#include<iostream>

#include<iomanip>

#include<fstream>

#include<cstring>

using namespace std;

class EMP\_CLASS

{

typedef struct EMPLOYEE

{

char name[10];

int emp\_id;

int salary;

char desig[20];

}Rec;

typedef struct INDEX

{

int emp\_id;

int position;

}Ind\_Rec;

Rec Records;

Ind\_Rec Ind\_Records;

public:

EMP\_CLASS();

void Create();

void Display();

void Update();

void Delete();

void Append();

void search();

};

EMP\_CLASS::EMP\_CLASS()

{

strcpy(Records.name,"");

}

void EMP\_CLASS::Create()

{

int i;

char ch='y';

fstream seqfile;

fstream indexfile;

i=0;

indexfile.open("IND.txt",ios::in | ios::out | ios::binary);

seqfile.open("EMP.txt",ios::in | ios::out | ios::binary);

do

{

cout<<"\n Enter Name";

cin>>Records.name;

cout<<"\n Enter Emp\_ID:";

cin>>Records.emp\_id;

cout<<"\n Enter Salary";

cin>>Records.salary;

cout<<"\n Enter Designation:";

cin>>Records.desig;

seqfile.write((char\*)& Records,sizeof(Records))<<flush;

Ind\_Records.emp\_id=Records.emp\_id;

Ind\_Records.position=i;

indexfile.write((char\*)& Ind\_Records,sizeof(Ind\_Records))<<flush;

i++;

cout<<"\n Do you want to add more records?";

cin>>ch;

}while(ch=='y');

seqfile.close();

indexfile.close();

}

void EMP\_CLASS::Display()

{

fstream seqfile;

fstream indexfile;

int i;

seqfile.open("EMP.txt",ios::in | ios::out | ios::binary);

indexfile.open("IND.txt", ios::in | ios::out | ios::binary);

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

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

cout<<"\n The Contents of file are..."<<endl;

i=0;

while(indexfile.read((char\*)&Ind\_Records,sizeof(Ind\_Records)))

{

i=Ind\_Records.position\*sizeof(Rec);

seqfile.seekg(i,ios::beg);

seqfile.read((char\*)& Records,sizeof(Records));

if(Records.emp\_id!=-1)

{

cout<<"\n Name:"<<Records.name;

cout<<"\n Emp\_ID:"<<Records.emp\_id;

cout<<"\n Salary:"<<Records.salary;

cout<<"\n Designation:"<<Records.desig;

cout<<"\n";

}

}

seqfile.close();

indexfile.close();

}

void EMP\_CLASS::Update()

{

int pos,id;

char New\_name[10];

char New\_desig[10];

int New\_salary;

cout<<"\n For Updation";

cout<<"\n Enter the Emp\_ID for searching";

cin>>id;

fstream seqfile;

fstream indexfile;

seqfile.open("EMP.txt",ios::in | ios::out | ios::binary);

indexfile.open("IND.txt",ios::in | ios::out | ios::binary);

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

pos=-1;

while(indexfile.read((char\*)&Ind\_Records,sizeof(Ind\_Records)))

{

if(id==Ind\_Records.emp\_id)

{

pos=Ind\_Records.position;

break;

}

}

if(pos==-1)

{

cout<<"\n The record is not present in the file ";

return;

}

else

{

cout<<"\n Enter the values for Updation...";

cout<<"\n Name:"; cin>>New\_name;

cout<<"\n Salary:"; cin>>New\_salary;

cout<<"\n Designation:"; cin>>New\_desig;

int offset=pos\*sizeof(Rec);

seqfile.seekp(offset);

strcpy(Records.name,New\_name);

Records.emp\_id=id;

Records.salary=New\_salary;

seqfile.write((char\*)&Records,sizeof(Records))<<flush;

cout<<"\n The record is updated";

}

seqfile.close();

indexfile.close();

}

void EMP\_CLASS::Delete()

{

int id, pos;

cout<<"\n For deletion";

cout<<"\nEnter the Emp\_ID for searching";

cin>>id;

fstream seqfile;

fstream indexfile;

seqfile.open("EMP.txt",ios::in|ios::out|ios::binary);

indexfile.open("IND.txt",ios::in|ios::out|ios::binary);

