Ass-2

Q1 Write a program to maintain a telephone directory. Use add () and show () methods to add new entries and

display the telephone numbers of person when the name of the person is given.

30/7/2016

#include<iostream.h>

#include<conio.h>

#include<string.h>

int count=0;

char a[20];

class phone

{

char no[12];

char name[20];

public:

void add()

{

cout<<"\nEnter Name :";

cin>>name;

cout<<"\nEnter Phone Number :";

cin>>no;

cout<<name<<endl<<no;

}

int show(char a[])

{

if(strcmp(a,name)==0)

{

cout<<"\n\nName :"<<name;

cout<<"\nPhone No :"<<no;

return 0;

}

else

return 1;

}

};

void main()

{

int i,ch,b;

phone d[5];

do

{

clrscr();

cout<<"\n\n1. Add details";

cout<<"\n2. Show details";

cout<<"\n0. Exit";

cout<<"\n\nEnter Your Choice :";

cin>>ch;

switch(ch)

{

case 1:

d[count].add();

count++;

break;

case 2:

cout<<"\nEnter Name :";

cin>>a;

for(i=0;i<5;i++)

{

b=d[i].show(a);

if(b==0)

break;

}

if(b==1)

cout<<"\nName not found!!!";

break;

case 0:break;

default:

cout<<"\nInvaild Input!!!";

}

getch();

}while(ch!=0);

}

Q2 Create two classes DM and DB which store the value of the distances. DM stores distance in meters and

centimeters and DB stores distance in feet and inches. Write a program that can add one object of DM with

another object of DB. Use a friend function to carry out the addition operation. And this function will

display answer in meter and centimeter.

13/8/16

#include <iostream.h>

#include <conio.h>

class db;

class dm

{

float mt;

int cm;

public:

void getdata(void);

void display(void);

friend dm add(dm,db);

};

class db

{

int feet;

float inches;

public:

void getdata(void);

void display(void);

friend dm add(dm,db);

};

void dm :: getdata(void)

{

clrscr();

cout<<"\t\tDM GETDATA FUNCTION\n\n";

cout<<"\n\nEnter Values for metres :-";

cin>>mt;

cout<<"Enter Values for centimetres:-";

cin>>cm;

}

void dm :: display(void)

{

cout<<"\n\nThe value of distance in metres is "<<mt;

cout<<"\nThe value of distance in Centimetres is "<<cm;

}

void db :: getdata(void)

{

clrscr();

cout<<"\t\tDB GETDATA FUNCTION\n\n";

cout<<"\n\nEnter Values for feet :-";

cin>>feet;

cout<<"Enter Values for inches :-";

cin>>inches;

}

void db :: display(void)

{

cout<<"\n\nThe value of distance in feet is "<<feet; cout<<"\nThe value of distance in inches is "<<inches;

}

dm add(dm a,db b)

{

dm temp;

temp.cm=a.cm+(b.feet\*30)+((b.inches\*30)/12.0);

temp.mt=a.mt+(temp.cm % 100);

return(temp);

}

void main()

{

dm a;

a.getdata();

db b;

b.getdata();

clrscr();

cout<<"\n\t\tAFTER CONVERSION AND THEIR ADDITION IS PERFORMED\n";

dm extra;

extra=add(a,b);

extra.display();

getch();

}

Q4 Write a program that consists of two classes, Rupee and Dollar. Provide conversion function to carry out

conversion from object of one type to another.

20/8/16

#include<iostream.h>

#include<conio.h>

class rupees;

class dollar

{

public:

float doll;

dollar()

{

doll=0;

}

dollar(float doll1)

{

doll=doll1;

}

int getdoll()

{

return(doll);

}

void display()

{

cout<<"\n\n Dollar:"<<"$"<<doll;

}

};

class rupee

{

public:

float rup;

rupee(float rup1)

{

rup=rup1;

}

rupee(dollar d)

{

rup=d.getdoll()\*(51.1265);

}

operator dollar()

{

dollar d;

d.doll=rup\*(0.0196);

return d;

}

void display()

{

cout<<"\n\n Rupees:"<<"Rs"<<rup;

}

};

void main()

{

float d1,r1;

clrscr();

cout<<"\n\t\t TYPE CONVERSION";

cout<<"\n\t\t \*\*\*\*\*\*\*\*\*\*\*\*";

cout<<"\n\n Enter the rupees:";

cin>>r1;

rupee rs(r1);

dollar dr;

dr=rs;

cout<<"\n\n After Conversion";

dr.display();

cout<<"\n\n Enter the Dollar:";

cin>>d1;

dollar dr1(d1);

rs=dr1;

cout<<"\n\n After Conversion";

rs.display();

getch();

}

Q5 Define a class to represent a bank account for handling 10 customers. Include the following members.

