**Kevin Woods**
**CS 161 – Assignment 8 Report**

## Understanding

This week's material project introduced structs and vectors and their relationship with arrays. The project carLot.cpp is a fairly complicated program that relies on most of the material from prior weeks.  We will need to create a struct for the dates, and a struct for the car variables that includes some Date structs. Next we need to create a Car vector to store the car variables in. We will display a menu to the user with the following options: add entry, list inventory, profit for month, or quit program. The add entry choice will prompt the user to enter the car variables and sale variable if it has been sold. The list inventory options will list all cars in inventory. The profit for a month option will display the sales profits for a given month (if there are any). The quit option will end the program.

## Design & Testing

Follows on the page…

```
carLot.cpp
    includes and using statements
        includes iostream, string, vector, iomanip
        using namespace std
    struct Date
        int day, month, year
    struct Car
        string make, model
        int year
        Date datePurchased, dateSold
        bool isSold
        double purchasePrice, salePrice
    bool check date
        if year < 0 || year >2015
            return false
        if month <1 || month > 12
            return false
        int days [12] = {max days in each month]
        if (day <1 || day > days[month-1]
            return false
        return true
    vector<Car> addEntry
        Car newCar
        int day, month, year

        prompt user to enter make
        accept make

        prompt user to enter model
        accept model

        prompt user to enter model year
        accept & validate year

        prompt user to enter date purchased
        acccept and validate date

        prompt user to enter purchase price
        accept purchase price

        set newCar.datePurchased = date purchased entered

        ask user if car is sold

        if sold
            prompt user to enter date of sale
            accept and validatedate of sale

            prompt user to enter sale price
            accept sale price

            set newCar.dateSold = date of sale

        else
            break

        return entry

    void List inventory (vector<Car> entry)

        int inventory = 0

        for loop through entrys

            if car is not sold

                list car

                inventory++

        print total inventory

    void profit (vector<Car> entry)

        double totalProfit = 0
        int  profit date

        promput user to enter profit date
        accept and validate profit date

        for loop through entrys
            if car sold same date
                add profit to totalProfit
        print totalProfit

    int main
        vector<Car> entry
        int choice;

        print menu
        accept menu choice

        while choice != Quit
            case 1
                entry = addEntry(entry)
                break
            case 2
                list inventory (entry)
                break
            case 3
                profit(entry)
                break
            default
                error invalid menu choice
            print menu
            accept choice
        return 0
```

**Testing**

carLot.cpp

| Prompt | Input | Expected OutPut | Actual Ouput |
|---|---|---|---|
| Add Entry (make, model, year, purchase price, day, month, year, is sold(sale day, month, year, price)), List Inventory, Profit (month, year), Quit | Add entry, (ford, taurus, 1991, 2005, 04, 26, 5000, no) | entry added, menu | entry added, menu |
| Add Entry (make, model, year, purchase price, day, month, year, is sold(sale day, month, year, price)), List Inventory, Profit (month, year), Quit | List inventory | ford, taurus, 1991, 5000 | ford, taurus, 1991, 5000 |
| Add Entry (make, model, year, purchase price, day, month, year, is sold(sale day, month, year, price)), List Inventory, Profit (month, year), Quit | add entry (toyota, prius, 2005, 2005, 04, 27, 8000, yes (2005, 04, 28, 10000) | entry added, menu | entry added, menu |
| Add Entry (make, model, year, purchase price, day, month, year, is sold(sale day, month, year, price)), List Inventory, Profit (month, year), Quit | list inventory | ford, taurus, 1991, 5000 toyota, prius, 2005, 8000 | ford, taurus, 1991, 5000 toyota, prius, 2005, 8000 |
| Add Entry (make, model, year, purchase price, day, month, year, is sold(sale day, month, year, price)), List Inventory, Profit (month, year), Quit | Profit(04, 2005) | 2000 | 2000 |
| Add Entry (make, model, year, purchase price, day, month, year, is sold(sale day, month, year, price)), List Inventory, Profit (month, year), Quit | proft(wqe, ) | infinite loop error | infinite loop error |
| Add Entry (make, model, year, purchase price, day, month, year, is sold(sale day, month, year, price)), List Inventory, Profit (month, year), Quit | 5 | invalid menu choice, menu prompt | invalid menu choice, menu prompt |

## Reflection

This week was pretty packed with the test and some fairly complicated exercises, not to mention the project! I was able to get my shopCart to work for the most part very quickly, which was quite suprising. It seems that structs were really not too difficult for me to comprehend. They are a logical addition to the C++ components we have used so far; at it was obvious right way why they may be important. findMode was a different story, I never got it working. I came to a point where I needed to focus all of my energy on carLot. Because exercise 1 was very similar, I actually put together a fairly decent carLot program. My error handling is a bit messed up. I wasn't sure of the best way to have the user input the full dates, and trying to have it all on one line may have led to my struggles. I was able to use the structs and vectors pretty well in the program, again shopCart helped quite a bit. Overall I am very happy with how my project turned out this week.