# COMP 2404
# Introduction to Software Engineering
# Assignment 3

Released: Monday February 26, 2018, 12:00 Noon

Due: Monday March 12, 2018, 12:00 Noon

## Instructions

In this assignment you will build upon the program you worked on in the previous assignment. You are free to build upon your solution or start from the skeleton code provided. If you are starting from your code, please make the following change: change the return type of the Shop::getCustomer function to a Customer pointer. Alter the body of the function to remove the dereference. This change has been made in the altered skeleton code provided. This function's use has changed slightly, it now takes a customer id and returns a pointer to the customer with that id. You will be implementing the get function called in the body of this function later in the assignment.

# Task 1: Enforcing const-ness throughout

Your first job will be to go through all of the code and decide which functions should be declared const. You should find several places throughout the program where this makes sense. We will also make the **id** data member in the **Customer** class **const**, as once a customer has been created their ID will never change. You will have to alter the code to make this so. You can not alter the parameters or return type of any function to accomplish any of this. The program should work just as it did before.

# Task 2: Introduce Linked Lists

In this part you will replace the **CustomerArray** and **VehicleArray** classes with new linked list classes. You will create new linked list classes called **CustomerList** and **VehicleList** which will hold the collection of customers and the collection of vehicles as a linked list, respectively. You will store the customer objects in alphabetical order based on their last name and you will store the vehicle object based on the year of the vehicle in descending order (ie. newest first) in their respective linked lists.

The **CustomerList** and **VehicleList** classes should:

- hold a pointer to the head of the list, but no pointer to the tail of the list
- provide an **add** function that takes a **Customer** or **Vehicle** pointer and adds the object in its correct place in the **CustomerList** or **VehicleList** class, respectively

- provide a **getSize** function that returns the size of the list. This value should not be stored, it should be calculated each time this function is called
- provide a **get** function in the **CustomerList** that takes an integer parameter (id) and returns a pointer to the **Customer** object in the list with that id. If no such object exists, return null
- manage its memory to avoid memory leaks

Notes:

- DO NOT use dummy nodes! Every node in each list must correspond to an object.
- You will modify your program to use a **CustomerList** and **VehicleList** objects instead of a **CustomerArray** and **VehicleArray** object to hold the objects.
- All classes will continue to interact with the **CustomerList** and **VehicleList** classes the same way they did with the **CustomerArray** and **VehicleArray** classes.
- Both lists will no longer have a maximum size. This means that the add functions in both list classes will no longer need to return an int as they should return void. You will have to change your add functions in the **Customer** and **Vehicle** classes to reflect this change.
- The changes to the rest of the classes should be minimal.
- The order in which your datafill is added to the list must test all cases: adding to the front, back, middle, etc.
- Your list must be stored in proper order of at all times
- You will have to update your Makefile to compile your new program.

# Task 3: Modify the “Print Customer Database” feature

Printing out the linked lists poses a new challenge in maintaining our existing design, specifically the separation of the UI and collection classes. We are not permitted to print out data from inside the collection class, since a collection class should not know how to interact with the outside world. We also cannot allow the UI class to traverse the product collection in order to print it to the screen, since that would require knowledge by the UI class of the internal configuration of the list, including the Node class, which would also violate encapsulation rules.

Our solution here will be to implement a formatting function in the **CustomerList** and **VehicleList** classes. These functions will have the prototype:

**void toString(string& outStr)**

where the **outStr** parameter is the result of the function concatenating all the object’s data formatted into one long string. This long string will contain all data found by traversing the linked list. The UI class will invoke the formatting function on the List objects, and then output the resulting formatted string to the screen. Take a look at the current printing code in the **View** as a hint on how to implement these functions. Your output does not have to be in the exact same format as before, just ensure that all of the same information is there in a readable way.

Note: **DO NOT** have your UI class traverse the linked list! **DO NOT** print to the screen from the list class!

# Task 4: Add some options to the menu

We will now add two new options to our user interface menu.

1. Option 2 on the list should now be **“Add Customer**”. When the user chooses this option, the program will prompt the use for all of the relevant information, create a customer object and add it to the Customer list. Write the appropriate functions in the view, controller, and shop classes (maintaining the overall proper design of the program) to do so.
2. Option 3 on the list will now be **“Add Vehicle”**. When the user chooses this option the program will prompt the user for a customer id. If it is invalid, tell the user (hint: use the **getCustomer** function in the shop class mentioned at the very top of the assignment). If not, prompt the user for all of the relevant information, create a vehicle object and add it to the customer’s vehicle list.

In both of these cases you need to think about which classes should do the interaction with the user, which should be creating objects, etc. You need to maintain the overall design of the program including the separation of the view, controller and entity classes. Take a look at how the print customer database option works and trace the code through the various classes to understand how these functions should be implemented.

