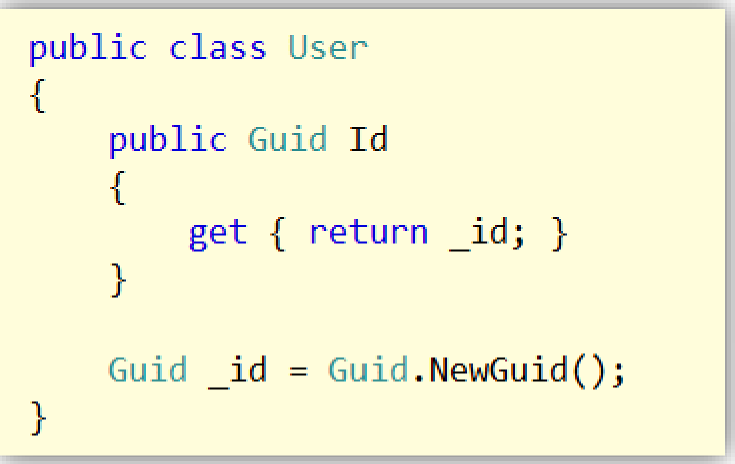
Language Features Part I

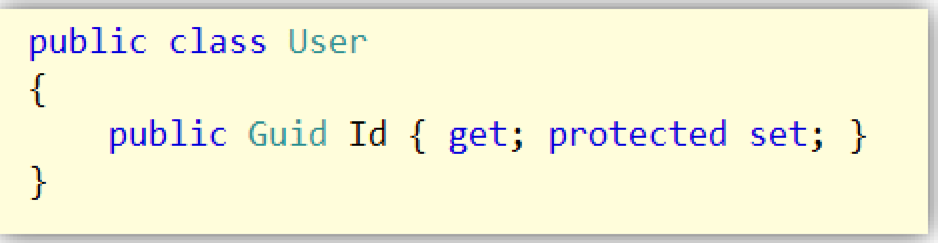
K. Scott Allen
odetocode.com
@OdeToCode

pluralsight
hardcore dev and IT training

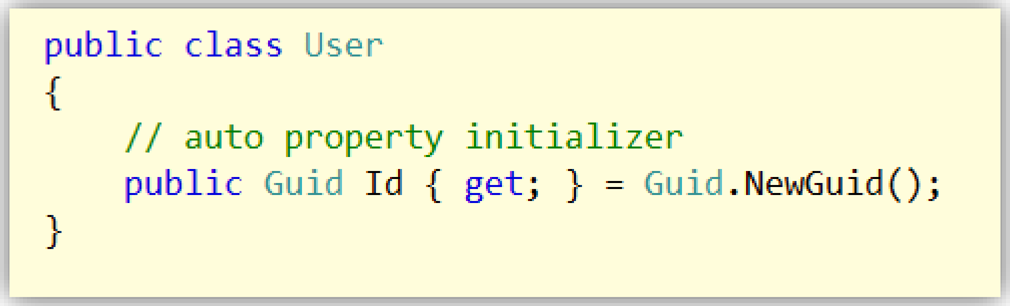
Auto Property Initializers



```
public class User
{
    public Guid Id
    {
        get { return _id; }
    }

    Guid _id = Guid.NewGuid();
}
```

