**8-10章练习**

1./\*

定义函数fun：根据p指针，计算该学生的4门课的平均分。

学生记录由学号，姓名，四门课成绩，平均分组成。

注意：请勿改动主函数main和其它函数中的任何内容，仅在函数fun的花括号中填入你编写的若干语句。

----------------------------------------------------------------------------------------\*/

#include<iostream>

#include<fstream>

using namespace std;

struct student

{

int id;

char name[20];

int score[4];

int avg;

};

void fun(student\* p)

{

/\*\*\*\*\*\*\*\*\*\*\*\*\*begin \*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int sum=0;

int i;

for(i=0;i<4;i++)

{

sum=sum+p->score[i];

}

p->avg=sum/4;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*end\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

}

int main()

{

student stu;

int i;

ifstream ifile("in.txt");

ofstream ofile("out.txt");

while(!ifile.eof())

{

ifile>>stu.id; //从文件读入学号

ifile>>stu.name; //从文件读入姓名

for(i=0;i<4;i++)

ifile>>stu.score[i]; //从文件读入四门课的成绩

fun(&stu);

cout<<stu.id<<" "<<stu.name<<" ";

for(i=0;i<4;i++)

cout<<stu.score[i]<<" ";

cout<<"平均成绩:"<<stu.avg<<endl;

//输出到out.txt文件

ofile<<stu.id<<" "<<stu.name<<" ";

for(i=0;i<4;i++)

ofile<<stu.score[i]<<" ";

ofile<<"平均成绩:"<<stu.avg<<endl;

}

ifile.close(); //关闭输入文件in.txt

ofile.close(); //关闭输出文件out.txt

return 0;

}

2./\*

定义函数fun：在n个学生记录中，根据学生的学号，查找该学生对应的成绩，若没有该学号，则返回-1。

学生的记录由学号和成绩组成，N名学生的数据已在主函数中放入结构体数组s中，要查找的学生学号从文件读入，依次将学生的学号和查找结果存放在输出文件中。

（用于字符串比较的函数是strcmp）。

例如in.txt中显示：

GA09

GA010

GA011

则out.txt中输出：

GA09 -1

GA010 -1

GA011 91

注意：请勿改动主函数main和其它函数中的任何内容，只能在begin与end之间写代码。

------------------------------------------------------------------------------------------\*/

#include<iostream>

#include<cstring>

#include<fstream>

using namespace std;

#define N 16

struct student

{ char num[10];

int score;

} ;

int fun (student stu[],int n, char num[])

{

/\*\*\*\*\*\*\*\*\*\* Begin \*\*\*\*\*\*\*\*\*\*/

int i=0;

bool flag=false;//找到标志

for(i=0;i<n;i++)

{

if(strcmp(stu[i].num,num)==0) //stu[i].num代表数组stu中每一项的字符 ，num为用户想要保存的学号（字符名）

{

flag=true;

break;

}

}

if(flag==true)

return stu[i].score;

else

return -1;

/\*\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*\*\*/

}

int main()

{ ifstream infile("in.txt",ios::in);

ofstream outfile("out.txt",ios::out);

student s[N]={{"GA05",85},{"GA03",76},{"GA02",69},{"GA04",85},{"GA01",91},{"GA07",72},{"GA08",64},{"GA06",87},{"GA015",85},{"GA013",91},{"GA012",64},{"GA014",91},{"GA011",91},{"GA017",64},{"GA018",64},{"GA016",72}};

int score;

char num[10];//保存要查找学号

while(infile>>num)

{

score=fun(s,N,num);

outfile<<num<<score<<endl;

}

infile.close();

outfile.close();

return 0;

}

3.

