Assigning roles to/from users:

1) What actions do we need to perform?
   a. Assign a role if not already assigned
      i. Parameters?
         1. userId
         2. roleName
   b. Check to see if assigned
      i. Parameters?
         1. userId
         2. roleName
   c. Remove from role if already assigned
      i. Parameters?
         1. userId
         2. roleName
   d. Get a list of all users for a given role
      i. Parameters?
         1. roleName
   e. Get a list of all roles assigned to a user
      i. Parameters?
         1. userId
   f. Get a list of all roles NOT assigned to a user
      i. Parameters?
         1. userId

2) Where do we need to do this?
   a. Controller?
   b. Create helper class?

3) Build the Helper now…
   {Hand Out}

4) How are we going to display this information?
   a. Can we use the existing project model, or do we need a view model?
   b. What do we need in the view model?
      i. The user
      ii. The list of roles to assign or remove
      iii. Placeholder for selections

5) Build the view model now…
   For e.g.

```csharp
public class AdminUserViewModel
{

    public ApplicationUser User { get; set; }
    public MultiSelectList Roles { get; set; }
    public string[] SelectedRoles { get; set; }


}
```

6) Now we need to set up the view model so we can send it to the view
   a. What controller actions do we need?
      i. AssignRoles
      ii. RemoveRoles

7) Build the controller actions now…

```csharp
//GET: EditUser

public ActionResult EditUser(string id)
{
    var user = db.Users.Find(id);
    AdminUserViewModel AdminModel = new AdminUserViewModel();
    UserRolesHelper helper = new UserRolesHelper();
    var selected = helper.ListUserRoles(id);
    AdminModel.Roles = new MultiSelectList(db.Roles, "Name", "Name", selected);
    AdminModel.Id = user.Id;
    AdminModel.Name = user.DisplayName;

    return View(AdminModel);
}
```

```csharp
//POST: EditUser

public ActionResult EditUser(AdminUserViewModel model)
 {
     var user = db.Users.Find(model.id);
     UserRolesHelper helper = new UserRolesHelper();
     foreach(var rolermv in db.Roles.Select(r => r.Id).ToList())
     {
         helper.RemoveUserFromRole(user.Id, rolermv);
     }

     foreach(var roleadd in db.Roles.Select(r => r.Id).ToList())
     {
         helper.AddUserToRole(user.Id, roleadd);
     }

     return RediretToAction("Index");
 }
```

8) Now we need to create the view
    a. We can scaffold the view if we want, or we can write from scratch...

9) Build the views now

   To add the select list...

   ```csharp
   @Html.ListBox("SelectedRoles", Model.Roles)
   ```