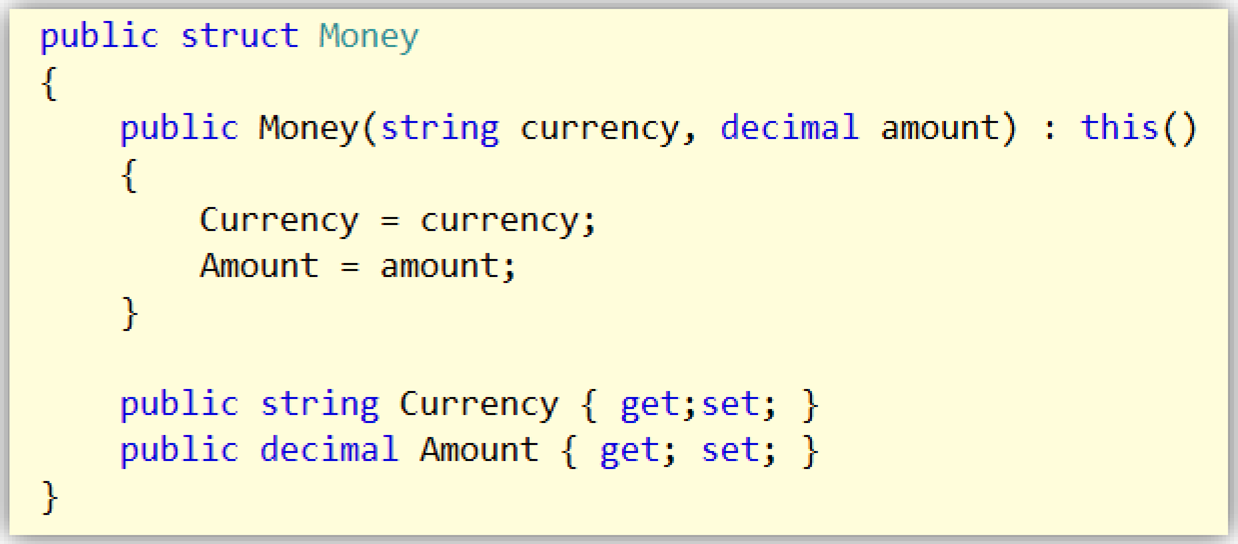


```
public class User
{
    public Guid Id { get; protected set; }
}
```

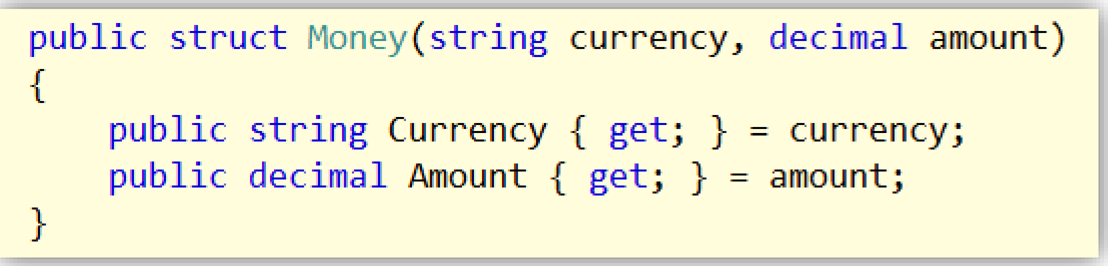


```
public class User
{
    // auto property initializer
    public Guid Id { get; } = Guid.NewGuid();
}
```

Primary Constructors

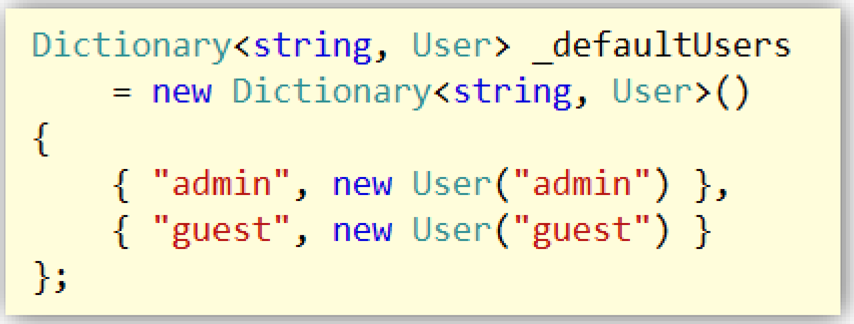


```
public struct Money
{
    public Money(string currency, decimal amount) : this()
    {
        Currency = currency;
        Amount = amount;
    }

    public string Currency { get;set; }
    public decimal Amount { get; set; }
}
```

