



面向对象C++ 1-4章习题课


2018.3

练习1:

- 编写一个C++风格的程序，输入两个整数，将它们由小到大的顺序输出。要求使用变量的引用。


问题：change函数如何写？

- `#include<iostream>`
- `using namespace std;`
- `int main()`
- `{void change(int &,int &);`
- `int a,b;`
- `cin>>a>>b;`
- `if (a>b) change(a,b);`
- `cout<<a<<" "<<b<<endl;`
- `return 0;`
- `}`

- 
-
- `void change(int &a1,int &b1)`
 - `{int temp;`
 - `temp=a1;`
 - `a1=b1;`
 - `b1=temp;`
 - `}`
 - 运行结果:
 - 56 23 ↙
 - 23 56


练习2:

- 写出以下程序的运行结果
- `#include<iostream.h>`
- `class A{`
- `public:`
- `void set (int i,int j) {x=i;y=j;}`
- `int get_y(){return y;}`
- `private:`
- `int x,y;`
- `};`



- `class box{`
- `public:`
- `void set(int l,int w,int s,int p){`
- `length=l;`
- `width=w;`
- `label.set(s,p);`
- `}`
- `int get_area(){`
- `return length*width;`
- `}`
- `private:`
- `int length,width;`
- `A label;`
- `};`

本题练习对象成员的应用。类Box的数据成员包括了类A的对象lable, 用于标识box对象在坐标上的位置。

- 
-
- `int main()`
 - `{`
 - `box b;`
 - `b.set(4,6,1,20);`
 - `cout<<b.get_area()<<endl;`
 - `return 0;`
 - `}`
 - 运行结果:
 - 24

练习3： 请几位同学上台设计函数

- 根据注释语句的提示，实现类**Date**的成员函数。

- `#include<iostream.h>`

- `class Date{`

- `public:`

- `void printDate();` `//显示日期`

- `void setDay(int d);` `//设置日的值`

- `void setMonth(int m);` `//设置月的值`

- `void setYear(int y);` `//设置年的值`

- `private:`

- `int day,month,year;`

- `};`


```
int main()
{
    Date testDay;
    testDay.setDay(5);
    testDay.setMonth(10);
    testDay.setYear(2003);
    testDay.printDate();
    return 0;
}
```



Date的成员函数?

- `void Date::setDay(int d)`
- `{`
- `day=d;`
- `}`
- `void Date::setMonth(int m)`
- `{`
- `month=m;`
- `}`
- `void Date::setYear(int y)`
- `{`
- `year=y;`
- `}`
- `void Date::printDate()`
- `{`
- `cout<<"Today is`
- `"<<year<<'. '<<month<<". "<<day<<endl;`
- `}`

练习4:

- 编写一个程序,已有若干学生的数据,包括学号,姓名,成绩,要求输入这些学生的数据,并计算出学生人数和平均成绩。
(要求将学生的人数和总成绩用静态数据成员表示)

- 
-
- `#include <iostream>`
 - `using namespace std;`
 - `class Student`
 - `{`
 - `public:`
 - `Student(char na[], char stuNo[], double sc);`
 - `void show();`
 - `static void showTotal();`
 - `private:`
 - `char name[25], studentNo[10];`
 - `double score;`
 - `static int num;//总人数`
 - `static double totalScore;//总成绩`
 - `};`



- Student::Student(char na[], char stuNo[], double sc)
- {
- strcpy(name, na);
- strcpy(studentNo, stuNo);
- score = sc;
- ++num;
- totalScore += sc;
- }

- void Student::show()
- {
- cout << "姓名: " << name << endl;
- cout << "学号: " << studentNo << endl;
- cout << "成绩: " << score << endl;
- }


- void Student::showTotal()
- { cout << "总人数: " << num << endl;
- cout << "平均成绩: " << totalScore/num << endl;
- }
- **int Student::num = 0;**//静态数据成员只能在类体外初始化
- **double Student::totalScore = 0;**
- int main()
- { Student s1("张无忌", "111254", 75);
- Student s2("李莫愁", "254114", 60);
- Student s3("小龙女", "112587", 88);
- s1.show();
- s2.show();
- s3.show();
- Student::showTotal();
- return 0;
- }

s1,s2,s3可在
一行中定义,
用“,”隔开


练习5:

- 1.编写一个程序输入3个学生的英语和计算机成绩，并按总分从高到低排序。要求设计一个学生类**Student**，其定义如下：
- `class Student`
- `{`
- `int english,computer,total;`
- `public:`
- `void getscore();` //获取一个学生的成绩
- `void display();` //显示一个学生的成绩
- `void sort(Student a);` //将若干个学生按总分从高到低排序
- `~Student();`
- `};`