Data members:

1) Name of the A/c Holder

2) Account number

3) Balance amount in the account

Member function:

1) To display the values

2) To deposit an amount

3) To withdraw an amount

30/7/16

#include<iostream.h>

#include<conio.h>

#include<iomanip.h>

#include<string.h>

int count=0;

class bank

{

char name[20],acc[16];

int bal;

public:

int match(char c[])

{

if(strcmp(c,acc)==0)

return 0;

else

return 1;

}

void add()

{

cout<<"\n\nName :";

cin>>name;

cout<<"\nAccount Number :";

cin>>acc;

cout<<"\nBalance :";

cin>>bal;

}

void dis()

{

cout<<setw(15)<<name;

cout<<setw(18)<<acc;

cout<<setw(10)<<bal<<endl;

}

void dipt()

{

int p;

cout<<"\nHow much money you want to deposit :";

cin>>p;

bal=bal+p;

}

void wdro()

{

int p;

cout<<"\nHow much money you want to withdhraw :";

cin>>p;

bal=bal-p;

}

};

void main()

{

int i,ch,j,k;

char ac[16];

bank b[5];

do

{

clrscr();

cout<<"\n\n1. Add details";

cout<<"\n2. display details";

cout<<"\n3. diposit";

cout<<"\n4. withdhraw";

cout<<"\n0. Exit";

cout<<"\n\nEnter Your Choice :";

cin>>ch;

switch(ch)

{

case 1:

b[count].add();

count++;

break;

case 2:

cout<<setw(15)<<"NAME"<<setw(18)<<"A//C NO."<<setw(10)<<"BALANCE\n";

cout<<"----------------------------------------------------\n";

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

{

b[i].dis();

}

break;

case 3:

cout<<"\nEnter A//C NO :";

cin>>ac;

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

{

j=b[i].match(ac);

if(j==0)

{

k=i;

break;

}

}

if(j==1)

cout<<"\nAccount not found!!!";

else

b[k].dipt();

break;

case 4:

// comp();

cout<<"\nEnter A//C NO :";

cin>>ac;

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

{

j=b[i].match(ac);

if(j==0)

{

k=i;

break;

}

}

if(j==1)

cout<<"\nAccount not found!!!";

else

b[k].wdro();

break;

case 0:break;

default:

cout<<"\nInvaild Input!!!";

}

getch();

}while(ch!=0);

}

Q7 Write a program that consists of 2 class time12 and time24. The first maintain time on 12-hour basis,

whereas the other one maintain time on 24-hour basis. Provide conversion function to carry out conversion

from object of one type to another.

13/8/16

#include<iostream.h>

#include<string.h>

#include<conio.h>

class time24

{

int hours,minutes,seconds;

public:

time24()

{

hours=minutes=seconds=0;

}

time24(int h,int m,int s)

{

hours=h;

minutes=m;

seconds=s;

}

void display()

{

if(hours<10)

cout<<'0';

cout<<hours<<":";

if(minutes<10)

cout<<'0';

cout<<minutes<<":";

if(seconds<10)

cout<<'0';

cout<<seconds;

}

friend class time12;

};

class time12

{

int pm;

int hour,minute;

char \*am\_pm;

public:

time12(time24);

void display()

{

cout<<hour<<":";

if(minute<10) cout<<'0';

cout<<minute<<" ";

cout<<am\_pm;

}

};

time12::time12(time24 t24)

{

int hrs24=t24.hours;

int pm=t24.hours<12 ? 0:1;

int min=t24.seconds<30 ? t24.minutes:t24.minutes+1;

if(min==60)

{

min=0;

++hrs24;

if(hrs24==12 || hrs24==24)

pm=(pm==1)? 0:1;

}

int hrs12=(hrs24<13) ? hrs24 : hrs24-12;

if(hrs12==0)