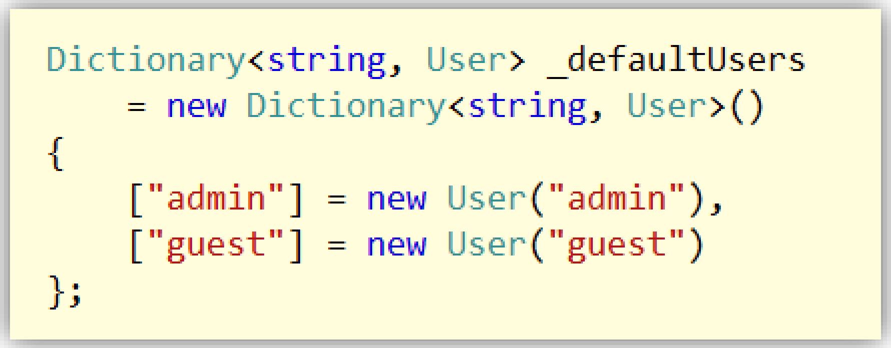


```
public struct Money(string currency, decimal amount)
{
    public string Currency { get; } = currency;
    public decimal Amount { get; } = amount;
}
```

Dictionary Initializer

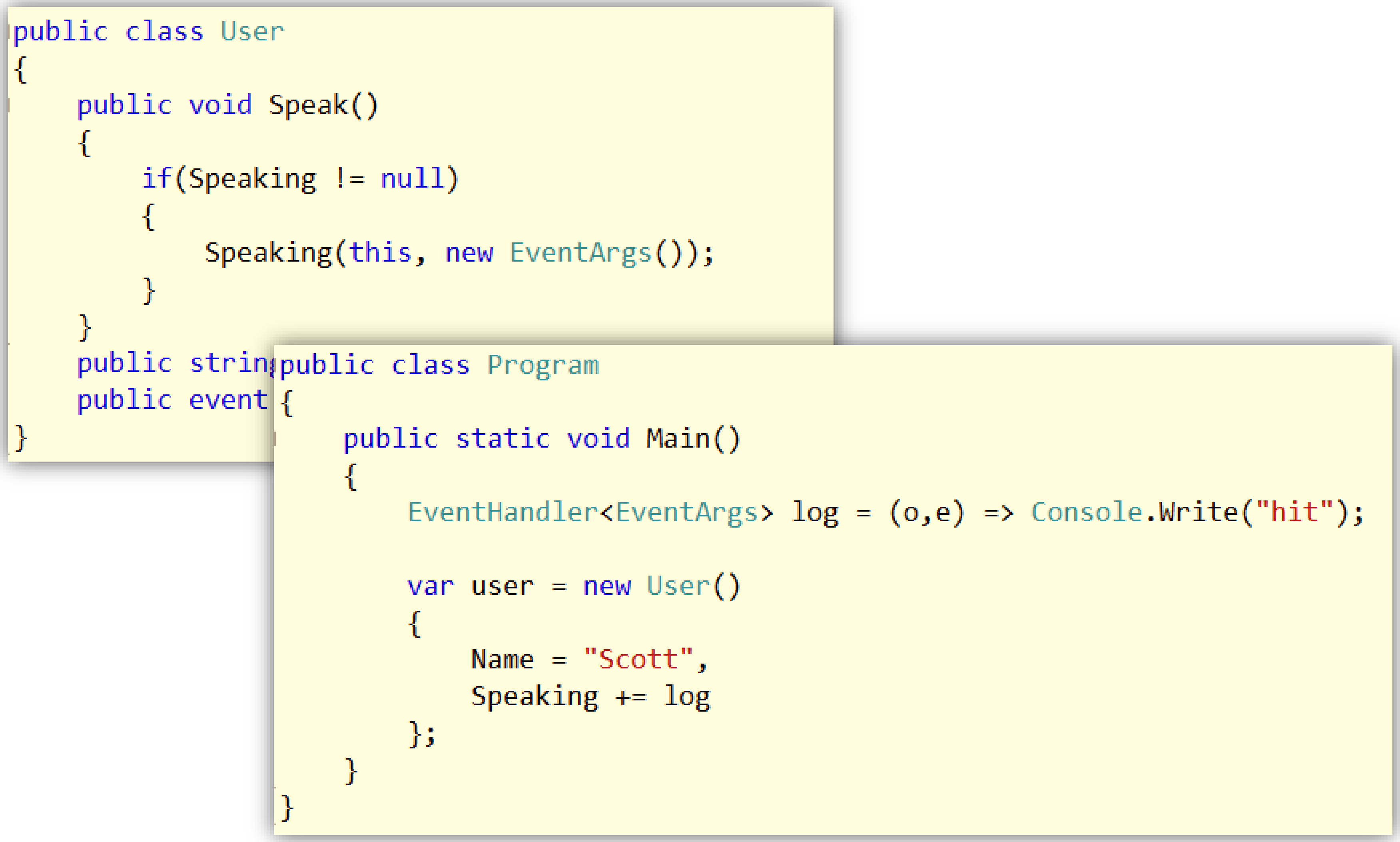


```
Dictionary<string, User> _defaultUsers
    = new Dictionary<string, User>()
{
    { "admin", new User("admin") },
    { "guest", new User("guest") }
};
```



```
Dictionary<string, User> _defaultUsers
    = new Dictionary<string, User>()
{
    ["admin"] = new User("admin"),
    ["guest"] = new User("guest")
};
```