/\*

声明一个三角形类Triangle，包括三个私有对象成员point1,point2,point3，这三个成员是点类Point的对象；

类Point包括公有成员x(float),y(float)表示点的坐标，包括构造函数给坐标初始化；

类Triangle包括构造函数，求面积的函数float Area()，包括依次输出三个点point1,point2,point3的坐标的函数Disp()；

程序中的输入文件in.txt包含以下数据：

1 1 2 2 3 3

0 0 0 4 4 4

1 1 4 4 1 4

输出文件out.txt包含以下数据（点的x,y坐标之间以一个空格分隔）：

1 1

2 2

3 3

0

0 0

0 4

4 4

8

1 1

4 4

1 4

4.5

提示：程序使用文件实现输入和输出，不可以用 cin 和 cout ；

只能在Begin和End之间输入代码，不能改动程序的任何其余部分。

---------------------------------------------------------------------------------------------------\*/

#include<fstream>

#include<cmath>

using namespace std;

ifstream infile("in.txt");

ofstream outfile("out.txt");

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Begin \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

class Point

{

public: float x,y;

Point(float x1,float y1) {x=x1;y=y1;}

};

class Triangle

{

private: Point point1,point2,point3;

public:

Triangle(Point &p1,Point &p2,Point &p3):point1(p1),point2(p2),point3(p3){}

float Area()

{

float s1 = sqrt((point1.x-point2.x)\*(point1.x-point2.x)+(point1.y-point2.y)\*(point1.y-point2.y));

float s2 = sqrt((point3.x-point2.x)\*(point3.x-point2.x)+(point3.y-point2.y)\*(point3.y-point2.y));

float s3 = sqrt((point3.x-point1.x)\*(point3.x-point1.x)+(point3.y-point1.y)\*(point3.y-point1.y));

float s = 0.5\*(s1+s2+s3);

return sqrt(s\*(s-s1)\*(s-s2)\*(s-s3));

}

double Disp(ostream & output)

{

output<<point1.x<<' '<<point1.y<<endl;

output<<point2.x<<' '<<point2.y<<endl;

output<<point3.x<<' '<<point3.y<<endl;

}

};

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int main( )

{

if(!infile) return -1;

if(!outfile) return -1;

float x1,y1,x2,y2,x3,y3;

while(!infile.eof())

{

infile>>x1>>y1>>x2>>y2>>x3>>y3;

Point p1(x1,y1),p2(x2,y2),p3(x3,y3);

Triangle t(p1,p2,p3);

t.Disp(outfile);

outfile<<t.Area()<<endl;

}

infile.close();

outfile.close();

return 0;

}

正常做法：

/\*

声明一个三角形类Triangle，包括三个私有对象成员point1,point2,point3，这三个成员是点类Point的对象；

类Point包括公有成员x(float),y(float)表示点的坐标，包括构造函数给坐标初始化；

类Triangle包括构造函数，求面积的函数float Area()，包括依次输出三个点point1,point2,point3的坐标的函数Disp()；

程序中的输入文件in.txt包含以下数据：

1 1 2 2 3 3

0 0 0 4 4 4

1 1 4 4 1 4

输出文件out.txt包含以下数据（点的x,y坐标之间以一个空格分隔）：

1 1

2 2

3 3

0

0 0

0 4

4 4

8

1 1

4 4

1 4

4.5

提示：程序使用文件实现输入和输出，不可以用 cin 和 cout ；

只能在Begin和End之间输入代码，不能改动程序的任何其余部分。

---------------------------------------------------------------------------------------------------\*/

#include<fstream>

#include<cmath>

using namespace std;

ifstream infile("in.txt");

ofstream outfile("out.txt");

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Begin \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

class Point

{

public: float x,y;

Point(float x1,float y1) {x=x1;y=y1;}

};

class Triangle

{

private: Point point1,point2,point3;

public:

Triangle(Point &p1,Point &p2,Point &p3):point1(p1),point2(p2),point3(p3){}

float Area()

{

float s1 = sqrt((point1.x-point2.x)\*(point1.x-point2.x)+(point1.y-point2.y)\*(point1.y-point2.y));

float s2 = sqrt((point3.x-point2.x)\*(point3.x-point2.x)+(point3.y-point2.y)\*(point3.y-point2.y));

float s3 = sqrt((point3.x-point1.x)\*(point3.x-point1.x)+(point3.y-point1.y)\*(point3.y-point1.y));

float s = 0.5\*(s1+s2+s3);

return sqrt(s\*(s-s1)\*(s-s2)\*(s-s3));

