**Person.h:**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Person Class Declaration

\* Written by Laith Assaf

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <iostream>

#include <string>

using namespace std;

class person

{

public:

person(); // Null constructor

bool get(istream &in); // Input first name, last name, and age

void put(ostream &out) const; // output first name, last name, and age

// Comparison operators

bool operator>(const person& other) const; // Greater than

bool operator<(const person& other) const; // Less than

bool operator==(const person& other) const; // Equal to

private:

string first\_name; //first name

string last\_name; //last name

int age; //age

};

**Person.cpp:**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Person Class Definitions

\* Written by Laith Assaf

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <iostream>

#include <iomanip>

#include <string>

using namespace std;

#include "Person.h"

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Constructor - no parameters

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

person::person()

{

first\_name = "";

last\_name = "";

age = 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* get()

\* Input first name, last name, and age from stream

\* Return: true if successful, false if failed

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

bool person::get(istream &in)

{

in >> first\_name >> last\_name >> age;

return(in.good());

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* put()

\* Output last name, first name, and age to stream

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void person::put(ostream &out) const

{

out << left << setw(12) << last\_name // Last name field: 12 characters

<< " " // 2 spaces after last name

<< setw(8) << first\_name // First name field: 8 characters

<< " " // 2 spaces after first name

<< right << setw(3) << age; // Age field: 3 characters

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* operator>

\* Compare if greater than another person

\* Return: true if greater than, false otherwise

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

bool person::operator>(const person& other) const

{

if (last\_name != other.last\_name)

return last\_name > other.last\_name;

if (first\_name != other.first\_name)

return first\_name > other.first\_name;

return age > other.age;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* operator<

\* Compare if less than another person

\* Return: true if less than, false otherwise

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

bool person::operator<(const person& other) const

{

if (last\_name != other.last\_name)

return last\_name < other.last\_name;

if (first\_name != other.first\_name)

return first\_name < other.first\_name;

return age < other.age;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* operator==

\* Compare if equal to another person

\* Return: true if equal to, false otherwise

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

bool person::operator==(const person& other) const

{

return (last\_name == other.last\_name &&

first\_name == other.first\_name &&

age == other.age);

}

**Main.cpp**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Program 01

\* Written by Laith Assaf

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

#include "Person.h"

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* sortArray()

\* Sorts an array of people using bubble sort

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void sortArray(person arr[], int size)

{

bool swapped = true; // Flag to indicate when a swap occurs

person temp; // Temporary person object for swapping

for (int i = 0; i < size - 1 && swapped; i++)

{

swapped = false;

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

{

if (arr[j] > arr[j + 1])

{

temp = arr[j];

arr[j] = arr[j + 1];

arr[j + 1] = temp;

swapped = true;

}

}

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* displayArray()

\* Displays an array of people

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void displayArray(const person arr[], int size)

{

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

{

arr[i].put(cout);

cout << endl;

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* main()

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int main()

{

const int MAX\_SIZE = 20; // Maximum number of people

person people[MAX\_SIZE]; // Array of people

int count\_of\_people = 0; // Counter for number of people read

string filename; // Name of input file

ifstream fin; // File input stream

// Prompt for file

cout << "Enter file name: ";

cin >> filename;

// Open file

fin.open(filename);

// Check for error

if (!fin)

{

cout << "Error opening file " << filename << endl;

return 1;

}

// Read data into array using loop and get function

while (count\_of\_people < MAX\_SIZE && people[count\_of\_people].get(fin))

count\_of\_people++;

// Close file

fin.close();

//Sort array using bubble sort

sortArray(people, count\_of\_people);

//Display header

cout << "\nLast First Age\n";

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

// Display array using display function

displayArray(people, count\_of\_people);

// Success

return 0;

}

**Output:**

base) laithassaf@Laiths-MacBook-Air-2 Program01 % ./program

Enter file name: Family.txt

Last First Age

------------ -------- ---

Bowman David 45

Bowman Frank 37

Bowman John 30

Bowman Mark 13

Bowman Mark 42

Bowman Richard 47

Christensen Ann 70

Cox Susan 36

Gueller Kathleen 34

Morales Carlos 68