

Exploring Program Flow and Statement Features



Alex Wolf

Software Developer

www.crywolfcode.com



Going Further with Flow and Statements

Program Flow

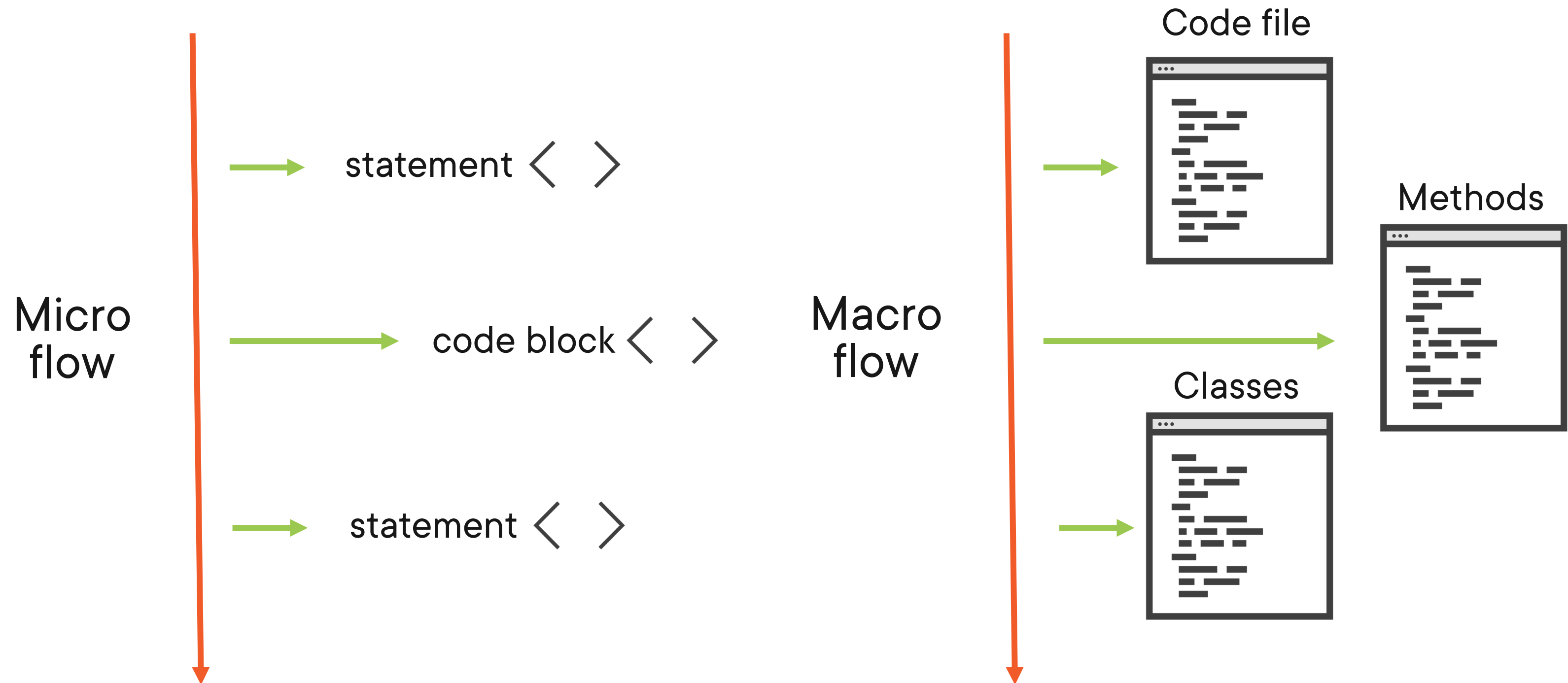
**Explore flow across the app as
a whole**

