STATICS

## Objective

In this lab you will implement a factory pattern. You will get a ‘Registration Plate’ from a ‘Registration Plate Factory’ class.

### **Part 1 - Step by step instructions**

In this section you will create a Vehicle class with a static member and instance members and use static members of a Factory class.

1. Open the **Labs** Console solution if not open already
2. Create a class called **Vehicle** with integer fields called **speed** and **lane**.  
   Also record the distance travelled using an int variable called **distanceTravelled**.
3. Create a constructor to set the initial speed and lane fields.
4. Add two methods called **Accelerate** and **Brake**.  
   **void** Accelerate (**int** amount)  
     
   The Accelerate method will increase the speed, but never more than **200**!   
   It also adds to the **distanceTravelled** (add 1m per unit of speed).  
    **void** **Brake**(**int** amount)  
   // to reduce the speed by the amount
5. It is never a good idea to use magic numbers like 200 (for maximum speed). Replace 200 with a constant like **private** **const** **int** ***MAX\_SPEED*** = 200;
6. Add another method as **string** **GetDetails()** to get the vehicle's details such as speed, lane, distanceTravelled and the plate number (see below)
7. Every vehicle has a registration **plate** which is a complex object.  
   Vehicle plates have information about the City, the country and the year of registration which should better be defined as a class.
8. Create a class called ***RegistrationPlate***.
9. Give this class a String field called ***regPlate***. Provide a public property method to read the value of this field but make the set part private.
10. Create a constructor to set the *regPlate field*.
11. Add a new field to the **Vehicle** class called ***registrationPlate*** of type *RegistrationPlate*.   
    Tip: ***RegistrationPlate*** *registrationPlate;*

You'll soon set this field using the **Factory pattern**.   
Please do not instantiate it yet!

## Registration plate Factory

1. Let's create a new **factory** class that creates instances of ***RegistrationPlate***.
2. Create a separate class called ***RegistrationPlateFactory***
3. Add the following static array of reg numbers to *RegistrationPlateFactory*.  
    **private** **static** String[] ***regPlates*** =   
    { "MRB1G", "RU16", "TOYS4US", "HNZ57", "PUT3", "JB007" };  
     
   Note, The array is static because it will be accessed by an static method.
4. Create a **static** method called **GetNextRegistrationPlate**()  
   This method should return an instance of *RegistrationPlate*.   
     
   To create an instance, you'll need the next registration plate from the regPlates array. How would you get the next regPlate index?  
     
   **Return null if you run out of regPlates**
5. Back in the Vehicle class's constructor, assign a registration plate to the vehicle using the *RegistrationPlateFactory* class.  
    **Tip:   
   *this****.registrationPlate =* ***RegistrationPlateFactory.* GetNextRegistrationPlate**();
6. Create a new static method called Lab3() in the Program class
7. Call Lab3() from within Main()
8. Comment out the call to the Lab2() method
9. In the Lab3() method, create an array of Vehicles, populated with three new Vehicles.
10. Print details of the vehicles created in the above step.   
    Please make sure the plates are correctly assigned.

### **Enhancing the Vehicle class**

1. How would you count the number of vehicles created?  
   I know you created 3! But what if different parts of your program create Vehicles?

Create a static method called **GetCount()** in the vehicle class to return the count of vehicles. Alternatively, you could create a **static** Property procedure called **Count** to return this value. Just make sure the Set part of the property procedure is **private**.  
  
**Tip:** Vehicle's constructor is invoked whenever a vehicle is created.   
Please see the slides for more help.

### **Writing code to use the Vehicles**

1. Return to the Lab3() method.

a) The initial speed of the Vehicles is **zero.**

b) Make sure they are placed in lanes 1, 2, 3

1. Write a while loop to race the cars (Accelerate them)   
   until the distance travelled by one of them is more than **1000**.

You'll Accelerate each vehicle by a random number between 1-10 by using the following code:

Random **rand** = **new** Random();  
**int** n = **rand.Next**(10)+1;

1. Display details of each vehicle as they Accelerate on each iteration of the while loop.
2. As soon as a vehicle has travelled 1000m or more, announce it as the winner and break out of the loop.
3. You can get creative and assign a driver to each vehicle   
   (create a Driver class).

Part 2 – using Statics with the graphical project you created earlier

In this part you'll make use of the static keyword.  
Clearly setting the world dimension to x=10, y=10, w=600, h=400 is not a good idea.   
  
All balls should share the same world coordinates. This is a perfect place to use the static keyword.

1. Create four ***static int*** fields in the **Ball** class called **worldX, worldY, worldW** and **worldH** to represent the x, y, width and height of the world.
2. Create static method to set the static fields for the world coordinates as   
   **public static void SetWorld(int wx, int wy, int ww, int wh)**
3. Modify the Ball class code to use the world's coordinates.

At the beginning of the Form Load event Invoke the SetWorld() method of the Ball class to set the world's coordinate for every Ball.  
  
 Ball.SetWorld(10, 10, 600, 400);

1. And modify the form\_paint event to use the new world coordinates.  
     
    private void Form1\_Paint(object sender, PaintEventArgs e)

{

e.Graphics.Clear(Color.Gray);

e.Graphics.FillRectangle(Brushes.Red, Ball.worldX, Ball.worldY, Ball.worldW, Ball.worldH);  
. . .

1. Run your code to see if it works.
2. Change the World!   
   Let’s have some fun with the World’s static fields.   
     
   Every time a ball hits the right edge of the world, add 5 to worldX and  
     
   when a ball hits the left edge of the world, subtract 5 to worldX.

\*\* End \*\*