## Constraints

- your program must not have any memory leaks (don’t worry if valgrind reports that some memory on the heap is “still reachable”)
- do not use any global variables
- your program must reuse functions everywhere possible
- your program must be thoroughly commented
- **your program must compile and run in COMP2404-2406- W18 Virtual Machine**

## Submission

You will submit in cuLearn, before the due date and time, one **tar** file that includes all the following:

- all source code, including the code provided
- a readme file that includes:
    - a preamble (program author, purpose, list of source/header/data files)
    - the exact compilation command
    - launching and operating instructions

# Grading [out of 28 marks]

**Marking components:**

- 4 marks: enforcing const-ness
    - 2 marks: const functions
    - 2 marks: const user id
- 11 marks: linked lists (6 for CustomerList, 5 for VehicleList)
    - 2 marks: overall structure
    - 1 mark: add function
    - 1 mark: getSize function
    - 1 mark: destructor
    - 1 mark: get function (CustomerList only)
- 5 marks: Print customer database
    - 2 marks for VehicleList portion
    - 3 marks for CustomerList portion
- 4 marks: Add customer
- 4 marks: Add vehicle

**Notes:**
In order to get credit for a marking component, the program must prove that the marking component executes successfully. This is usually accomplished by printing out correct data.

# Deductions

- Packaging errors:
    - 10% for missing readme
- Major programming and design errors:
    - 50% of a marking component that uses global variables
    - 50% of a marking component that consistently fails to use correct design principles, including separate functions
    - 50% of a marking component where unauthorized changes have been made to provided code or prototypes
- Minor programming errors:
    - 10% for consistently missing comments or other bad style
    - 10% for consistently failing to perform basic error checking
- Execution errors:
    - 100% of a marking component that can’t be tested because the code doesn’t compile or execute in the VM
    - 100% of a marking component that can’t be tested because the feature isn’t used in the code
    - 100% of a marking component that can’t be proven to run successfully because data is not printed out

# Sample Console Output

User input highlighted.

```
> make clean
rm -f main.o ShopController.o View.o Shop.o CustomerList.o VehicleList.o Customer.o
Vehicle.o mechanicshop
> make
g++ -c main.cc
g++ -c ShopController.cc
g++ -c View.cc
g++ -c Shop.cc
g++ -c CustomerList.cc
g++ -c VehicleList.cc
g++ -c Customer.cc
g++ -c Vehicle.cc
g++ -o mechanicshop main.o ShopController.o View.o Shop.o CustomerList.o VehicleList.o
Customer.o Vehicle.o
> ./mechanicshop



          **** Toby's Auto Mechanic Information Management System ****

                                  MAIN MENU

         1. Print Customer Database

         2. Add Customer

         3. Add Vehicle

         0. Exit

Enter your selection:  1

CUSTOMERS:

Customer ID 1001

    Name:                           Abigail Atwood
    Address:                        43 Carling Dr.
    Phone Number:                    (613)345-6743

    1 vehicle(s):

         Green 2016   Subaru Forester (40000km)
```

Customer ID 1002

    Name:                          Brook Banding
    Address:                     1 Bayshore Dr.
    Phone Number:                (613)123-7456

    2 vehicle(s):

        White 2018      Honda Accord (5000km)
        White 1972 Volkswagon Beetle (5000km)

Customer ID 1004

    Name:                            Eve Engram
    Address:                    75 Bronson Ave.
    Phone Number:                (613)456-2345

    3 vehicle(s):

         Blue 2017      Toyota Prius (10000km)
         Gold 2015       Toyota Rav4 (20000km)
        Green 2013    Toyota Corolla (80000km)

Customer ID 1003

    Name:                            Ethan Esser
    Address:                    245 Rideau St.
    Phone Number:                (613)234-9677

    1 vehicle(s):

        Black 2010     Toyota Camery (50000km)

Customer ID 1000

    Name:                          Maurice Mooney
    Address:                2600 Colonel By Dr.
    Phone Number:                (613)728-9568

    1 vehicle(s):

          Red 2007       Ford Fiesta (100000km)

