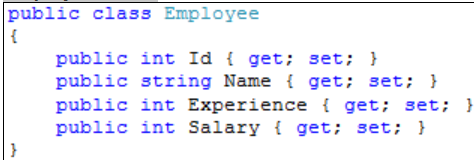
**Part 37 - C# Tutorial - Delegates Example - I**

**Where did use delegates in your project - Part 1**

**Where did you use delegates in your project?**  
or  
**How did you use delegates in your project?**  
or  
**Usage of delegates in a Real Time Project?**  
  
**This is a very common c sharp interview question**. Delegates is one of the very important aspects to understand. Most of the interviewers ask you to explain the usage of delegates in a real time project that you have worked on.  
  
Delegates are extensively used by framework developers. Let us say we have a class called **Employee** as shown below.

**Employee Class**



The **Employee** class has the following properties.  
**1. Id**  
**2. Name**  
**3. Experience**  
**4. Salary**  
  
Now, I want you to write a method in the **Employee** class, which can be used to promote employees. The method should take a list of Employee objects as a parameter, and should print the names of all the employees who are eligible for a promotion. But the logic, based on which the employee gets promoted should not be hard coded. At times, we may promote employees **based on their experience** and at times we may promote them **based on their salary** or may be **some other condition**. So, the logic to promote employees **should not be hard coded with in the method**.  
  
To achieve this, we can make use of **delegates**. So, now I would design my class as shown below. We also, created a **delegate EligibleToPromote**. This delegate takes **Employee** object as a parameter and **returns a boolean**. In the **Employee** class, we have **PromoteEmpoloyee** method. This method takes in a list of Employees and a Delegate of type **EligibleToPromote** as parameters. The method, then loops thru each employee object, and passes it to the delegate. If the delegate returns true, then them Employee is **promoted**, else not promoted. So, with in the method **we have not hard coded any logic on how we want to promote employees**.