-
- `#include <iostream.h>`
 - `class Student`
 - `{`
 - `public:`
 - `Student(){};`
 - `Student(int e,int c,int t):english(e),computer(c),total(t){}`
 - `void getscore();` //获取一个学生成绩
 - `void display();` //显示一个学生成绩
 - `void sort(Student *);` //将若干个学生按总分从高到低排序
 - `~Student();`
 - `private:`
 - `int english;`
 - `int computer;`
 - `int total;`
 - `};`

- 
-
- `void Student::getscore()`
 - `{`
 - `cout<<"输入英语成绩: ";`
 - `cin>>english;`
 - `cout<<"输入计算机成绩: ";`
 - `cin>>computer;`
 - `total=english+computer;`
 - `}`
 - `void Student::display()`
 - `{`
 - `cout<<"英语="<<english<<"计算机`
`= "<<computer<<"总分="<<total<<endl;`
 - `}`


-
- void Student::sort(Student *p)
 - {int tmp,i;
 - for(i=0;i<2;i++)
 - if(total < p->total)
 - {tmp=total;
 - total = p->total;
 - p->total = tmp;
 - tmp=english;
 - english = p->english;
 - p->english = tmp;
 - tmp=computer;
 - computer = p->computer;
 - p->computer =tmp;
 - }
 - }




- int main()
- {Student * A[3];
- for(int j=0;j<3;j++)
- {A[j]=new Student;
- cout<<"学生"<<j+1<<endl;
- A[j]->getscore();
- }
- int i;
- for(j=0;j<2;j++)
- for(i=0;i<2;i++)
- A[i]->sort(A[i+1]);//冒泡法由高分到低分排序
- cout<<endl<<"排序结果如下: "<<endl;
- for(i=0;i<3;i++)
- A[i]->display();
- return 0;
- }

练习6:

- 请为**fraction**类（分数类）定义下列重载运算符函数
- (1)加法运算 $+$ 。
- (2)赋值运算 $=$ 。
- (3)提取运算 $>>$ 。
- **class Fraction**
- {
- **private:**
- **int fz;** //分子
- **int fm;** //分母
- **public:**
- ...
- **};**

- 
-
- `#include <iostream.h>`
 - `class Fraction`
 - `{`
 - `private:`
 - `int fz; //分子`
 - `int fm; //分母`
 - `public:`
 - `Fraction(int nfz,int nfm):fz(nfz),fm(nfm){}`
 - `friend Fraction operator + (Fraction &,Fraction &);`
 - `Fraction & operator =(Fraction &f);`
 - `friend istream & operator >>(istream &,Fraction &);`
 - `friend ostream & operator <<(ostream &,Fraction &);`
 - `};`




- Fraction operator + (Fraction & f1, Fraction & f2)
- { int nfz = f1.fz*f2.fm+f1.fm*f2.fz;
- int nfm = f1.fm*f2.fm;
- return Fraction(nfz,nfm);
- }
- Fraction & Fraction::operator =(Fraction & f)
- { fz=f.fz;
- fm=f.fm;
- return *this;
- }
- istream & operator >>(istream & is, Fraction & f)
- { cout<<"请输入分子: "<<endl;
- is>>f.fz;
- cout<<"请输入分母: "<<endl;
- is>>f.fm;
- return is;
- }

-
- ostream & operator <<(ostream & os, Fraction & f)
 - { os<<f.fz<<"/"<<f.fm;
 - return os;
 - }
 - int main()
 - {
 - Fraction ff1(3,4),ff2(3,8);
 - cin>>ff2;
 - ff1=ff2;
 - cout<<ff1<<" " <<ff2<<endl;
 - return 0;
 - }

练习7:

- 设计并测试类**Point**，其数据成员是直角坐标系的点坐标。友元函数**distance**用来计算两点间的距离。

- 
-
- `#include <iostream.h>`
 - `#include <math.h>` //后面用到 `sqrt`函数
 - `class Point`
 - `{`
 - `private:`
 - `float x, y;`
 - `public:`
 - `Point(float xx, float yy){ x=xx; y=yy; }`
 - `float GetX(){ return x; }`
 - `float GetY(){ return y; }`
 - `friend double distance(Point&, Point&);`
 - `};`

-
- `double distance(Point& a, Point& b)`
 - `{float dx = a.x - b.x;`
 - `float dy = a.y - b.y;`
 - `cout<<" 用友元函数 distance(). 显示两点的坐标 : \n ";`
 - `cout<<a.x<<" , "<<a.y<<" "<<b.x<<" ,`
`"<<b.y<<endl;`
 - `return sqrt(dx*dx+dy*dy);`
 - `}`
 - `int main()`
 - `{`
 - `double d;`
 - `Point p1(10.5, 20.5), p2(2.5, 6.5);`
 - `d=distance(p1, p2); //访问私有成员x,y`
 - `cout<<" 两点的距离 : d = "<<d<<endl;`
 - `return 0;`
 - `}`



完

- 周二上机调试验证本次习题课讲的内容。