# Welcome to AnyCar Manufacturing Plant

Dear Student,

Great Work! Your application can now process orders, exhibits the 4 pillars of OOP, and detects and notifies users of errors. The only issue is that we are using a very flawed and basic configuration file format. Text files are notoriously easy to corrupt and incredibly difficult to use. So, we have decided to transition to XML format. Within this assignment you can see that our customers have created XML files for their configuration files. You need to update your application to read in and process the new XML format. If you need anything else, please let me know.

Thank you,
Your Supervisor
AnyCar
Manufacturing Technical Manager

# Application Requirements

You will transform your application to read and process all requests with the new XML Format. To do this you must create a helper class called ConfigParserXML that will process all requests. Finally, you will remove all getters/setters methods code to only receive and return the proper attribute. NO ADDITIONAL CODE WILL RESIDE WITHIN VEHILCE AND CHILD OBJECT GETTERS/SETTERS BESIDE ACCESS TO THE ATTRIBUTE. Make sure you update your Exceptions from Lab 3 and update the file names.

## AnyCarManufactor.java

AnyCarManufactor is the main file of your application. Main will initiate the application.

Global Variables:
`None`

### Main
Type: Public Static
Input: String Array args – Command line arguments
Return: Void
Description: `Main` Method for initiating AnyCar application. When called will call `request` to initiate the interaction with the user.

### request
Type: Public Static
Input: None
Return: Void
Description: `request` user to input PID(String) and calls `buildVehicle` to start building each vehicle. `Request` will continue to prompt the user until the user inputs -1, then request will return.

### buildVehicle
Type: Public Static
Input: String pid - product ID number
Return: Void
Description: Call ConfigParserXML, create a Vehicle object, and print Vehicle object:

# Vehicle.java

Abstract Vehicle object

Private Variables:

- `String Make`
- `String Model`
- `String[] Options`
- `String[] Parts`
- `double Price`

## Constructor
Input: String make – Vehicle Make's, String model – Vehicle's Model, String[] Options, String[] Parts, and double Price

## getMake/setMake
Getter: returns private string Make
Setter:

- Input: string make
- Operations: assigns private string make with input value

## getModel/setModel
Getter: returns private string Model
Setter:

- Input: string model
- Operations: assigns private string model with input value

## getOptions/setOptions
Getter: returns private String Array options
Setter:

- Input: String Array Options
- Operations: assigns private string options with input value

getParts/setParts
Getter: returns private String Array parts
Abstract Setter:

3

- Input: String Array Options
- Operations: assigns private String Array parts value

## getPrice/setPrice

Getter: returns private double price
Setter:
      Input: double price
      Return: void
      Description: assigns private double price with input value

## pidFormat

REMOVE

## toString

Same as Lab 1 (Car.java)

# Sadan.java

Sedan extends Vehicle. See UML Diagram

# Coupe.java

Coupe extends Vehicle. See UML Diagram

# Minivan.java

Minivan extends Vehicle. See UML Diagram

# SUV.java

SUV extends Vehicle. See UML Diagram

# Truck.java

Truck extends Vehicle. See UML Diagram

# ModelFormat.java

REMOVE

# ConfigParserXML.java

Helper class that processes requests with their XML configuration file. You must use pidFormat from lab and at least one method request. You can add any other methods you like but, you must have at least request and pidFormat.

request
Type: public
Input: String pid -> product id
Return: String Array of the Vehicle attributes
Description: Reads product ID using previously defined behavior (see below). Returns a String Array with the following behavior:
[type, make, model, options, parts, price, vehicle_option1, vehicle_option2 (if exists)]
Example 110111111111

["Sedan", "AnyCar", "ThisAutoSuperSedanB", ["Exterior Color: Blue" , "Interior Color: Leather-Tan", "Powertrain: Automatic", "Seat Type: Multi-Point", "Radio Type: AM/FM",  "Tire Size:  All-Weather", "Rim Size: 18", "Miscellaneous: CrusieControl"], ["Model: model1 $ 10000.00", "Exterior Color: Blue $ 3000.00", "Interior Color: Leather-Tan $ 5000.00", "Powertrain: Automatic $ 7500.00", "Seat Type: Multi-Point $ 800.00", "Radio Type: AM/FM $ 500.00", "Tire Size:  All-Weather $ 500.00", "Rim Size: 18 $ 1100.00", "Miscellaneous: CrusieControl  $ 2500.00", "Hatchback: No $ 0.00", "Engine: 4 $ 500.00 ], 31300.00, false, 4]


Example 130111111111

["Sedan", "AnyCar", "ThisAutoSuperMiniVanB", ["Exterior Color: Blue" , "Interior Color: Leather-Tan", "Powertrain: Automatic", "Seat Type: Multi-Point", "Radio Type: AM/FM",  "Tire Size:  All-Weather", "Rim Size: 18", "Miscellaneous: CrusieControl"], ["Model: model1 $ 10000.00", "Exterior Color: Blue $ 3000.00", "Interior Color: Leather-Tan $ 5000.00", "Powertrain: Automatic $ 7500.00", "Seat Type: Multi-Point $ 800.00", "Radio Type: AM/FM $ 500.00", "Tire Size:  All-Weather $ 500.00", "Rim Size: 18 $ 1100.00", "Miscellaneous: CrusieControl  $ 2500.00", "Seats 7 $ 500.00" ], 31300.00, 7]


- The first digit is the make
    - 1: ThisAuto
    - 2: ThatAuto
    - 3: OtherAuto
- The second digit is vehicle type
    - 1: Sedan
    - 2: Coupe
    - 3: Minivan
    - 4: SUV
    - 5: Truck


## pidFormat
Type: private
Input: int m – id of manufacture
Return: Integer Array of pid Format
Description: Exhibits the following format:

m = 1.     ThisAuto
       a.  Returns [4,5,6,7,8,9,10,11,12]
        index 0.   First 4 digits is the model
        index 1.   Digit 5 Exterior Color

index 2. Digit 6 Interior Color
index 3. Digit 7 Powertrain
index 4. Digit 8 Seat Type
index 5. Digit 9 Radio Type
index 6. Digit 10 Tire Size
index 7. Digit 11 Rim Size
index 8. Digit 12 Miscellaneous

m = 2.     ThatAuto

a. Returns [6,7,8,9,10,11,12,13,15]
index 0. First 6 digits is the model
index 1. Digit 7 Exterior Color
index 2. Digit 8 Interior Color
index 3. Digit 9 Powertrain
index 4. Digit 10 Seat Type
index 5. Digit 11 Radio Type
index 6. Digit 12 Tire Size
index 7. Digit 13 Rim Size
index 8. Digits 14 and 15 Miscellaneous

m = 3.     OtherAuto

a. Returns [4,5,6,7,8,9,10,11,16]
index 0. First 4 digits is the model
index 1. Digit 5 Exterior Color
index 2. Digit 6 Interior Color
index 3. Digit 7 Powertrain
index 4. Digit 8 Seat Type
index 5. Digit 9 Radio Type
index 6. Digit 10 Tire Size
index 7. Digit 11 Rim Size
index 8. Digit 12 thru 16 Miscellaneous