{

hrs12=12;

pm=0;

}

if(pm==1)

{

am\_pm=" pm";

}

else

{

am\_pm=" am";

}

hour=hrs12;

minute=min;

}

int main()

{

clrscr();

int h1,m1,s1,x=1;

while(x==1)

{

cout<<"enter 24-hour time:\n";

cout<<"Hours(0-23):";

cin>>h1;

if(h1>23)

return(1);

cout<<"Minutes:";

cin>>m1; cout<<"Seconds:";

cin>>s1;

time24 t24(h1,m1,s1);

cout<<"you entered:";

t24.display();

cout<<endl;

time12 t12=t24;

cout<<"\n12-hour time:\n";

t12.display();

cout<<endl;

x=0;

}

getch();

return 0;

}

Q9 Create a class with at least two data members and overload >> and << operators.

3/9/16

#include<iostream.h>

#include<conio.h>

class vector

{

int v;

public:

friend istream & operator>>(istream & in, vector &s1)

{

in>>s1.v;

return in;

}

friend ostream & operator<<(ostream & on, vector &s1)

{

on<<s1.v;

return on;

}

};

void main()

{

clrscr();

vector v1,v2;

cout<<"v1 :";

cin>>v1;

cout<<"v2 :";

cin>>v2;

cout<<"v1 :"<<v1<<endl;

cout<<"v2 :"<<v2<<endl;

getch();

}

Q10 Create a class “sample” that stores the weight and height. Ovreload the (+=)/(-=) operators to add two values.

3/9/16

#include<iostream.h>

#include<conio.h>

class sample

{

float height,weight;

public:

void operator +=(sample);

void operator -=(sample);

void get(float h,float w)

{

height=h;

weight=w;

}

void put()

{

cout<<"\n\nheight :"<<height;

cout<<"\nweight :"<<weight;

}

};

void sample::operator +=(sample s)

{

cout<<"\n\noverload +=...";

height=height+s.height;

weight=weight+s.weight;

}

void sample::operator -=(sample s)

{

cout<<"\n\noverload -=...";

height=height-s.height;

weight=weight-s.weight;

}

void main()

{

float h,w;

sample s1,s2;

clrscr();

cout<<"\n\nenter value for s1...";

cout<<"\nheight: ";

cin>>h;

cout<<"\nweight :";

cin>>w;

s1.get(h,w);

cout<<"\n\nenter value for s2...";

cout<<"\nheight: ";

cin>>h;

cout<<"\nweight :";

cin>>w;

s2.get(h,w);

s1+=s2;

s1.put();

s1-=s2;

s1.put();

getch();

}

Q11 Wap to compare and concatenate two strings using == and + operators overloading respectively.

3/9/16

#include<iostream.h>

#include<conio.h>

#include<string.h>

class str

{

int len;

char \*p;

public:

str()

{

len=0;

p=0;

}

str(const char \*s)

{

len=strlen(s);

p=new char[len+1];

strcpy(p,s);

}

/\* str(const str \*s)

{

len=s.len;

p=new char[len+1];

strcpy(p,s.p);

} \*/

~str()

{

delete p;

}

friend str operator +(const str &s,const str &t);

friend int operator ==(const str &s,const str &t);

friend void show(const str s);

};

str operator +(const str &s,const str &t)

{

str temp;

temp.len=s.len+t.len;

temp.p=new char[temp.len+1];

strcpy(temp.p,s.p);

strcat(temp.p,t.p);

return (temp);

}

int operator==(const str &s,const str &t)

{

int m=strlen(s.p);

int n=strlen(t.p);

if(m<=n)

return (1);

else

return (0);

}

void show(const str s)

{

cout<<s.p;

}

void main()

{

// char \*a,\*b;

clrscr();

/\* cout<<"\nenter string 1 :";

cin>>a;

str s1=a;

cout<<"\nenter string 2 :";

cin>>b; \*/

str s1="janak";

str s2="vaghela";

str s3,t1,t2;

t1=s1;

t2=s2;

s3=s1+s2;

cout<<"\n\nAfter concate...\n\n\t";

show(s3);

cout<<endl<<endl;

if(s1==s2)

{

show(s1);

cout<<" smaller than ";

show(s2);

}

else

{

show(s2);

cout<<" smaller than ";

show(s1);

}

getch();

}

Ass-3

Q1 Create a class emp which consists of members empno, empname. Derive a new class income from emp class. Income class consists of data members like basic salary, da, hra. Derive another class expense from emp which consists of 3 data members income tax, laundry and food. Derive a new class from income and expense and find out net saving of employee.

