Sample s1(20);  Sample s2(30);  s1.print(s1); s2.print(s2);  } |

3、代码题

某公司由若干个子公司组成，Budget类用来计算公司的预算。该类包含一个静态的数据成员CorpBudget，用来存储整个公司的预算额。当调用函数成员addBudget时，将参数增加到CorpBudget中。程序结束时，CorpBudeget的值将是整个公司的预算额。

根据描述，尽可能完善代码。

思路：每个子公司都是一个对象。static的数据成员往往用于统计功能。