}

void Disp()

{

outfile<<point1.x<<' '<<point1.y<<endl;

outfile<<point2.x<<' '<<point2.y<<endl;

outfile<<point3.x<<' '<<point3.y<<endl;

}

};

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int main( )

{

if(!infile) return -1;

if(!outfile) return -1;

float x1,y1,x2,y2,x3,y3;

while(!infile.eof())

{

infile>>x1>>y1>>x2>>y2>>x3>>y3;

Point p1(x1,y1),p2(x2,y2),p3(x3,y3);

Triangle t(p1,p2,p3);

t.Disp();

outfile<<t.Area()<<endl;

}

infile.close();

outfile.close();

return 0;

}

4./\*

定义时间类Time，包括私有成员变量hour(int)、minute(int)、second(int)；

成员函数set\_time(int h,int m,int s)将读入的3个整数规范化后给3个成员变量赋值，使得

second满60向minute进1，并规范化至[0,59]；minute满60向hour进1，并规范化至[0,59]；hour满24则规范化至[0,23]；

成员函数show\_time(ofstream & output)读取3个成员变量，格式化为00:00:00形式后，输出到文件中；

程序中的输入文件in.txt包含以下数据：

0 0 0

1 59 62

1 63 63

2 63 59

3 123 63

3 126 125

23 59 67

26 68 61

输出文件out.txt包含以下数据（英文冒号）：

00:00:00

02:00:02

02:04:03

03:03:59

05:04:03

05:08:05

00:00:07

03:09:01

程序从文件in.txt中读取数据，输出到文件out.txt中，不可以用 cin 和 cout ；

只能在Begin和End之间输入代码，不能改动程序的任何其余部分。

---------------------------------------------------------------------------------------------------\*/

#include <fstream>

#include <iomanip>

using namespace std;

ifstream infile("in.txt");

ofstream outfile("out.txt");

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Begin \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

class Time

{public:

void set\_time(int h,int m,int s);

void show\_time(ofstream & output);

private:

int hour;

int minute;

int second;

};

void Time::set\_time(int h,int m,int s)

{

second = s % 60;

minute = m + s/60;

hour = h + minute /60;

hour = hour %24;

minute = minute % 60;

}

void Time::show\_time(ofstream & output)

{

output<<setfill('0');

output<<setw(2)<<hour<<":"<<setw(2)<<minute<<":"<<setw(2)<<second<<endl;

}

法二：

class Time

{ private:

int hour;

int minute;

int second;

public:

int set\_time(int h,int m,int s)

{ while(s>0&&s/60>=1)

{ m=m+1;

s=s-60;

}

while(m>0&&m/60>=1)

{ h=h+1;

m=m-60;

}

while(h>=24)

{ h=h-24; }

hour=h;

minute=m;

second=s;

}

void show\_time(ofstream & output)

{

output<<setfill('0');

output<<setw(2)<<hour<<":"<<setw(2)<<minute<<":"<<setw(2)<<second<<endl;

}

};

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int main()

{

if(!infile) return -1;

if(!outfile) return -1;

Time t;

int h,m,s;

while(!infile.eof())

{

infile>>h>>m>>s;

t.set\_time(h,m,s);

t.show\_time(outfile);

}

infile.close();

outfile.close();

return 0;

}

5.

/\*

设计抽象基类Shape,包括求面积的成员函数double area();

由Shape派生出3个派生类：Circle(圆形)、Rectangle(矩形)、Triangle(三角形)

Circle包括半径radius，Rectangle包括长width和宽height，Triangle包括底width和高height；

分别设计Circle(圆形)、Rectangle(矩形)、Triangle(三角形)的构造函数和求面积函数area()，

使得printArea() 函数能够正确输出面积值；

程序中的输入文件in.txt包含以下数据：

2.0

3.5 2.0

3.5 2.0

输出文件out.txt包含以下数据：

12.5664

7

3.5

提示：程序使用文件实现输入和输出，不可以用 cin 和 cout ；

只能在Begin和End之间输入代码，不能改动程序的任何其余部分。

#include <fstream>

using namespace std;

ifstream infile("in.txt");

ofstream outfile("out.txt");