#include<iostream.h>

#include<conio.h>

class emp

{

int eno;

char name[20];

public:

void get()

{

cout<<"\nEnter emp No :";

cin>>eno;

cout<<"\nName :";

cin>>name;

}

void put()

{

cout<<"\nEno :"<<eno;

cout<<"\nName :"<<name;

}

};

class income:public virtual emp

{

protected:

int bs,da,hra;

public:

void getdata()

{

cout<<"\nbasic salary :";

cin>>bs;

cout<<"\nDA :";

cin>>da;

cout<<"\nHRA :";

cin>>hra;

}

void putdata()

{

cout<<"\nbasic salary :"<<bs;

cout<<"\nDA :"<<da;

cout<<"\nHRA :"<<hra;

}

};

class exp:public virtual emp

{

protected:

int it,laundry,food;

public:

void getexp()

{

cout<<"\income tax :";

cin>>it;

cout<<"\nlaundry :";

cin>>laundry;

cout<<"\nfood :";

cin>>food;

}

void putexp()

{

cout<<"\nincome tax :"<<it;

cout<<"\nlaundry :"<<laundry;

cout<<"\nfood :"<<food;

}

};

class cal:public income,public exp

{

int ns,saving;

public:

void netsal()

{

ns=bs+hra+da;

saving=ns-it-laundry-food;

cout<<"\n\nnet salary :"<<ns<<"\nnet saving :"<<saving;

}

};

void main()

{

cal e;

clrscr();

e.get();

e.getdata();

e.getexp();

cout<<"\n\nAfter calculation.....\n";

e.put();

e.putdata();

e.putexp();

e.netsal();

getch();

}

2) Create a base class shape and declare 2 data members x a

nd y in shape.

Derive two classes rectangle and triangle from shape class.

Write function to calculate area in rectangle and triangle class.

i. Area of rectangle=x\*y

ii.Area of triangle=1/2\*x\*y

#include<iostream.h>

#include<conio.h>

class shape{

protected:

int x;

int y;

public:

shape(int a,int b)

{

x=a;y=b;

}

};

class rectangle : public shape{

public:

rectangle(int a,int b):shape(a,b) { }

void area()

{

cout<<"\nrectangle area is "<<x\*y;

}

};

class triangle : public shape{

public:

triangle(int a,int b):shape(a,b){

}

void area(){

cout<<"\ntriangle area is "<<(0.5)\*x\*y;

}

};

void main()

{

clrscr();

rectangle r(15,20);

triangle t(15,10);

r.area();

t.area();

getch();

}

Q4 Class student contains roll no, name and course as data member and input\_student and display\_student as member function. A derived class exam is created from the class student with publicly inherited. The derived class contains marks1, mark2, mark3 as marks of three subjects and input\_marks and display\_result as member functions. Create an array of object of the exam class and display the result of 5 students.

#include<iostream.h>

#include<conio.h>

class student

{

protected:

int no;

char name[20],course [20];

public:

void gets()

{

cout<<"\nenter the no:";

cin>>no;

cout<<"\nenter the name:";

cin>>name;

cout<<"\nenter the course:";

cin>>course;

}

void displays()

{

cout<<"\nroll\_no:"<<no;

cout<<"\nname:"<<name;

cout<<"\ncourse:"<<course;

}

};

class mark:public student

{

protected:

int m1,m2,m3;

int total;

public:

void get\_mark()

{

cout<<"\nenter the mark 1:";

cin>>m1;

cout<<"\nenter the mark 2:";

cin>>m2;

cout<<"\nenter the mark 3:";

cin>>m3;

}

void display\_mark()

{

cout<<"\nmark1:"<<m1;

cout<<"\nmark2:"<<m2;

cout<<"\nmark3:"<<m3;

}

void result()

{

total=m1+m2+m3;

cout<<"\n\ntotal:"<<total;

}

};

const int size=5;

void main()

{

mark s[size];

int i;

clrscr();

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

{

cout<<"\ndetails of student"<<i+1;

s[i].gets();

s[i].get\_mark();

}

cout<<"\n";

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

{

cout<<"\nstudent"<<i+1;

s[i].displays();

s[i].display\_mark();

s[i].result();

}

getch();

}

Q7 Class polygon contains data member width and height and public method set\_value() to assign values to width and height. Class Rectangle and Triangle are inherited from polygon class. Both the classes contain public method calculate\_area() to calculate the area of Rectangle and Triangle. Use base class pointer to access the derived class object and show the area calculated.

