```
#include <iostream>
#include <fstream>
#include "Agency.h"

using namespace std;

Car::Car()
{
	make = new char[20];
	model = new char[20];
	year = 0;
	price = 0.00;
	available = false;
}

Car::Car(char * newMake, char * newModel, int newYear, float newPrice, bool availability)
{
	//Make
	make = new char[20];
	char * makeO = make;
	while(*newMake != '\0')
	{
		*make = *newMake;
		make++;
		newMake++;
	}
	*make = '\0';
	make = makeO;

	//Model
	model = new char[20];
	char * modelO = model;
	while(*newModel != '\0')
	{
		*model = *newModel;
		model++;
		newModel++;
	}
	*model = '\0';
	model = modelO;

	year = newYear;
	price = newPrice;
	available = availability;
}

Car::Car(const Car& car)
{
	char * makeptr = car.make;
```

```
	char * modelptr = car.model;

	//Make
	make = new char[20];
	char * makeO = make;
	while(*makeptr != '\0')
	{
		*make = *makeptr;
		make++;
		makeptr++;
	}
	*make = '\0';
	make = makeO;

	//Model
	model = new char[20];
	char * modelO = model;
	while(*modelptr != '\0')
	{
		*model = *modelptr;
		model++;
		modelptr++;
	}
	*model = '\0';
	model = modelO;

	year = car.year;
	price = car.price;
	available = car.available;
}

Car::~Car()
{
	delete[] make;
	delete[] model;

	make = NULL;
	model = NULL;
}

void Car::copy(Car car)
{
	char * makeptr = car.make;
	char * modelptr = car.model;

	//Make
	char * makeO = make;
	while(*makeptr != '\0')
	{
```

```
                *make = *makeptr;
                make++;
                makeptr++;
        }
        *make = '\0';
        make = makeO;

        //Model
        char * modelO = model;
        while(*modelptr != '\0')
        {
                *model = *modelptr;
                model++;
                modelptr++;
        }
        *model = '\0';
        model = modelO;

        year = car.year;
        price = car.price;
        available = car.available;
}

void Car::print() const
{
        cout << make << " ";
        cout << model << " ";
        cout << year << " ";
        cout << "$" << price << " per day ";
        cout << "Available: " << boolalpha << available << " ";
}

char * Car::getMake() const
{
        return make;
}

char * Car::getModel() const
{
        return model;
}

int Car::getYear() const
{
        return year;
}

float Car::getPrice() const
{
```

```
	return price;
}

bool Car::getAvailable() const
{
	return available;
}

void Car::setMake(char * newMake)
{
	char * makeO = make;
	while(*newMake != '\0')
	{
		*make = *newMake;
		make++;
		newMake++;
	}
	*make = '\0';
	make = makeO;
}

void Car::setModel(char * newModel)
{
	char * modelO = model;
	while(*newModel != '\0')
	{
		*model = *newModel;
		model++;
		newModel++;
	}
	*model = '\0';
	model = modelO;
}

void Car::setYear(int newYear)
{
	year = newYear;
}

void Car::setPrice(float newPrice)
{
	price = newPrice;
}

void Car::setAvailable(bool newAvailable)
{
	available = newAvailable;
}
```

```
Agency::Agency()
{
	name = new char [20];
	zipcode = new int [5];
	inventory = new Car[15];
}

Agency::Agency(const Agency& agency)
{
	name = new char [20];
	zipcode = new int [5];
	inventory = new Car[15];

	char * nameptr = agency.name;
	int * zipptr = agency.zipcode;
	Car * inventoryptr = agency.inventory;

	char * nameO = name;
	int * zipO = zipcode;
	Car * inventoryO = inventory;

	while(*nameptr != '\0')
	{
		*name = *nameptr;
		name++;
		nameptr++;
	}

	for(int i = 0; i < 5; i++)
	{
		*zipcode = *zipptr;
		zipcode++;
		zipptr++;
	}

	for(int i = 0; i < 15; i++)
	{
		*inventory = *inventoryptr;
		inventory++;
		inventoryptr++;
	}

	name = nameO;
	zipcode = zipO;
	inventory = inventoryO;
}

Agency::~Agency()
{
```

```
	delete[] name;
	delete[] zipcode;
	delete[] inventory;

	name = NULL;
	zipcode = NULL;
	inventory = NULL;
}

void Agency::readInData(char * filename)
{
	ifstream fin;
	fin.open(filename);
	char zipchar;
	int zipint;
	int * zipO = zipcode;
	char * temp = new char[20];
	float newPrice;
	int newYear;
	bool availability;
	Car * cars = inventory;
	Car * carO = cars;

	fin >> name;

	for(int i = 0; i < 5; i++)
	{
		fin >> zipchar;
		zipint = (zipchar - 48);
		*zipcode = zipint;
		zipcode++;
	}
	zipcode = zipO;

	for(int i = 0; i < 15; i++)
	{
		fin >> newYear;
		(*cars).setYear(newYear);
		fin >> temp;
		(*cars).setMake(temp);
		fin >> temp;
		(*cars).setModel(temp);
		fin >> newPrice;
		(*cars).setPrice(newPrice);
		fin >> availability;
		(*cars).setAvailable(availability);

		cars++;
	}
```

```
	cars = carO;

	delete[] temp;

	temp = NULL;
}

void Agency::print() const
{
	cout << name << " ";

	Car * carptr = inventory;
	int * zipptr = zipcode;

	for(int i = 0; i < 5; i++)
	{
		cout << *zipptr;
		zipptr++;
	}
	zipptr = zipcode;

	cout << endl;

	for(int i = 0; i < 15; i++)
	{
		(*carptr).print();
		cout << endl;
		carptr++;
	}
	carptr = inventory;
}

void Agency::printAvailableCars() const
{
	Car * cars = inventory;
	Car * carsO = cars;
	for(int i = 0; i < 15; i++)
	{
		if((*cars).getAvailable())
		{
			(*cars).print();
			cout << endl;
		}
		cars++;
	}
	cars = carsO;
}

void Agency::findMostExpensive() const
```

```
{
	Car * cars = inventory;
	Car * carO = cars;
	int mostExpensiveNumber;
	float highestPrice = 0.0;
	for(int i = 0; i < 15; i++)
	{
		if((*cars).getPrice() > highestPrice)
		{
			mostExpensiveNumber = i;
			highestPrice = (*cars).getPrice();
		}
	}
	cars = carO;
	for(int i = 0; i < mostExpensiveNumber; i++)
	{
		cars++;
	}
	cout << "Most Expensive Car: ";
	(*cars).print();
	cout << endl;
	cars = carO;
}

float Agency::estimateCost(int carN, int days) const
{
	Car * car = inventory;
	Car * carO = car;
	float cost;
	for(int i = 0; i < (carN-1); i++)
	{
		car++;
	}
	cost = ((*car).getPrice()) * days;
	car = carO;
	return cost;
}

void Agency::sortByMake()    // alphabetical
{

}

void Agency::sortByPrice()   // highest to lowest
{

}

void Agency::searchByMake(char * makeName) const
```

{

}