IN\_LAP

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

// Workshop 4: Constructors

// File name : Passenger.h

// Date : Oct 04, 2017

// Author : Eunhee Kim

//

// Description

// Class definaition and function declarations

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

#ifndef SICT\_PASSENGER\_H

#define SICT\_PASSENGER\_H

namespace sict{

        class Passenger {

                private:

                char passName[32];

                char dest[32];

                public:

                Passenger();

                Passenger(const char\* p\_name, const char\* p\_dest);

                bool isEmpty() const;

                void display() const;

        };

}

#endif

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

// Workshop 4: Constructors

// File name : Passenger.cpp

// Date : Oct 04, 2017

// Author : Eunhee Kim

//

// Description

// Set an object to a safe empty state and

// display the contents of the Passenger object

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

#include <iostream>

#include <cstring>

#include "Passenger.h"

using namespace std;

namespace sict{

        Passenger::Passenger(){

                passName[0] = '\0';

        }

        Passenger::Passenger(const char\* p\_name, const char\* p\_dest){

                if(p\_name == nullptr || strlen(p\_name) == 0){

                        passName[0] = '\0';

                }else if(p\_dest == nullptr || strlen(p\_dest) == 0){

                        passName[0] = '\0';

                }else{

                        strcpy(passName, p\_name);

                        strcpy(dest, p\_dest);

                }

        }

        bool Passenger::isEmpty() const{

                if(passName[0] == '\0'){

                        return true;

                }else{

                        return false;

                }

        }

        void Passenger::display() const{

                if(isEmpty()){

                        cout << "No passenger!" << endl;

                }else{

                        cout << passName << " - " << dest << endl;

                }

        }

}

AT\_HOME

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

// Workshop 4: Constructors

// File name : Passenger.h

// Date : Oct 05, 2017

// Author : Eunhee Kim

//

// Description

// Class definaition and function declarations

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

#ifndef SICT\_PASSENGER\_H

#define SICT\_PASSENGER\_H

namespace sict{

        class Passenger {

                private:

                char passName[32];

                char dest[32];

                int year;

                int month;

                int day;

                public:

                Passenger();

                Passenger(const char\* p\_name, const char\* p\_dest);

                Passenger(const char\* p\_name, const char\* p\_dest, int p\_year, int p\_month, int p\_day);

                const char\* name() const;

                bool canTravelWith(const Passenger&) const;

                bool isEmpty() const;

                void display() const;

        };

}

#endif

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

// Workshop 4: Constructors

// File name : Passenger.cpp

// Date : Oct 05, 2017

// Author : Eunhee Kim

//

// Description

// Set an object to a safe empty state, adding date

// information of the Passenger class and

// display the contents of the Passenger object

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

#include <iostream>

#include <cstring>

#include <iomanip>

#include "Passenger.h"

using namespace std;

namespace sict{

        Passenger::Passenger(){

                passName[0] = '\0';

                dest[0] = '\0';

                year = 0;

                month = 0;

                day = 0;

        }

        Passenger::Passenger(const char\* p\_name, const char\* p\_dest){

                if(p\_name == nullptr || strlen(p\_name) == 0){

                        passName[0] = '\0';

                }else if(p\_dest == nullptr || strlen(p\_dest) == 0){

                        passName[0] = '\0';

                }else{

                        if(strlen(p\_name) > 31){

                                strncpy(passName, p\_name, 31);

                                passName[31] = '\0';

                        }else{

                                strncpy(passName, p\_name, strlen(p\_name));

                                passName[strlen(p\_name)] = '\0';

                        }

                        if(strlen(p\_dest) > 31){

                                strncpy(dest, p\_dest, 31);

                                dest[31] = '\0';

                        }else{

                                strncpy(dest, p\_dest, strlen(p\_dest));

                                dest[strlen(p\_dest)] = '\0';

                        }

                        year = 2017;

                        month = 7;

                        day = 1;

                }

        }

        Passenger::Passenger(const char\* p\_name, const char\* p\_dest, int p\_year, int p\_month, int p\_day){

                if(p\_name == nullptr || strlen(p\_name) == 0){

                        passName[0] = '\0';

                }else if(p\_dest == nullptr || strlen(p\_dest) == 0){

                        passName[0] = '\0';

                }else if(p\_year < 2017 || p\_year > 2020){

                        passName[0] = '\0';

                }else if(p\_month < 1 || p\_month > 12){

                        passName[0] = '\0';

                }else if(p\_day < 1 || p\_day > 31){

                        passName[0] = '\0';

                }else{

                       if(strlen(p\_name) > 31){

                                strncpy(passName, p\_name, 31);

                                passName[31] = '\0';

                        }else{

                                strncpy(passName, p\_name, strlen(p\_name));

                                passName[strlen(p\_name)] = '\0';

                        }

                        if(strlen(p\_dest) > 31){

                                strncpy(dest, p\_dest, 31);

                                dest[31] = '\0';

                        }else{

                                strncpy(dest, p\_dest, strlen(p\_dest));

                                dest[strlen(p\_dest)] = '\0';

                        }

                        year = p\_year;

                        month = p\_month;

                        day = p\_day;

                }

        }

        const char\* Passenger::name() const{

                if(!isEmpty()){

                        return passName;

                }else{

                        return nullptr;

                }

        }

        bool Passenger::canTravelWith(const Passenger& p) const{

                if(year == p.year && month == p.month && day == p.day){

                        return true;

                }else{

                        return false;

                }

        }

       bool Passenger::isEmpty() const{

                if(passName[0] == '\0'){

                        return true;

                }else{

                        return false;

                }

        }

        void Passenger::display() const{

                if(isEmpty()){

                        cout << "No passenger!" << endl;

                }else{

                        cout << passName << " - " << dest << " on " << year << "/" << setw(2) << setfill('0') << month

                        << "/" << setw(2) << setfill('0') << day << endl;

                }

        }

}