**Comp2522 Inner classes and Factories**

**Purpose**: Examine a simple factory design pattern and use of more advanced inner class features

**Description**: You are designing a “factory” for Ford Cars. Ford was the founder of the assembly line (not the car as often thought). His first two models ModelA and later ModelT are to be manufactured by your “factory”

Follow the pattern provided for a Factory design pattern with the following information:

A Car can turn. When a Car turns the distance driven is reset to 0

A Car can drive. When a Car drives it moves 1 unit forward.

Cars stop when they reach a max distance (see specific models)

All Cars contain a class Construct that has a DEFINED method constructCar() that builds (calls methods) the engine (method) and then the body (method) for the specific car (DEFINED METHODS buildEngine(), buildBody()). This Construct is part of the **interface** found for a Car! NOTE: you must create an INTERFACE called Car that holds the class Construct.

A CarFactory has a simple method to return a Car via getCar()

A ModelA (class) is a Car (implements interface Car) with a modified Construct class (therefore a class that extends Construct) providing a buildEngine() method to display “basic engine”, a buildBody() method that displays “ModelA body”. Note: no modification is done to method “constructCar()” that was inherited. A MAXDISTANCE it can travel in one direction is 3. The drive() method displays the current distance driven and checks if the max distance has been achieved (returning true or false). The turn() method displays “turning ModelA” and resets the distance. It has a static member for providing the factory needed to produce the Car. Note that the constructor is private and creates a Construct object and calls the constructCar() method on it.

A ModelT is a similar Car as above with the displays indicating ModelT instead of ModelA and a MAXDISTANCE of 5.

The FordFactory class has been provided to test your code, do NOT alter. See output below