-------------------------------------------------------------------------------\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Begin \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

class Shape

{public:

virtual double area() const =0;

};

class Circle:public Shape

{public:

Circle(double r):radius(r){}

virtual double area() const {return 3.14159\*radius\*radius;};

protected:

double radius;

};

class Rectangle:public Shape

{public:

Rectangle(double w,double h):width(w),height(h){}

virtual double area() const {return width\*height;}

protected:

double width,height;

};

class Triangle:public Shape

{public:

Triangle(double w,double h):width(w),height(h){}

virtual double area() const {return 0.5\*width\*height;}

protected:

double width,height;

};

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void printArea(const Shape &s)

{outfile<<s.area()<<endl;}

int main()

{

if(!infile) return -1;

if(!outfile) return -1;

double r,w,h;

infile>>r;

Circle circle(r);

printArea(circle);

infile>>w>>h;

Rectangle rectangle(w,h);

printArea(rectangle);

infile>>w>>h;

Triangle triangle(w,h);

printArea(triangle);

infile.close();

outfile.close();

return 0;

}

7.

/\*

定义电影类Movie，包括私有对象成员电影名name(string)，开始时间startTime和结束时间endTime；

startTime和endTime是Time类的对象；Time类包括公有成员hour(int)，minute(int)和构造函数；

Movie类包括构造函数，求电影时长(多少分钟)的函数 int Duration()，显示电影播放信息的函数Display();

程序中的输入文件in.txt包含以下数据:

The Fast and the Furious 8

12 30 14 50

Smurfs: The Lost Village

13 9 15 45

Battle of Memories

15 20 16 54

输出文件out.txt包含以下数据（英文冒号）：

The Fast and the Furious 8

start time:12:30

140minutes

Smurfs: The Lost Village

start time:13:9

156minutes

Battle of Memories

start time:15:20

94minutes

程序从文件in.txt中读取数据，输出到文件out.txt中，不可以用 cin 和 cout ；

只能在Begin和End之间输入代码，不能改动程序的任何其余部分。

----------------------------------------------------------------------------------------------------\*/

#include <fstream>

using namespace std;

ifstream infile("in.txt");

ofstream outfile("out.txt");

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Begin \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

class Time

{

public:

int hour,minute;

Time (int h,int m)

{hour = h; minute= m;}

};

class Movie

{

private:

string name;

Time startTime,endTime;

public:

Movie(string str,Time &stime,Time &etime):startTime(stime),endTime(etime){name=str;}

int Duration()

{

int d;

d = (endTime.hour - startTime.hour)\*60+(endTime.minute-startTime.minute);

return d;

}

void Display(ofstream &output)

{

output<<name<<'\n'<<"start time:"<<startTime.hour<<':'<<startTime.minute<<'\n';

}

};

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int main()

{

if(!infile) return -1;

if(!outfile) return -1;

string str;

int hour1,minute1,hour2,minute2;

while(!infile.eof())

{

getline(infile,str);

infile>>hour1>>minute1>>hour2>>minute2;

infile.ignore();

Time startT(hour1,minute1),endT(hour2,minute2);

Movie mov(str,startT,endT);

mov.Display(outfile);

outfile<<mov.Duration()<<"minutes"<<endl;

}

infile.close();

outfile.close();

return 0;

}

/\*

定义箱子类Box，包括私有成员变量高度height(int)、宽度width(int)、长度length(int)，

包括构造函数和成员函数volume()；

构造函数要能够实现main()函数中对象box1,box2,box3,box4的正确构造；

成员函数volume(),求箱子的体积:height\*width\*length

程序中的输入文件in.txt包含以下数据:

(共2组数据，第1组中第1个箱子的height\*width\*length为5\*3\*2，第3个箱子的height\*width为2\*3)

5 3 2

2 3

6 3 5

5 3

输出文件out.txt包含以下数据:

30

30

60

60

90

90

150

150

程序从文件in.txt中读取数据，输出到文件out.txt中，不可以用 cin 和 cout ；

只能在Begin和End之间输入代码，不能改动程序的任何其余部分。

--------------------------------------------------------------------------------------------------\*/