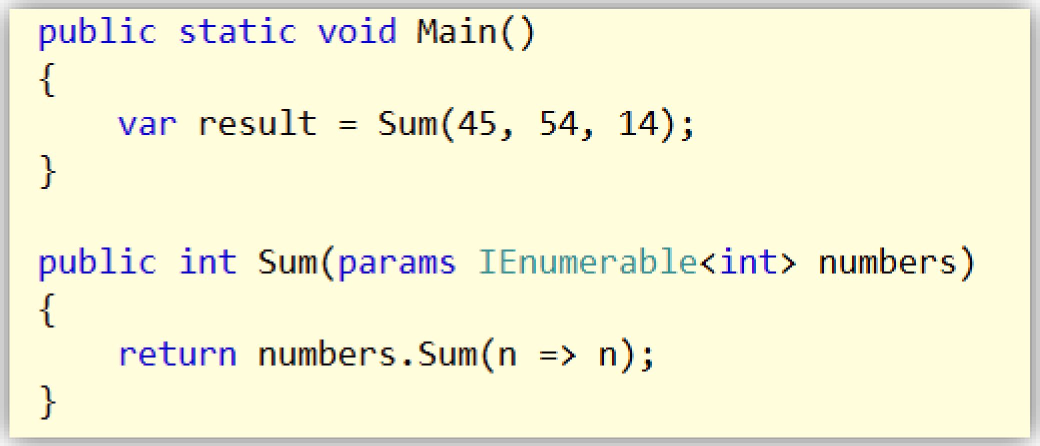
Event Initializers



```
public class User
{
    public void Speak()
    {
        if(Speaking != null)
        {
            Speaking(this, new EventArgs());
        }
    }
    public strin
    public event
}
```

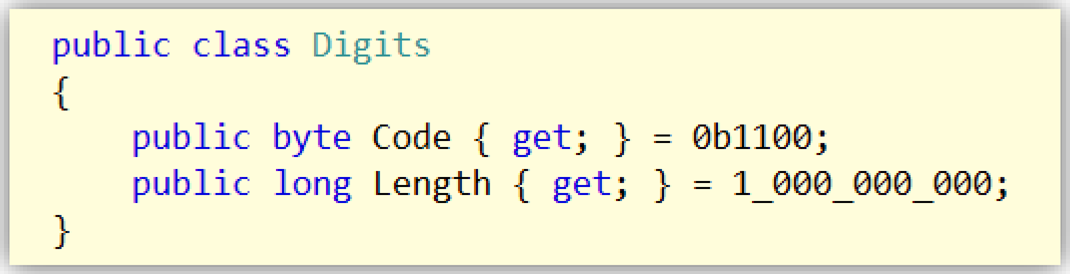
```
public class Program
{
    public static void Main()
    {
        EventHandler<EventArgs> log = (o,e) => Console.Write("hit");

        var user = new User()
        {
            Name = "Scott",
            Speaking += log
        };
    }
}
```

