**Person.h:**

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

\* Person Class Declaration

\* Written by Laith Assaf

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

class person

{

private:

string lastName; // Person's last name

string firstName; // Person's first name

int age; // Person's age

int id; // Person's unique identifier

double gpa; // Person's grade point average

public:

// Default constructor - initializes person with empty values

person();

// Parameterized constructor - creates person with given values

person(string last, string first, int age, int id, double gpa);

// Overloaded stream insertion operator for output

friend ostream& operator<<(ostream& out, const person& p);

// Overloaded stream extraction operator for input

friend istream& operator>>(istream& in, person& p);

// Overloaded equality operator to compare person with an ID

bool operator==(int searchId) const;

// Overloaded less than operator to compare two persons

bool operator<(const person& other) const;

// Returns the person's GPA

double getGPA() const;

// Returns the person's age

int getAge() const;

};

**People.h**

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

\* People Class Declaration

\* Written by Laith Assaf

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

#include "Person.h"

class people

{

private:

person\* map; // Pointer to dynamic array of person objects

int len; // Number of persons currently in the array

static const int MAX\_SIZE = 20; // Maximum size of the array

public:

// Default constructor - creates empty array of person objects

people();

// Destructor - deallocates dynamic memory

~people();

// Inserts a new person into the array in ascending order

// Returns true if successful, false if array is full

bool insert(const person& p);

// Displays all persons in the array with statistics

// Shows number of persons, average age and GPA

void display(ostream& out) const;

// Finds a person by ID in the array

// Returns true if found, false otherwise

bool find(int searchId) const;

// Removes a person with the given ID from the array

// Returns true if successful, false if not found

bool remove(int searchId);

};