/\*File Name :file\_note.cc

Author Name :D.Bhagya raju

Created Date :16-04-2020

Description :To perform the list of operations specified

Requirements :#include<iostream>,#include<fstream>,#include<vector>,#include<stdio.h>,#include<string.h>\*/

#include<iostream>

//including the container

#include<vector>

#include<fstream>

#include<stdio.h>

#include<string.h>

using namespace std;

class student{

charc\_Key[500];

charc\_Value[100];

public:

/\*Function Name :getkey

Parameters :no parameter

Return Type :no return type

Usage :to get the input of key the private member\*/

voidgetkey()

{

cout<<"Enter key:";

getchar();

cin.get(c\_Key,500,'\n');

}

/\*Function Name :getvalue

Parameters :no parameter

Return Type :no return type

Usage :to get the input of value the private member\*/

voidgetvalue()

{

cout<<"Enter value:";

getchar();

cin.get(c\_Value,100,'\n');

}

/\*Function Name :outkey

Parameters :no parameter

Return Type :char return type

Usage :to return the key\*/

char\* outkey()

{

returnc\_Key;

}

/\*Function Name :outvalue

Parameters :no parameter

Return Type :string return type

Usage :to return value\*/

char\* outvalue()

{

returnc\_Value;

}

void display()

{

cout<<c\_Key<<"\t"<<c\_Value<<endl;

}

};

/\*Function Name :removefile

Parameters :no parameter

Return Type :no return type

Usage :to remove file if file is empty\*/

voidremovefile()

{

student s;

intiCount;

ifstream in("note.ini",ios::in);

while(1)

{

//reading the contents in file

in.read((char\*)&s,sizeof(s));

iCount++;

//checking for end of file

if(in.eof())break;

}

if(iCount==0)

{

cout<<"file is removed"<<endl;

//to remove file

remove("note.ini");

}

}

/\*Function Name :searchupdate

Parameters :two parameter(vector object and char)

Return Type :int return type

Usage :to update the duplicate values\*/

intsearchupdate(vector<student>&std,char\* cOption)

{

student s;

//iterator creation to go word to word in file

vector<student>:: iterator it;

//inorder to write updated value opened in write mode

ofstreamoutfile("note.ini",ios::trunc|ios::out);

for(it=std.begin();it!=std.end();it++)

{

s=\*it;

if(strcmp(s.outkey(),cOption)==0)

{

//the value is been deleted

std.erase(it);

cout<<"Enter value to update"<<endl;

s.getvalue();

//the new value is pushed into file

std.push\_back(s);

return 1;

}

}

for(it=std.begin();it!=std.end();it++)

{

s=\*it;

//the file again written with updated values

outfile<<s.outkey()<<"="<<s.outvalue()<<endl;

}

outfile.close();

}

/\*Function Name :addtofile

Parameters :one vector object parameter

Return Type :no return type

Usage :to add records in to file and store them in to a vector\*/

voidaddtofile(vector<student>&std)

{

student s;

vector<student>:: iterator it;

charcOption='y';

while(cOption=='y')

{

ofstreamoutfile("note.ini",ios::trunc|ios::out);

s.getkey();

if(searchupdate(std,s.outkey())==1)

//checking for if the key is already present in the file or not

cout<<"updated"<<endl;

else

{

s.getvalue();

std.push\_back(s);

for(it=std.begin();it!=std.end();it++)

{

s=\*it;

outfile<<s.outkey()<<"="<<s.outvalue()<<endl;

}

}

cout<<"Do you want to append student data(y/n)?";

cin>>cOption;

outfile.close();

}

}

/\*Function Name :deletefromfile

Parameters :one vector object parameter

Return Type :no return type

Usage :to delete a record from file\*/

voiddeletefromfile(vector<student>&std)

{

charckey[500];

student s;

vector<student>:: iterator it;

ofstreamoutfile("note.ini",ios::trunc|ios::out);

cout<<"Enter the key to delete:";

getchar();

cin.get(ckey,500,'\n');

for(it=std.begin();it!=std.end();it++)

{

s=\*it;

if(strcmp(s.outkey(),ckey)==0)

{

std.erase(it);

cout<<"Deleted"<<endl;

}

}

removefile();

for(it=std.begin();it!=std.end();it++)

{

s=\*it;

outfile<<s.outkey()<<"="<<s.outvalue()<<"\n";

}

outfile.close();

}

/\*Function Name :showfromfile

Parameters :no parameter

Return Type :no return type

Usage :to display the contents in the file\*/

voidshowfromfile()

{

student s;

ifstream in("note.ini",ios::in);

charcdata[500];

while(!in.eof())

{

in.getline(cdata,500);

cout<<cdata<<endl;

}

}

/\*Function Name :searching

Parameters :one vector object parameter

Return Type :no return type

Usage :to search for a key or value and displaying\*/

void searching(vector<student>&std)

{

student s;

charcCheck;

charcSearch\_key[500];

charcSearch\_value[100];

vector<student>:: iterator it;

ifstream in("note.ini",ios::in);

cout<<"do u want search key or value(k/v):";

cin>>cCheck;

if(cCheck=='k')

{

cout<<"Enter key u want to search:";

getchar();

cin.get(cSearch\_key,500,'\n');

for(it=std.begin();it!=std.end();it++)

{

s=\*it;

if(strcmp(cSearch\_key,s.outkey())==0)

cout<<s.outkey()<<"="<<s.outvalue()<<endl;

}

}

else if(cCheck=='v')

{

cout<<"Enter value u want to search:";

getchar();

cin.get(cSearch\_value,100,'\n');

for(it=std.begin();it!=std.end();it++)

{

s=\*it;

if(strcmp(cSearch\_value,s.outvalue())==0)

cout<<s.outkey()<<"="<<s.outvalue()<<endl;

}

}

else

cout<<"Enter k or v"<<endl;

}

int main(intargc,char\* argv[])

{

if(argc==2)

{

if(strcmp(argv[1],"-h")==0)

{

cout<<"select required option"<<endl;

}

}

else

{

vector<student>std;

vector<student>:: iterator it;

charcOpt;

while(1)

{

cout<<"Select your option 1.add 2.delete 3.display 4.search"<<endl;

cin>>cOpt;

switch(cOpt)

{

case '1':

addtofile(std);

break;

case '2':

deletefromfile(std);

break;

case '3':

showfromfile();

break;

case '4':

searching(std);

break;

default:

return 0;

}

}

}

return 0;

}