#include <fstream>

using namespace std;

ifstream infile("in.txt");

ofstream outfile("out.txt");

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Begin \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

class Box

{

private:

int height;

int width;

int length;

public:

Box(int h=10,int w=10,int len=10)

{height=h;

width=w;

length=len;

}

int volume( )

{return(height\*width\*length); }

};

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int main( )

{

if(!infile) return -1;

if(!outfile) return -1;

int n1,n2,n3;

while(!infile.eof())

{

infile>>n1>>n2>>n3;

Box box1(n1,n2,n3),box2;

box2=box1;

infile>>n1>>n2;

Box box3(n1,n2);

Box box4(box3);

outfile<<box1.volume()<<'\n'<<box2.volume()<<'\n'

<<box3.volume()<<'\n'<<box4.volume()<<endl;

}

infile.close();

outfile.close();

return 0;

}

/\*

定义学生类Student，包括私有成员变量学号(int)、性别(char)、年龄(int)，包括构造函数，用于对私有成员初始化；

成员函数Display(ostream &output)用于输出：学号(int)、性别(char)、年龄(int)，将这些数据输出到输出流；

程序中的输入文件in.txt包含以下数据（3列数据分别表示学号、性别、年龄） ：

1001 f 18

1002 m 19

1003 m 17

1004 f 18

1005 f 19

输出文件out.txt包含以下数据(数据之间用一个空格隔开）：

1001 f 18

1002 m 19

1003 m 17

1004 f 18

1005 f 19

程序从文件in.txt中读取数据，输出到文件out.txt中，不可以用 cin 和 cout ；

只能在Begin和End之间输入代码，不能改动程序的任何其余部分。

---------------------------------------------------------------------------------------------------\*/

#include <fstream>

using namespace std;

ifstream infile("in.txt");

ofstream outfile("out.txt");

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Begin \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

class Student

{

private:

int num;

char sex;

int age;

public:

Student(int n,char s,int a)

{

num=n; sex=s; age=a;

}

void Display(ostream &output)

{

output<<num<<' '<<sex<<' '<<age<<endl;

}

};

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int main()

{

if(!infile) return -1;

if(!outfile) return -1;

int n;char s;int a;

while(!infile.eof())

{

infile>>n>>s>>a;

Student s1(n,s,a);

s1.Display(outfile);

}

infile.close();

outfile.close();

return 0;

}

/\*

基于下面的main()函数，写出完整的程序，能够编译和运行；

程序中的输入文件in.txt包含以下数据

6S

6Plus

7S

7Plus

定义类Apple，包括私有成员变量name(string)，包括构造函数和析构函数,使得输出文件out.txt包含以下数据

Apple6S

~Apple6S

Apple6Plus

~Apple6Plus

Apple7S

~Apple7S

Apple7Plus

~Apple7Plus

提示：程序使用文件实现输入和输出，不可以用 cin 和 cout ；

只能在Begin和End之间输入代码，不能改动程序的任何其余部分。

---------------------------------------------------------------------------------------------------\*/

#include<string>

#include<fstream>

using namespace std;

ifstream infile("in.txt");

ofstream outfile("out.txt");

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Begin \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

class Apple

{public:

Apple(string str)

{name=str;

outfile<<"Apple"<<name<<endl;}

~Apple( )

{outfile<<"~Apple"<<name<<endl;}

private:

string name;

};

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int main()

{

if(!infile) return -1;

if(!outfile) return -1;

string name;

while(infile>>name)

{

Apple \*pA = new Apple(name);

delete pA;

}

infile.close();

outfile.close();

return 0;

}

/\*

基于下面的main()函数，写出完整的程序，能够编译和运行；

程序中的输入文件in.txt包含以下数据：

1001

1002

1003

定义类Student，包括私有成员变量学号(int)，包括构造函数和析构函数,使得输出文件out.txt包含以下数据：

Constructor1001

Constructor1002

Constructor1003

Destructor1003

Destructor1002

Destructor1001

提示：程序使用文件实现输入和输出，不可以用 cin 和 cout ；

只能在Begin和End之间输入代码，不能改动程序的任何其余部分。

---------------------------------------------------------------------------------------------------\*/