Customer ID 1005

```
    Name:                      Victor Vanvalkenburg
    Address:                       425 O'Connor St.
    Phone Number:                     (613)432-7622

    4 vehicle(s):

         Black 2016       GM Escalade (40000km)
           Red 2015         GM Malibu (20000km)
        Purple 2012          GM Envoy (60000km)
        Orange 2012    GM Trailblazer (90000km)



Press enter to continue...


          **** Toby's Auto Mechanic Information Management System ****

                                MAIN MENU

        1. Print Customer Database

        2. Add Customer

        3. Add Vehicle

        0. Exit

Enter your selection:  2

First name: Andrew
Last name: Zordon
Address: 123 Forest Ave.
Phone number: (416)555-2345

Press enter to continue...


          **** Toby's Auto Mechanic Information Management System ****

                                MAIN MENU

        1. Print Customer Database

        2. Add Customer

        3. Add Vehicle

        0. Exit
```

```
Enter your selection:  1

CUSTOMERS:

Customer ID 1001

    Name:                          Abigail Atwood
    Address:                       43 Carling Dr.
    Phone Number:                   (613)345-6743

    1 vehicle(s):

         Green 2016   Subaru Forester (40000km)


Customer ID 1002

    Name:                           Brook Banding
    Address:                       1 Bayshore Dr.
    Phone Number:                   (613)123-7456

    2 vehicle(s):

         White 2018      Honda Accord (5000km)
         White 1972 Volkswagon Beetle (5000km)


Customer ID 1004

    Name:                              Eve Engram
    Address:                      75 Bronson Ave.
    Phone Number:                    (613)456-2345

    3 vehicle(s):

          Blue 2017      Toyota Prius (10000km)
          Gold 2015       Toyota Rav4 (20000km)
         Green 2013    Toyota Corolla (80000km)


Customer ID 1003

    Name:                             Ethan Esser
    Address:                      245 Rideau St.
    Phone Number:                    (613)234-9677

    1 vehicle(s):

         Black 2010     Toyota Camery (50000km)
```

```
Customer ID 1000

    Name:                           Maurice Mooney
    Address:                  2600 Colonel By Dr.
    Phone Number:                   (613)728-9568

    1 vehicle(s):

           Red 2007       Ford Fiesta (100000km)


Customer ID 1005

    Name:                      Victor Vanvalkenburg
    Address:                       425 O'Connor St.
    Phone Number:                  (613)432-7622

    4 vehicle(s):

         Black 2016       GM Escalade (40000km)
           Red 2015         GM Malibu (20000km)
        Purple 2012          GM Envoy (60000km)
        Orange 2012    GM Trailblazer (90000km)


Customer ID 1006

    Name:                           Andrew Zordon
    Address:                      123 Forest Ave.
    Phone Number:                   (416)555-2345

Press enter to continue...


         **** Toby's Auto Mechanic Information Management System ****

                                MAIN MENU

        1. Print Customer Database

        2. Add Customer

        3. Add Vehicle

        0. Exit

Enter your selection:  3
Customer ID: 1008
```

```
Invalid choice.

Press enter to continue...


         **** Toby's Auto Mechanic Information Management System ****

                                MAIN MENU

        1. Print Customer Database

        2. Add Customer

        3. Add Vehicle

        0. Exit

Enter your selection:  3
Customer ID: 1006


Make: Ford
Model: Torus
Colour: Grey
Year: 2010
Mileage: 200000

Press enter to continue...


         **** Toby's Auto Mechanic Information Management System ****

                                MAIN MENU

        1. Print Customer Database

        2. Add Customer

        3. Add Vehicle

        0. Exit

Enter your selection:  1

CUSTOMERS:

Customer ID 1001

    Name:                           Abigail Atwood
    Address:                        43 Carling Dr.
```

Phone Number: (613)345-6743

1 vehicle(s):

Green 2016 Subaru Forester (40000km)

Customer ID 1002

Name: Brook Banding
Address: 1 Bayshore Dr.
Phone Number: (613)123-7456

2 vehicle(s):

White 2018 Honda Accord (5000km)
White 1972 Volkswagon Beetle (5000km)

Customer ID 1004

Name: Eve Engram
Address: 75 Bronson Ave.
Phone Number: (613)456-2345

3 vehicle(s):

Blue 2017 Toyota Prius (10000km)
Gold 2015 Toyota Rav4 (20000km)
Green 2013 Toyota Corolla (80000km)

Customer ID 1003

Name: Ethan Esser
Address: 245 Rideau St.
Phone Number: (613)234-9677

1 vehicle(s):

Black 2010 Toyota Camery (50000km)

Customer ID 1000

Name: Maurice Mooney
Address: 2600 Colonel By Dr.
Phone Number: (613)728-9568

1 vehicle(s):

```
          Red 2007       Ford Fiesta (100000km)


Customer ID 1005

    Name:                      Victor Vanvalkenburg
    Address:                       425 O'Connor St.
    Phone Number:                    (613)432-7622

    4 vehicle(s):

         Black 2016       GM Escalade (40000km)
           Red 2015         GM Malibu (20000km)
        Purple 2012          GM Envoy (60000km)
        Orange 2012    GM Trailblazer (90000km)


Customer ID 1006

    Name:                            Andrew Zordon
    Address:                       123 Forest Ave.
    Phone Number:                    (416)555-2345

    1 vehicle(s):

          Grey 2010        Ford Torus (200000km)


Press enter to continue...



         **** Toby's Auto Mechanic Information Management System ****

                                MAIN MENU

        1. Print Customer Database

        2. Add Customer

        3. Add Vehicle

        0. Exit

Enter your selection:  0

>
```