#include <iostream>

#include <cstring>

Using namespace std;

Class PersonAdditionalInfo {

Char address[20], license[20], insurance[20];

Long int contact;

Public:

PersonAdditionalInfo() {

Strcpy(address, “XYZ”);

Strcpy(license, “XY-0000000000”);

Strcpy(insurance, “XY00000000X”);

Contact = 0;

}

~PersonAdditionalInfo() {

Cout << “I am in Destructor”;

}

Friend class Person; // Declare Friend class

};

Class Person {

Char name[20], dob[10], blood[10];

Float ht, wt;

Static int count; // Static variable

PersonAdditionalInfo\* pai;

Public:

Person() {

Strcpy(name, “XYZ”);

Strcpy(dob, “dd/mm/yy”);

Strcpy(blood, “A +”);

Ht = 0;

Wt = 0;

Pai = new PersonAdditionalInfo;

}

Person(Person\* p1) {

Strcpy(name, p1->name);

Strcpy(dob, p1->dob);

Strcpy(blood, p1->blood);

Ht = p1->ht;

Wt = p1->wt;

Pai = new PersonAdditionalInfo;

Strcpy(pai->address, p1->pai->address);

Strcpy(pai->license, p1->pai->license);

Strcpy(pai->insurance, p1->pai->insurance);

Pai->contact = p1->pai->contact;

}

Static void ShowCount() {

Cout << “\nNo of records count=” << count << “\n”;

}

~Person() {

Cout << “\nI am in Destructor\n”;

}

Void GetData(char name[20]);

Inline void Display();

};

Void Person::GetData(char name[20]) {

Strcpy(this->name, name);

Cout << “\n Enter date of birth: “;

Cin >> dob;

Cout << “\n Enter blood group: “;

Cin >> blood;

Cout << “\n Enter height: “;

Cin >> ht;

Cout << “\n Enter weight: “;

Cin >> wt;

Cout << “\n Enter address: “;

Cin >> pai->address;

Cout << “\n Enter License number: “;

Cin >> pai->license;

Cout << “\n Enter Insurance policy number: “;

Cin >> pai->insurance;

Cout << “\n Enter Contact number: “;

Cin >> pai->contact;

Count++;

}

Void Person::Display() {

Cout << “\t” << name;

Cout << “\t” << dob;

Cout << “\t” << blood;

Cout << “\t” << ht;

Cout << “\t” << wt;

Cout << “\t” << pai->address;

Cout << “\t” << pai->license;

Cout << “\t” << pai->insurance;

Cout << “\t” << pai->contact;

}

Int Person::count;

Int main() {

Person\* p1, \* p2;

Int ch;

P1 = new Person; // Call default constructor & dynamic memory allocation

P2 = new Person(p1); // Call copy constructor

Cout << “\tName”;

Cout << “\tDob”;

Cout << “\tBlood”;

Cout << “\tHt”;

Cout << “\tWt”;

Cout << “\tAddress”;

Cout << “\tLicense”;

Cout << “\tInsurance”;

Cout << “\tContact”;

Cout << endl;

Cout << “Default Constructor Value \n”;

P1->Display();

Cout << “\n”;

Cout << “Copy Constructor Value \n”;

P2->Display();

Int n;

Cout << “\nEnter how many records you want: “;

Cin >> n;

Person p3[n]; // Array of objects

Char name[20];

Int x = 0;

Do {

Cout << “\nWelcome to Personal database system”;

Cout << “\n1. Enter the record”;

Cout << “\n2. Display the record”;

Cout << “\n3. Exit”;

Cin >> ch;

Switch (ch) {

Case 1: {

Cout << “\nEnter the Name: “;

Cin >> name;

P3[x].GetData(name);

Person::ShowCount(); // Calls static function

X++;

Break;

}

Case 2: {

Cout << “\tName”;

Cout << “\tDob”;

Cout << “\tBlood”;

Cout << “\tHt”;

Cout << “\tWt”;

Cout << “\tAddress”;

Cout << “\tLicense”;

Cout << “\tInsurance”;

Cout << “\tContact”;

For (int I = 0; I < n; i++) {

Cout << “\n”;

P3[i].Display(); // Calls inline function

}

Break;

}

}

} while (ch != 3);

Delete p1; // Dynamic memory de-allocation

Delete p2;

Return 0;

}