#include <fstream>

using namespace std;

ifstream infile("in.txt");

ofstream outfile("out.txt");

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Begin \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

class Student

{public:

Student(int n)

{num=n;

outfile<<"Constructor"<<num<<endl;}

~Student()

{outfile<<"Destructor"<<num<<endl;}

private:

int num;

};

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void CreateObjects()

{

int n[3];

infile>>n[1]>>n[2]>>n[3];

Student s1[3]={n[1],n[2],n[3]};

}

int main()

{

if(!infile) return -1;

if(!outfile) return -1;

CreateObjects();

infile.close();

outfile.close();

return 0;

}

/\*

基于下面的main()函数，写出完整的程序，能够编译和运行；

程序中的输入文件in.txt包含以下数据:

Huawei

Sumsung

iPhone7

定义类Phone，包括私有成员变量name(string)，包括构造函数，成员函数setName()修改name,

常成员函数getName()获得name，使得输出文件out.txt包含以下数据;

Huawei

Sumsung

iPhone7

提示：程序使用文件实现输入和输出，不可以用 cin 和 cout ；

只能在Begin和End之间输入代码，不能改动程序的任何其余部分。

---------------------------------------------------------------------------------------------------\*/

#include<string>

#include<fstream>

using namespace std;

ifstream infile("in.txt");

ofstream outfile("out.txt");

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Begin \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

class Phone

{public:

Phone(){}

Phone(string str) {name=str;}

void setName(string str) {name = str;}

string getName() const {return name;}

private:

string name;

};

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int main()

{

if(!infile) return -1;

if(!outfile) return -1;

Phone ph1,ph2;

string name;

infile>>name;

ph1.setName(name);

infile>>name;

ph2.setName(name);

Phone const \*pPhone = &ph1;

outfile<<pPhone->getName()<<endl;

pPhone = &ph2;

outfile<<pPhone->getName()<<endl;

infile>>name;

const Phone ph3(name);

outfile<<ph3.getName()<<endl;

infile.close();

outfile.close();

return 0;

}

设计函数fun:从保存n个学生记录的stu1数组中，找出成绩在60和80分之间（包括60和80）的学生数据，并保存在sut2数组中，函数返回值表示找到的学生人数。

其中，学生记录由学号和成绩组成。

请勿改动主函数main和其他函数中的任何内容，仅在函数fun的花括号中填入你编写的若干语句。

------------------------------------------------------------------------------------------------------------------------------------\*/

#include <iostream>

using namespace std;

#include<fstream>

#include<iomanip>

#define N 16

struct student

{

char num[10];

int score;

};

int fun(student stu1[],int n,student stu2[])

{

/\*\*\*\*\*\*\*\*\*\* Begin \*\*\*\*\*\*\*\*\*\*/

int i,j=0;

for (i=0;i<n;i++)

if (stu1[i].score>=60&&stu1[i].score<=80)

{

stu2[j]=stu1[i];

j++;

}

return j;

/\*\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*\*\*/

}

int main()

{

student stu1[N]={{"GA05",85},{"GA03",76},{"GA02",69},{"GA04",85},{"GA01",91},{"GA07",72},{"GA08",64},{"GA06",87},{"GA015",85},{"GA013",91},{"GA012",64},{"GA014",91},{"GA011",91},{"GA017",64},{"GA018",64},{"GA016",72}};

student stu2[N];

int i,n;

n=fun(stu1,N,stu2);

cout<<"所有的学生数据："<<endl;

for(i=0;i<N;i++)

cout<<stu1[i].num<<setw(4)<<stu1[i].score<<endl;

cout<<"60-80分之间的学生数据："<<endl;

for(i=0;i<n;i++)

cout<<stu2[i].num<<setw(4)<<stu2[i].score<<endl;

//结果写入out.txt

ofstream ofile;

ofile.open("out.txt");

ofile<<n<<endl;

for(i=0;i<n;i++)

ofile<<stu2[i].num<<setw(4)<<stu2[i].score<<endl;

ofile.close();

