First and FirstOrDefault

First, FirstOrDefault

✓ First and FirstOrDefault both are used (independently) to return first element from a datasource.

✓ If the element is not available at first index -

  ✓ First will throw an exception.

  ✓ But FirstOrDefault will return default value of data source element.

```
0 references
static void Main(string[] args)
{
    List<int> numbers = new List<int>() { 1,2,3,4,5,6,7,8,9,10};

▶ var ms = numbers.First();
         ● ms 1
    Console.ReadLine();
}
```

```
0 references
static void Main(string[] args)
{
    List<int> numbers = new List<int>() { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };

    var ms = numbers.Where(x => x > 5).First();

    var ms1 = numbers.First(x => x > 5);

    Console.ReadLine();
}
```

always use the ms1 filtering it will reduce the time complexity

```csharp
static void Main(string[] args)
{
    var users = new List<User>()
    {
        new User(){Id = 1, UserName = "Admin", Password="Admin" },
        new User(){Id = 2, UserName = "UserA", Password="UserA" },
        new User(){Id = 3, UserName = "UserB", Password="UserB" },
        new User(){Id = 4, UserName = "UserC", Password="UserC" }
    };

    var ms = users.FirstOrDefault(x => x.UserName == "Admin" && x.Password == "Admin");

    var mixedS = (from user in users
                  select user).FirstOrDefault(x => x.UserName == "Admin" && x.Password == "Admin");

    Console.ReadLine();
}
```