// =======================

// Included: Dog.h, Source.cpp

// =======================

// HW 9f

// =======================

// Christian Falucho

// CMPR 121

// =======================

/\*================ CONTENTS FROM Dog.h ===================

#include <iostream>

#include <string>

using namespace std;

class Dog {

private:

string name;

float weight;

int age;

public:

Dog(string name, float weight, int age){

this->name = name;

this->weight = weight;

this->age = age;

};

~Dog(){};

void displayDog(){

cout << "Name: " << name << endl;

cout << "Weight: " << weight << endl;

cout << "Age: " << age << endl;

};

bool operator>=(int age){

return (this->age >= age) ? true : false;

};

bool operator<(Dog& dog) const {

return (this->weight < dog.weight) ? true : false;

};

bool operator==(Dog& dog) const{

return (this->name == dog.name) ? true : false;

};

friend ostream & operator<<(ostream& stream, Dog& dog){

stream << "Name: " << dog.name << endl

<< "Weight: " << dog.weight << " pounds." << endl

<< "Age: " << dog.age << " years old." << endl;

return stream;

};

};

/\*================ CONTENTS FROM Dog.h ===================\*/

/\*================ CONTENTS FROM Source.cpp ===================

#include <iostream>

#include <string>

using namespace std;

class Dog {

private:

string name;

float weight;

int age;

public:

Dog(string name, float weight, int age){

this->name = name;

this->weight = weight;

this->age = age;

};

~Dog(){};

void displayDog(){

cout << "Name: " << name << endl;

cout << "Weight: " << weight << endl;

cout << "Age: " << age << endl;

};

bool operator>=(int age){

return (this->age >= age) ? true : false;

};

bool operator<(Dog& dog) const {

return (this->weight < dog.weight) ? true : false;

};

bool operator==(Dog& dog) const{

return (this->name == dog.name) ? true : false;

};

friend ostream & operator<<(ostream& stream, Dog& dog){

stream << "Name: " << dog.name << endl

<< "Weight: " << dog.weight << " pounds." << endl

<< "Age: " << dog.age << " years old." << endl;

return stream;

};

};

/\*================ CONTENTS FROM Source.cpp ===================

\*/

/\*

===========================================================

=== CODE OUTPUT ===

===========================================================

\*/

A screenshot of a computer

AI-generated content may be incorrect.

/\*

===========================================================

=== CODE OUTPUT ===

===========================================================

\*/

// =======================

// Included: Car.h, Car.cpp, Source.cpp

// =======================

// HW 9f\_1

// =======================

// Christian Falucho

// CMPR 121

// =======================

/\*================ CONTENTS FROM Car.h ===================\*/

#include <iostream>

#include <string>

using namespace std;

class Car{

private:

string model;

int year;

static int carCount;

public:

Car();

Car(string, int);

~Car();

void setCar(string, int);

int getCount();

void displayCar();

friend bool areSame(Car&, Car&);

};

/\*================ CONTENTS FROM Car.h ===================\*/

/\*================ CONTENTS FROM Car.cpp ===================\*/

#include <iostream>

#include <string>

#include "Car.h"

Car::Car(){

model = "\0";

year = 0;

carCount++;

}

Car::Car(string model, int year){

this->model = model;

this->year = year;

carCount++;

}

Car::~Car(){}

void Car::setCar(string model, int year){

this->model = model;

this->year = year;

}

int Car::carCount = 0;

int Car::getCount(){

return carCount;

}

void Car::displayCar(){

cout << "Model: " << model << endl;

cout << "Year: " << year << endl;

}

bool areSame(Car& car1, Car& car2){

return (car1.model == car2.model) &&

(car1.year == car2.year);

}

/\*================ CONTENTS FROM Car.cpp ===================\*/

/\*================ CONTENTS FROM Source.cpp ===================\*/

#include <iostream>

#include <string>

#include "Car.h"

using namespace std;

int main(){

Car myCar;

Car yourCar("Toyota", 2025);

cout << "My Car\n";

myCar.displayCar();

cout << endl;

cout << "Your Car\n";

yourCar.displayCar();

cout << endl;

myCar.setCar("Ford", 2002);

cout << "My Car\n";

myCar.displayCar();

cout << endl;

if (areSame(myCar, yourCar))

{

cout << "The two cars are the same.\n";

}else{

cout << "The two cars are not the same.\n";

}

cout << endl;

cout << myCar.getCount() << " have been declared.\n";

return 0;

/\*================ CONTENTS FROM Source.cpp ===================\*/

/\*

===========================================================

=== CODE OUTPUT ===

===========================================================

\*/

A screen shot of a computer

AI-generated content may be incorrect.

/\*

===========================================================

=== CODE OUTPUT ===

===========================================================

\*/