#include<iostream.h>

#include<conio.h>

class polygon

{

protected:

int width,height;

public:

void setval()

{

cout<<"\n\nWidth :";

cin>>width;

cout<<"\nheight :";

cin>>height;

}

virtual void cal\_area() {}

};

class rectangle:public polygon

{

public:

void cal\_area()

{

cout<<"\n\narea of rectangle :"<<height\*width;

}

};

class triangle:public polygon

{

float x;

public:

void cal\_area()

{

x=(0.5\*height\*width);

cout<<"\n\narea of triangle :"<<x;

}

};

void main()

{

polygon \*p1,\*p2;

rectangle r;

triangle t;

clrscr();

p1=&r;

p2=&t;

cout<<"\n\ncalculation for rectangle.....\n";

p1->setval();

p1->cal\_area();

cout<<"\n\ncalculation for triangle.....\n";

p2->setval();

p2->cal\_area();

getch();

}

Q8 Write a program to create a class shape with functions to find area of and display the name of the shape and other essential component of the class. Create derived classes circle, rectangle and trapezoid each having overridden functions area and display. Write a suitable program to illustrate virtual functions.

#include<iostream.h>

#include<conio.h>

class shape

{

protected:

int x;

int y;

public:

shape(int a,int b)

{

x=a;y=b;

}

};

class rectangle : public shape

{

public:

rectangle(int a,int b):shape(a,b) { }

void area()

{

cout<<"\nrectangle area is "<<x\*y;

}

};

class circle : public shape

{

public:

circle(int a,int b):shape(a,b){ }

void area()

{

cout<<"\ntriangle area is "<<3.14\*x\*x;

}

};

class trapezoid : public shape

{

public:

trapezoid(int a,int b):shape(a,b){ }

void area()

{

int h,ans;

cout<<"\n H :";

cin>>h;

ans=(x+y)\*h\*0.5;

cout<<"\ntriangle area is "<<ans;

}

};

void main()

{

clrscr();

rectangle r(15,20);

circle c(15,10);

trapezoid t(8,10);

r.area();

c.area();

t.area();

getch();

}

Q-9 Write a program with Student as abstract class and create derive classes Engineering,Medicine and Science from base class Student. Create the objects of the derived classes and process them and access them using array of pointer of type base class Student.

#include<iostream.h>

#include<conio.h>

class student

{

protected:

int r;

public:

virtual void getdata(){cout<<"\n";}

virtual void display(){cout<<"\n";}

};

class Engineering:public student

{

char name[30];

public:

void getdata()

{

cout<<"\n\t\tRoll No.: ";

cin>>r;

cout<<"\n\t\tName : ";

cin>>name;

}

void display()

{

cout<<"\n\t\tRoll No. : "<<r;

cout<<"\n\t\tName : "<<name;

}

};

class medicine:public student

{

char name[30];

public:

void getdata()

{

cout<<"\n\t\tRoll No.: ";

cin>>r;

cout<<"\n\t\tName : ";

cin>>name;

}

void display()

{

cout<<"\n\t\tRoll No. : "<<r;

cout<<"\n\t\tName : "<<name;

}

};

class science :public student

{

char name[30];

public:

void getdata()

{

cout<<"\n\t\tRoll No.: ";

cin>>r;

cout<<"\n\t\tName : ";

cin>>name;

}

void display()

{

cout<<"\n\t\tRoll No. : "<<r;

cout<<"\n\t\tName : "<<name;

}

};

void main()

{

student \*s[3];

Engineer e;

medicine m;

science sc;

s[1]=&e;

s[2]=&m;

s[3]=&sc;

cout<<"\nEngineer....";

s[1]->getdata();

s[1]->display();

cout<<"\nMedicine....";

s[2]->getdata();

s[2]->display();

