# Computing, Year 3

# Rich Web Applications

# Lab sheet 4

# Going Further with TypeScript Syntax

## Overview

In this lab, you'll enhance the Film class from the previous lab so that it holds more details about reviews submitted by amateur film-goers and professional film critics. You'll define 3 new classes:

* Review  
  This will be an abstract class that holds basic information about a film review, i.e. just a score between 1 and 5.
* BriefReview  
  This class will inherit from Review, and will contain a brief comment from an amateur film-goer (e.g. 'I loved this film').
* CriticalReview  
  This class will also inherit from Review, and will represent a film review from a professional film critic. The class will contain the name of the media outlet that the film critic works for (e.g. a newspaper or magazine), plus a list of detailed observations about the film.

## Roadmap

There are 6 exercises in this lab.  
Here is a brief summary of the tasks you will perform in each exercise; more detailed instructions follow later:

1. Defining the Review abstract class
2. Defining the BriefReview class
3. Defining the CriticalReview class
4. Refactoring the Film class to hold reviews
5. Running the application

## Exercise 1: Defining the Review abstract class

Open a browser and go to http://www.typescriptlang.org/play.

At the top of the code, define the Review abstract class. The class should hold a numeric score (e.g. between 1 and 5) and have a simple toString() method to return the score in a nice string format. Also implement a constructor as appropriate.

## Exercise 2: Defining the BriefReview class

Define the BriefReview class. The class should inherit from Review and should define an additional instance variable to hold a brief comment about the film. Implement a constructor as appropriate, and also override toString() to return all the information in this class and the superclass.

## Exercise 3: Defining the CriticalReview class

Define the CriticalReview class. The class should also inherit from Review and should define an additional instance variable to hold a variable-length list of observations. Implement a constructor and toString() method here too.

## Exercise 4: Refactoring the Film class to hold reviews

At the moment, the Film class has an instance variable named scores that holds a collection of numeric scores for the film. Refactor the Film class so that it holds a collection of Review objects instead (i.e. an array that can hold BriefReview and CriticalReview objects).

You'll need to make some changes to the methods in the class as well:

* Rename the rate() method to review(). Change the method signature so that it receives a Review parameter rather than a simple number. Modify the method implementation so that it adds the review to the collection of reviews.
* Refactor the averageScore() method so that it plucks score values out of the collection of reviews.
* Define a new method named reviewsAsString(), which returns a single string containing all the review information for the film.

## Exercise 5: Running the application

Modify the client code so that it submits some reviews for film. Then call averageScore() and reviewsAsString() to get review-related info from the film. Display the info on the browser Console window. Run the application and verify the correct details are displayed.