}

设计函数fun:从保存n个学生记录的stu1数组中，找出低于平均分的学生数据，并保存在sut2数组中，函数返回值表示找到的学生人数。

学生的记录由学号和成绩组成，N名学生的数据已在主函数中放入结构体数组中。

请勿改动主函数main和其他函数中的任何内容，仅在函数fun的花括号中填入你编写的若干语句。

-------------------------------------------------------------------------------------------------------------------------\*/

#include <iostream>

using namespace std;

#include<fstream>

#include<iomanip>

#define N 8

struct student

{

char num[10];

int score;

};

int fun(student stu1[],int n,student stu2[])

{

/\*\*\*\*\*\*\*\*\*\* Begin \*\*\*\*\*\*\*\*\*\*/

int i,j=0,sum=0;

float avg=0;

for (i=0;i<n;i++)

{

sum=sum+stu1[i].score;

}

avg=sum/n;

for (i=0;i<n;i++)

if (stu1[i].score<avg)

{

stu2[j]=stu1[i];

j++;

}

return j;

/\*\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*\*\*/

}

int main()

{

student stu1[N]={{"GA05",85},{"GA03",76},{"GA02",69},{"GA04",85},{"GA01",91},{"GA07",72},{"GA08",64},{"GA06",87}};

student stu2[N];

int i,n;

n=fun(stu1,N,stu2);

cout<<"所有学生数据："<<endl;

for(i=0;i<N;i++)

cout<<stu1[i].num<<setw(4)<<stu1[i].score<<endl;

cout<<n<<endl;

cout<<"低于平均分学生数据："<<endl;

for(i=0;i<n;i++)

cout<<stu2[i].num<<setw(4)<<stu2[i].score<<endl;

//结果写入out.txt

ofstream ofile;

ofile.open("out.txt");

ofile<<n<<endl;

for(i=0;i<n;i++)

ofile<<stu2[i].num<<setw(4)<<stu2[i].score<<endl;

ofile.close();

}

一．选择题

1.设有以下程序段，则表达式的值不为100的是\_\_\_\_D. (++p)->b\_\_\_。

struct st

{

int a,\*b;

};

int main()

{

int m1[] ={10,100}, m2[]={100,200};

st \*p, x[]={99,m1,100,m2};

p=x;

...

}

2.若有以下定义和语句：

struct student

{

int num,age;

};

student stu[3]={{1001,20},{1002,19},{1003,21}};

student \*p=stu;

则以下结果不是1002的是\_\_D. (++stu)->num\_\_\_\_\_。

3.设有以下的结构体说明和变量定义，如下图所示，指针p指向变量one，指针q指向变量two，则不能将结点连接到结点one之后的语句是\_\_\_\_A. p.next=&two;\_\_\_。

struct node

{

int n;

struct node \*next;

}one, two, \*p=&one, \*q=&two;



4.有以下说明和定义语句:

struct student

{ int age;

char num[8];

} ;

struct student stu[3] = {{20,"20041"}, {21,"20042"},{19,"20043"}};

struct student \*p = stu;

以下选项中引用结构体变量成员的表达式错误的是( C.stu[3].age )

.

5.根据以下定义，能输出字母M的语句是 B.cout<<class[2].name[0]<<endl; 。

struct person

{

char name[9];

Int age;

};

person class[4]={{"John",17},{"Paul",19},{"Mary",18},{"Adam",16}};

6.以下程序的运行结果是（“1”）。

#include<iostream>

using namespace std;

class Test

{

public:

Test(){}

Test(Test&t)

{

cout<<1;

}

};

Test fun(Test&u)

{

Test t=u;

return t;

}

int main()

{

Test x,y;

x=fun(y);

return 0;

7.构造函数是在( 创建对象)时被执行的.

8.下列关于类和对象的描述中不正确的是( 一个类只能有一个对象 ).

9.通常，复制构造函数的参数是( 某个对象的引用名 ).

