Moq

Building on previous labs (event & generics), we will create a class to represent *Company* and use mocks to decouple employees from the company for testing.

Steps

- Optional - create a new xUnit project
    - copy source files from the event and generic projects
- Or, continue with your last project
    - copy source files from the generic project if needed
- Using Nuget, search and install Moq
- Create a new class named *CompanyTest*
- Create a new class named *Company*
- Using the *Quick Refactorings Tool (screwdriver)* extract the interface
- *IEmployee* from the class **Employee**
    - Note - exclude the deconstructors
- Design tests to specify the behavior of the new *Company* class
    - *CompanyTest Constructor*:
        - Create a new company with a name and id; stash in an instance variable.
        - Create 2 or 3 mock IEmployee objects - stash in instance variables
    - *TestCompanyHire* create a test to hire and verify that the employees where hired
    - *TestCompanyPay* create a test to hire and pay employees - verify that they are paid
    - Verify that the tests fail
- Implement *Company* so that tests pass

The initial *Company* class is shown below:

```
public class Company
{
    public Company(string name, string taxId)
    {
        throw new NotImplementedException();
    }

    public string Name { get; set; }
    public string TaxId { get; set; }
    public GenericArrayList<IEmployee> Employees { get; set; }
    public void Hire(IEmployee emp)
    {
        throw new NotImplementedException();
    }
    public double Pay()
    {
        throw new NotImplementedException();
    }
}
```