C#  
EXCEPTIONS

## Objective

The primary objectives for this lab are to be able to throw and catch exceptions and have a full understanding of the .NET runtime’s propagation mechanism for exceptions until a handler is found.

## Overview

Firstly, you will handle the exceptions by throwing an exception object in your Account class if balance goes into red.  
  
In a second practical you will enhance the MotorwaySimulator to throw an exception when the **RegistrationFactory** runs out of registration plates.

### **Part 1**

### Using try/throw/catch in the Account class

1. Locate the **Account** class you created earlier

Add a **Close()** method to the Account class which displays a simple message.

1. Back in the Program class create a method called **Lab6()**,
2. Call Lab6() from within the Main() method.
3. In the Lab6() method, create an instance of Account with a balance of £100.
4. Withdraw £50 and then display the account’s details using the getDetails() method.
5. Withdraw £600 and display the account's details  
   Please note, there is no Withdrawal limit in our friendly bank!
6. Fix the **Withdrawal()** method so that it throws an **ArgumentException** when the balance becomes negative.
7. Catch this exception in Lab6() and display a suitable user-friendly message.
8. Build and run the program now.   
   Make sure you get the Account class to throw the exception and   
   make sure it is handled in Lab6().
9. Add a **finally** clause to the exception handler in Lab6() to Close the account (using the Close() method) no matter if there is an exception or not.

### **Part 2**

1. Let's get back in the motorway simulator lab which you created earlier.
2. The factory class will at some time run out of registration plates.   
     
   How would you handle this situation?   
   How would you inform the Vehicle class that we've ran out of reg plates?

How would the Vehicle class inform the caller we cannot create a vehicle?

Implement the required exception mechanisms the best you can

**\*\* End \*\***