Working with Classes

Jesse Liberty

@jesseliberty http://jesseliberty.com

Flavors of Properties

- Full
- Automatic
- Read Only
- Read Only With Initialization

“Full” Property

```
private string name;

public string Name{
    get { return  name; }
    set {  name  = value; }
}
```

Automatic

```
public string Name { get; set; }
```

```
private string name;

public string Name{
    get { return  name; }
    set {  name  = value; }
}
```

Read Only

```
public string Name { get; }
```

Initialized

```
public string Name { get; } = "Jesse Liberty";
```

Method Overloading

Two methods with the same name

```
public double computeSalary(double start){}
public double computeSalary(double start, double bonus){}
public double computeSalary(double start, string level){}
```

Two Or More Methods May Have The Same Name As Long As...

They differ in the number of parameters and/or

They differ in the type of parameters

Constructors Create Your Object

```
Public class Worker{

  public Worker(int age, string name){
    Age = age;
    Name = name;
  }
}
```

Constructors Have …

- No return value (not even Null)
- Same name as the class
- Zero or more parameters

Default Constructor

A constructor with no parameters.

If you don’t provide
any constructors, a default
constructor is provided
for you.

Implicit Default Constructor

```
public class Worker {

    public string Name { get; } = "Jesse Liberty";

    public int Age { get; } = 40;

    private double salary;
    public double Salary {
        get { return salary; }
        set { salary = value; }
    }

 Worker worker = new Worker();
```

If you do have any constructors, and you want a default constructor, you must create it explicitly.

Need Explicit Default Constructor

```
public class Worker {
    public int Age { get; set; }

    public string Name { get; set; }

    public Worker(int age, string name) {
        Age = age;
        Name = name;
    }

}


 Worker worker = new Worker();  //error!
```

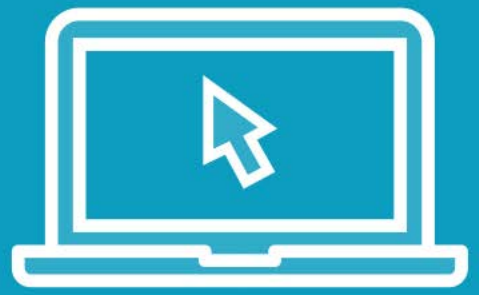
Creating Explicit Constructor

```
public class Worker {

    public Worker(int age, string name) {
        Age = age;
        Name = name;
    }

    // explicit default constructor
    public Worker(){
        Office = 101;
    }
```

Demo

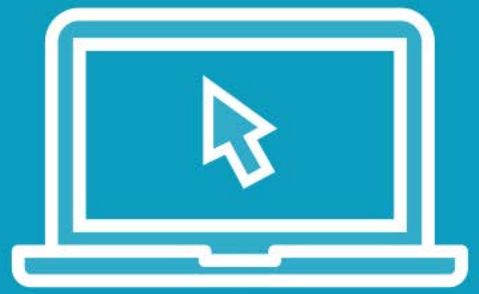


Adding Constructors

Static Members Are Accessed Through the Class, Not Through an Object

Static Classes
Cannot Be Instantiated

Demo



Static Members