params and IEnumerable



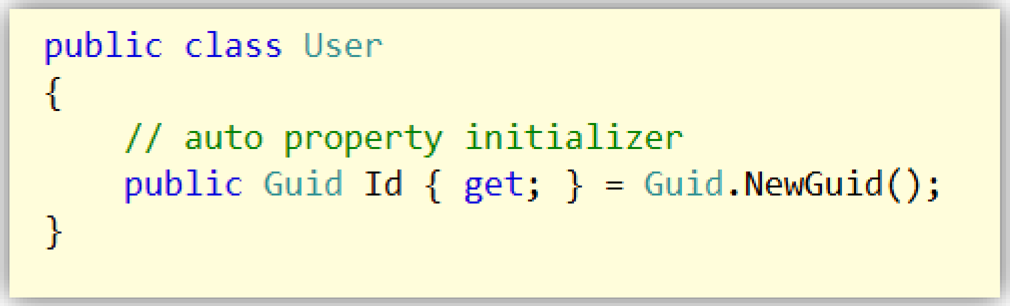
```
public static void Main()
{
    var result = Sum(45, 54, 14);
}

public int Sum(params IEnumerable<int> numbers)
{
    return numbers.Sum(n => n);
}
```

Literals and Separators

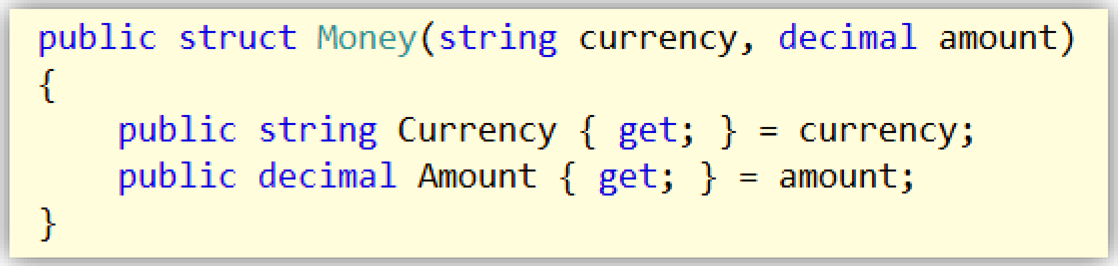


```
public class Digits
{
    public byte Code { get; } = 0b1100;
    public long Length { get; } = 1_000_000_000;
}
```

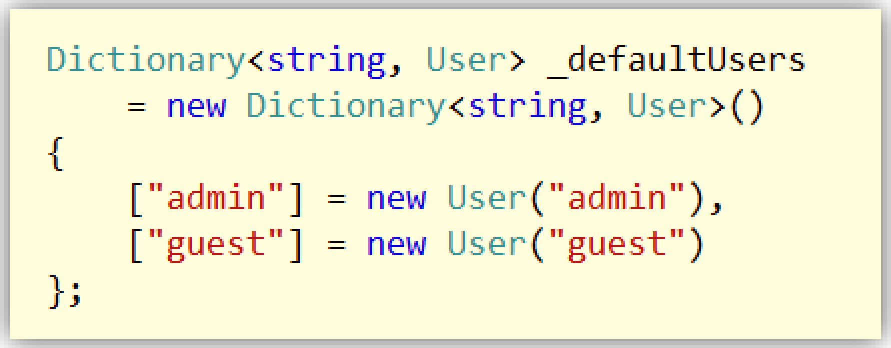
Conclusion



```
public class User
{
    // auto property initializer
    public Guid Id { get; } = Guid.NewGuid();
}
```



```
public struct Money(string currency, decimal amount)
{
    public string Currency { get; } = currency;
    public decimal Amount { get; } = amount;
}
```



```
Dictionary<string, User> _defaultUsers
    = new Dictionary<string, User>()
{
    ["admin"] = new User("admin"),
    ["guest"] = new User("guest")
};
```