}

q-10 discuss the syntax for creating user define manipulators. Design a single manipulator to provide the following output specifications for float values:

1. 10 column width
2. Right-justified
3. Tow digits precision
4. Filling of unused places with \*
5. Trailing zeros shown

#include<iostream.h>

#include<conio.h>

#include<iomanip.h>

ostream & formate(ostream & output)

{

output.width(10);

output.setf(ios::right,ios::adjustfield);

output.precision(2);

output.fill('\*');

output.setf(ios::showpoint);

return output;

}

void main()

{

clrscr();

cout<<formate<<"132.3200";

getch();

}

11.8 the program uses the "STOCK.DAT" file created for five items and performs the following operations on the file:

1. adds a new item to the file.

2. modifies the details of an item.

3. displays the contents of the file.

#include<iostream.h>

#include<conio.h>

#include<fstream.h>

#include<iomanip.h>

class Inventory

{

clrscr();

char name[10];

int code;

float cost;

public:

void getdata(void);

void putdata(void);

};

void Inventory::getdata(void)

{

cout<<"Enter name:";

cin>>name;

cout<<"Enter code:";

cin>>code;

cout<<"Enter cost:";

cin>>cost;

}

void Inventory::putdata(void)

{

cout<<setw(10)<<name

<<setw(10)<<code

<<setprecision(2)

<<setw(10)<<cost

<<endl;

}

int main()

{

clrscr();

Inventory item;

fstream inoutfile("STOCK.DAT",ios::ate | ios::in | ios::out | ios::binary);

inoutfile.seekg(0,ios::beg);

cout<<"current contents of stock"<<"\n";

while(inoutfile.read((char \*) & item,sizeof item));

{

item.putdata();

}

inoutfile.clear();

cout<<"add item\n\n";

item.getdata();

char ch;

cin.get(ch);

inoutfile.write((char \*) & item,sizeof(item));

inoutfile.seekg(0);

cout<<"contents append file"<<"\n";

while(inoutfile.read((char \*) & item,sizeof item));

{

item.putdata();

}

int last=inoutfile.tellg();

int n=last/sizeof(item);

cout<<"Number of objects =" <<n<<"\n";

cout<<"Total bytes in the file="<<last<<"\n";

cout<<"enter objects number to be updated \n";

int object;

cin>>object;

cin.get(ch);

int location = (object-1) \* sizeof(item);

if(inoutfile.eof())

inoutfile.clear();

inoutfile.seekp(location);

cout<<"enter new values of the object \n";

item.getdata();

cin.get();

inoutfile.write((char \*) & item,sizeof item)<<flush;

inoutfile.seekp(0);

cout<<"contents of updated file:\n";

while(inoutfile.read((char \*) & item,sizeof item));

{

item.putdata();

}

inoutfile.close();

getch();

return 0;

}

Class “emp” containing the data members name of employee,emp no ,basic salary,allowance. Take appropriate member function and get data into class, then after write data into data file called “emp.dat” until user choice. Display the information of all employee from a file along with pf, total salary.use read() and write() binary function for file.

#include<iostream.h>

#include<conio.h>

#include<fstream.h>

#include<iomanip.h>

class emp

{

clrscr();

char name[10];

int eno,salary,allow;

public:

void readdata(void);

void writedata(void);

};

void emp::readdata(void)

{

cout<<"Enter name:";

cin>>name;

cout<<"Enter no:";

cin>>eno;

cout<<"Enter salary:";

cin>>salary;

cout<<"Enter allow:";

cin>>allow;

}

void emp::writedata(void)

{

cout<<setiosflags(ios::left)

<<setw(10)<<name

<<setiosflags(ios::right)

<<setw(10)<<eno

<<setprecision(2)

<<setw(10)<<salary

<<setw(10)<<allow

<<endl;

}

int main()

{

emp e[3];

fstream file;

file.open("emp.DAT",ios::in | ios::out);

cout<<"Enter details for three emp \n";

for(int i=0;i<3;i++)

{

e[i].readdata();

file.write((char \*) & e[i],sizeof(e[i]));

}

file.seekg(0);

cout<<"\output\n\n";

for(i=0;i<3;i++)

{

file.read((char \*) & e[i],sizeof(e[i]));

e[i].writedata();

}

file.close();

getch();

return 0;

}