Advanced Statements

**Explore more specific, but
useful statement features**



Program Flow Scope Considerations



Demo



Controlling Program Flow through User Input



Demo



Enhancing Program Flow using Loops



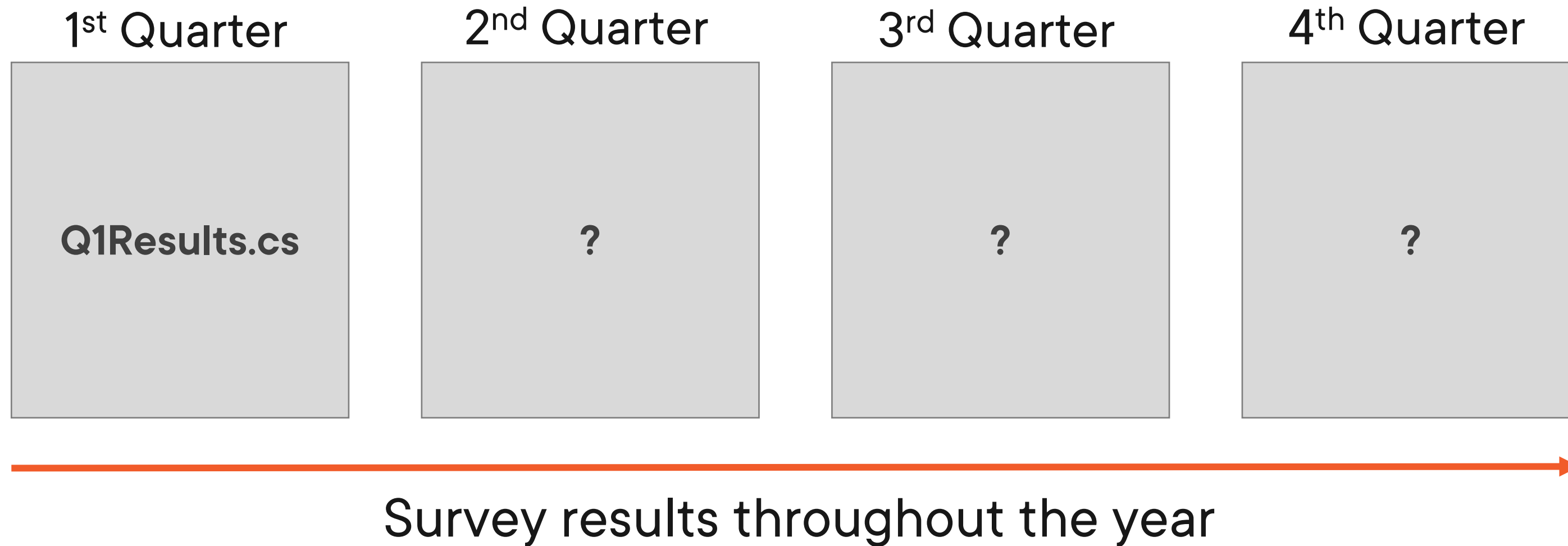
Demo



Improving Program Flow using Dynamic Data



Dynamic Data Considerations



Demo



Generating Reports using Dynamic Data



Exploring Advanced Selection Statement Features



```
double discountRate;
```

```
if (product.Category == "Coffee")
```

```
{
```

```
    discountRate = .25
```

```
}
```

```
else
```

```
{
```

```
    discountRate = .33
```

```
}
```

◀ A simple conditional assignment...

◀can still end up pretty verbose

```
var discountRate = product.Category == "Coffee" ? .25 : .33;
```

```
var discountRate = product.Category == "Coffee" ?  
GetDiscount("Coffee") : GetDiscount("Other");
```

Introducing Ternary Statements

The conditional (ternary) operator can greatly simplify assignments and value selections

Ternary statements can also execute methods that return values in place of constants



A note about Ternary statements.



Switch Statement Expression Syntax

Program.cs

```
int categoryId;  
switch (productName)  
{  
    case "Granola":  
        categoryId = 1;  
        break;  
    case "Latte":  
        categoryId = 2;  
        break;  
}
```

Program.cs

```
var categoryId = productName switch {  
    "Granola" => 1,  
    "Latte" => 2,  
    "Mug" => 3,  
}
```

```
int discountId = 25;
switch (discountId)
{
    case int id when id < 10:
        // returns false
        break;

    case int id when id < 10 && id > 20:
        // returns false
        break;

    case int id when id > 20 && id < 30:
        // returns true, code section would
        execute
        break;
}
```

◀ **discountId is copied to local id variable**

◀ **Each nested expression is evaluated**

◀ **If the result is true, that block executes**

```
Shape myShape = new Rectangle() { width = 20 };  
  
switch (myShape)  
{  
  
    case Square s when shape.width < 10:  
        // returns false  
        break;  
  
    case Rectangle r when shape.width > 10:  
        // returns true  
        break;  
  
    case Circle c when c.diameter > 10:  
        // returns false  
        break;  
  
}
```

◀ Checks that myShape is a Rectangle with a width greater than 10

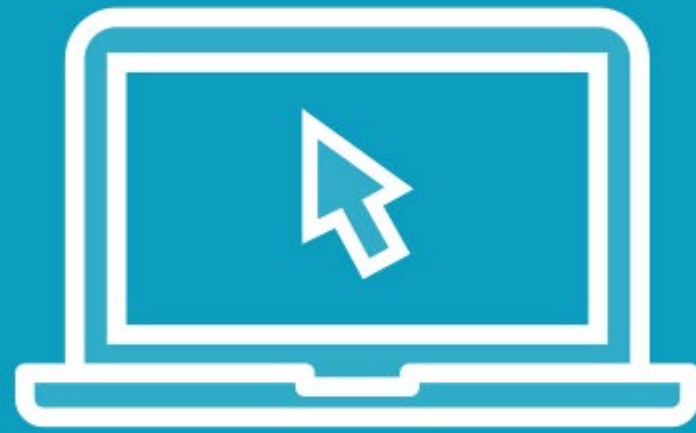
Demo



Exploring Additional Switch Statement Features



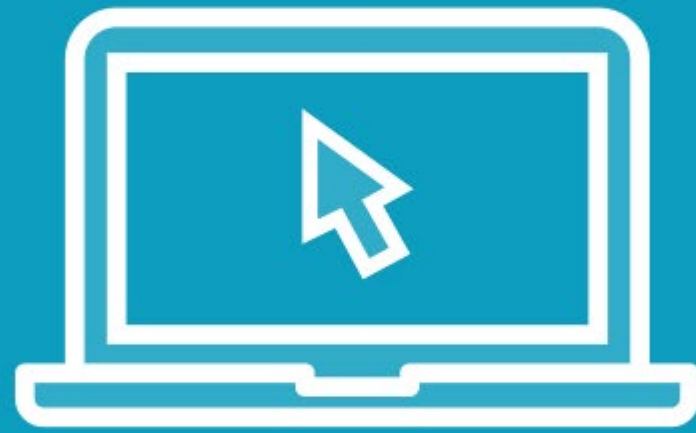
Demo



Streamlining if/else Selection Syntax



Demo



Touring the Final State of the App



Overview/ Summary



- Program flow can be controlled at the macro and micro level
- Program flow can also be influenced by user actions and behaviors
- Ternary operators are an alternative option to streamline conditional value assignment
- Switch Statements also provide a streamlined expression syntax for certain scenarios
- Pattern matching using keywords like “if” or “when” allows for additional decision making features



Thank you, and good luck!