1. C++语言结构体类型变量在程序执行期间（ C.所有成员一直驻留在内存中 ）。

以下程序运行后的输出结果是 C.2041 2044 。

struct STU

{ char name[10]; int num; };

void f1(struct STU c)

{ struct STU b = {"LiSiGuo", 2042};

c = b;

}

void f2(struct STU \*c)

{ struct STU b = {"SunDan", 2044};

\*c = b;

}

int main()

{ struct STU a = {"YangSan", 2041}, b = {"WangYin", 2043};

f1(a); f2(&b);

cout<<a.num<<' '<<b.num<<endl;

return 0;

}

12.static struct {int a1;float a2;char a3;}a[10]={1,3.5,'A'}; 说明数组a是地址常量，它有10个结构体型的下标变量，采用静态存

储方式，其中被初始化的下标变量是（ B.a[0] ）。

13.以下程序的输出结果是 6 。

int main()

{ struct cmplx

{ int x;；

int y；

}cnum[2] = {1, 3, 2, 7};

cout<<cnum[0].y / cnum[0].x \*cnum[1].x<<endl;

}

13设有以下程序段，则表达式的值不为100的是\_\_(++p)->b\_\_\_\_\_。

.struct st

{

int a,\*b;

};

int main()

{

int m1[] ={10,100}, m2[]={100,200};

st \*p, x[]={99,m1,100,m2};

p=x;

...

}

14.设有以下语句

typedef struct S

{

int g;

char h;

} T;

则下面叙述中正确的是 可以用T定义结构体变量 。



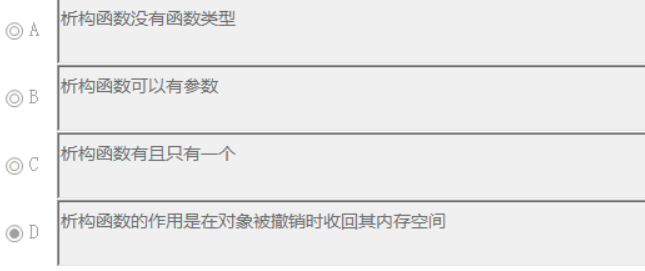
15.当定义一个结构体变量时，系统分配给它的内存是（各成员所需内存量的总和 ）。

16.下列有关类和对象的说法中，正确的是（ 类与对象的关系和数据类型与变量的关系相似 ）。

17.下列关于类和对象的描述中不正确的是( 一个类只能有一个对象 ).

18.假定C为一个类，则执行"C a1(20), a2[5];"语句时，自动调用该类构造函数的次数为 ( 6 ).

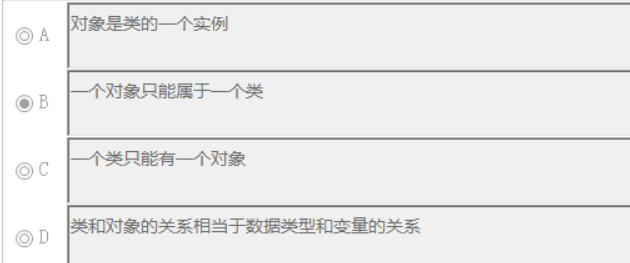
19.下列关于析构函数的描述中不正确的是( B.析构函数可以有参数 ).



20.类成员的访问权限不包括（**A**）



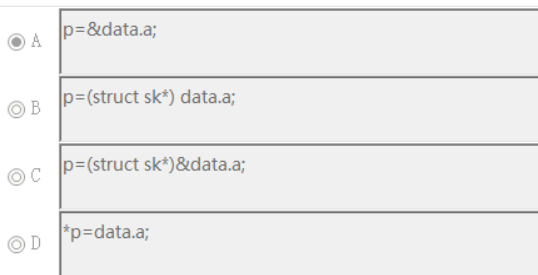
21：下列错误的是（C）



1. 相同结构体类型的变量之间，可以（ 赋值 ）。

23.设有如下定义： struct sk {int a; float b;} data,\*p;

若要使p指向data中的a域，正确的赋值语句是（ C.p=(struct sk\*)&data.a; ）。



24.设有结构体变量声明struct date {int year,month,day;}today;

则sizeof(struct date)和sizeof(today)的值分别为 ( D.12，12 )。

