//empsorting.h

#include<iostream>

using namespace std;

class EmpSorting

{

private:

int empid;

char name[20];

double salary;

public:

EmpSorting();

EmpSorting(int, const char \*, double);

void Accept();

void Display();

void bubblesort(EmpSorting \*,int);

};

//empsorting.cpp

#include"empsorting.h"

EmpSorting::EmpSorting()

{

this->empid = 0;

strcpy(this->name, "null");

this->salary = 0.0;

}

EmpSorting::EmpSorting(int id, const char \*name, double sal)

{

this->empid = id;

strcpy(this->name,name);

this->salary = sal;

}

void EmpSorting::Accept()

{

cout << "Enter empid ;";

cin >> this->empid;

cout << "Enter name :";

cin >> this->name;

cout << "Enter salary :";

cin >> this->salary;

}

void EmpSorting::Display()

{

cout << "EmpId :" << this->empid << endl;

cout << "Name :" << this->name << endl;

cout << "Salary :" << this->salary << endl;

}

void EmpSorting:: bubblesort(EmpSorting \*emp,int n)

{

EmpSorting temp;

for (int i = 0; i < n - 1; i++)

{

for (int j = 0; j < n -i- 1; j++)

{

if (emp[j].salary > emp[j + 1].salary)

{

temp = emp[j];

emp[j] = emp[j + 1];

emp[j + 1] = temp;

}

}

}

}

void EmpSorting::selectionsort(EmpSorting \*arr, int n)

{

EmpSorting temp;

for (int i = 0; i < n - 1; i++)

{

for (int j = i + 1; j < n;j++)

{

int r = strcmp(arr[i].name, arr[j].name);

if (r > 0)

{

temp = arr[i];

arr[i] = arr[j];

arr[j] = temp;

}

}

}

}

//main.cpp

#include"empsorting.h"

#include<conio.h>

int menu\_list()

{

int choice;

cout << "0.Exit" << endl;

cout << "1.Accept" << endl;

cout << "2.Display" << endl;

cout << "3.Bubble sort" << endl;

cout << "4.Selection sort" << endl;

cout << "Enter the choice :";

cin >> choice;

return choice;

}

int main()

{

int choice,n;

EmpSorting sort[3];

while ((choice = menu\_list()) != 0)

{

switch (choice)

{

case 0:exit(0);

case 1:

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

{

sort[i].Accept();

}

break;

case 2:

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

{

sort[i].Display();

}

break;

case 3:

sort->bubblesort(sort, 3); //Non member function so used arrow->

cout << "Data is sorted...." << endl;

break;

case 4:

sort->selectionsort(sort, 3);

cout << "selection sort...." << endl;

break;

}

}

\_getch();

return 0;

}