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

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

pos=-1;

while(indexfile.read((char\*)&Ind\_Records,sizeof(Ind\_Records)))

{

if(id==Ind\_Records.emp\_id)

{

pos=Ind\_Records.position;

Ind\_Records.emp\_id=-1;

break;

}

}

if(pos==-1)

{

cout<<"\nThe record is not present in the file";

return;

}

int offset=pos\*sizeof(Rec);

seqfile.seekp(offset);

strcpy(Records.name,"");

Records.emp\_id=-1;

Records.salary=-1;

strcpy(Records.desig,"");

seqfile.write((char\*)&Records,sizeof(Records))<<flush;

offset=pos\*sizeof(Ind\_Rec);

indexfile.seekg(offset);

Ind\_Records.emp\_id=-1;

Ind\_Records.position=pos;

indexfile.write((char\*)&Ind\_Records,sizeof(Ind\_Records))<<flush;

seqfile.seekg(0);

indexfile.close();

seqfile.close();

seqfile.close();

cout<<"\nThe record id Deleted!!!";

}

void EMP\_CLASS::Append()

{

fstream seqfile;

fstream indexfile;

int pos;

indexfile.open("IND.txt",ios::in|ios::binary);

indexfile.seekg(0,ios::end);

pos=indexfile.tellg()/sizeof(Ind\_Records);

indexfile.close();

indexfile.open("IND.txt",ios::app|ios::binary);

seqfile.open("EMP.txt",ios::app | ios::binary);

cout<<"\n Enter the record for appending";

cout<<"\n Name"; cin>>Records.name;

cout<<"\nEmp\_ID:"; cin>>Records.emp\_id;

cout<<"\nSalary:"; cin>>Records.salary;

cout<<"\nDesignation:"; cin>>Records.desig;

seqfile.write((char\*)&Records,sizeof(Records));

Ind\_Records.emp\_id=Records.emp\_id;

Ind\_Records.position=pos;

indexfile.write((char\*)&Ind\_Records,sizeof(Ind\_Records))<<flush;

seqfile.close();

indexfile.close();

cout<<"\nThe record is Appended!!!";

}

void EMP\_CLASS::search()

{

fstream seqfile;

fstream indexfile;

int id,pos,offset;

cout<<"\nEnter the Emp\_ID for searching the record";

cin>>id;

indexfile.open("IND.txt",ios::in|ios::binary);

pos=-1;

while(indexfile.read((char\*)&Ind\_Records,sizeof(Ind\_Records)))

{

if(id==Ind\_Records.emp\_id)

{

pos=Ind\_Records.position;

break;

}

}

if(pos==-1)

{

cout<<"\nRecord is not present in the file";

return;

}

offset=pos\*sizeof(Records);

seqfile.open("EMP.txt",ios::in|ios::binary);

seqfile.seekg(offset,ios::beg);

seqfile.read((char\*)&Records,sizeof(Records));

if(Records.emp\_id==-1)

{

cout<<"\nRecord is not present in the file";

return;

}

else

{cout<<"\nThe record is present in the file and it is...";

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

cout<<"\nEmp\_ID:"<<Records.emp\_id;

cout<<"\nSalary:"<<Records.salary;

cout<<"\nSalary:"<<Records.desig;

}

seqfile.close();

indexfile.close();

}

int main()

{

EMP\_CLASS List;

char ans='y';

int choice;

do

{

cout<<"\n Main Menu"<<endl;

cout<<"\n 1.Create";

cout<<"\n 2.Display";

cout<<"\n 3.Update";

cout<<"\n 4.Delete";

cout<<"\n 5.Append";

cout<<"\n 6.Search";

cout<<"\n 7.Exit";

cout<<"\n Enter your choice";

cin>>choice;

switch(choice)

{

case 1: List.Create();

break;

case 2: List.Display();

break;

case 3: List.Update();

break;

case 4:List.Delete();

break;

case 5:List.Append();

break;

case 6: List.search();

break;

}

cout<<"\n Do you want to go back to back to Main Menu";

cin>>ans;

}while(ans=='y');

return 0;

}