

Avoiding Common Mistakes



Craig Golightly

Senior Software Consultant

@seethatgo | www.seethatgo.com

Overview



Syntax mistakes

Common errors

- Control statements
- Variables

Demo – find and fix

Best practices to follow

- Standard libraries



Missing Values



Usually generate a syntax error

Forgetting ;

Not closing

- ()
- { }
- []
- ' '
- " "

Code editor can identify



Multi-line Strings

```
String multi1 =  
    "this is an example\n" +  
    "of a multi-line string\n" +  
    "be sure to remember newlines\n";
```

```
String multi2 =  
    """  
        this is another way  
        to create a multi-line string.  
        It's easier to create and edit.  
    """;
```



If Statements

```
if (a==b)
{
    doTrue();
}
else
{
    doFalse();
}
```

Sometimes used

```
if (a==b) {
    doTrue();
} else {
    doFalse();
}
```

Very widely used

```
if (a==b)
    doTrue();
else
    doFalse();
```

Valid with one line
Error prone when edit

```
if (a==b){doTrue();}else{doFalse();}
if (a==b)doTrue();else doFalse();
```

Sometimes used for single if statements



```
boolean a = true;  
boolean b = false;  
if [a=b] {  
    doTrue();  
} else {  
    doFalse();  
}
```

◀ = is assignment operator

◀ statement will always evaluate to false
◀ a has changed value and is now false

```
boolean a = true;  
boolean b = false;  
if [a==b] {  
    doTrue();  
} else {  
    doFalse();  
}
```

◀ == is comparison operator

◀ a has not changed value and is still true



Comparing Objects

`.equals()`

Define equality for Class

Customizable

Compare attribute values

`==`

Only true if same instance

Not typically used with Objects



```
public class Bucket {  
    private int size;  
    private String color;  
    ...  
  
    Bucket one = new Bucket(10, "red");  
    Bucket two = new Bucket(10, "red");  
    if [one.equals(two)] {  
        doTrue();  
    } else {  
        doFalse();  
    }  
  
    if [one == two] {  
        doTrue();  
    } else {  
        doFalse();  
    }
```

◀ Class with two attributes - size and color

◀ Two different instances of Bucket Class with same attribute values

◀ .equals() can compare size and color attributes and return true

◀ Always false unless variables one and two reference same instance of Bucket object



Switch Statement

```
switch (color) {  
    case "red":  
        System.out.println("In red");  
        break;  
    case "orange":  
        System.out.println("In orange");  
        break;  
    case "yellow":  
        System.out.println("In yellow");  
        break;  
    default:  
        System.out.println("In default");  
        break;  
}
```

color = red;
//output
In red

```
switch (color) {  
    case "red":  
        System.out.println("In red");  
    case "orange":  
        System.out.println("In orange");  
    case "yellow":  
        System.out.println("In yellow");  
    default:  
        System.out.println("In default");  
}
```

color = red;
//output
In red
In orange
In yellow
In default



```
switch (color) {  
    case "red" -> System.out.println("In red");  
    case "orange" -> System.out.println("In orange");  
    case "yellow" -> System.out.println("In yellow");  
    default -> System.out.println("In default");  
}
```

```
color = red;  
//output  
In red
```

◀ Arrow statement - no fall through behavior

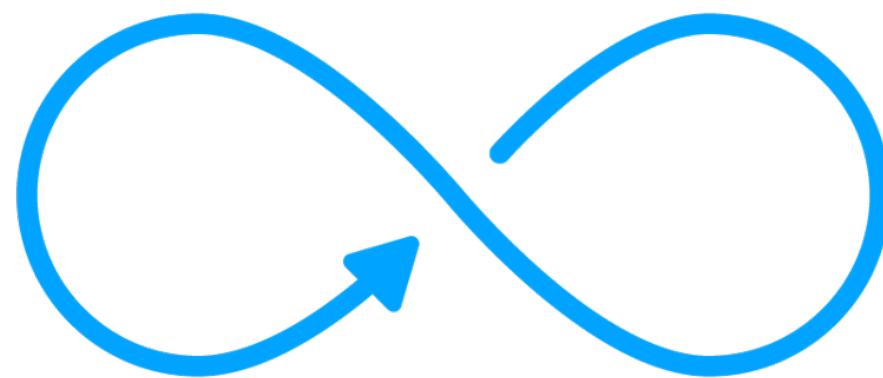


Phil Karlton and Leon Bambrick

“There are 2 hard problems
in computer science: cache
invalidation, naming things,
and off-by-1 errors.”



Debugging Loops



Off-by-1 errors

- Forgetting to add equals
 - $<$ vs \leq
- Starting / stopping on wrong index
 - 0 or 1 based?

Infinite loops

- Incorrect control variable update
 - Incrementing instead of decrementing
- Wrong test expression
- Not initializing or updating control variable



Variables

Case Sensitive

value != Value

Scope

Class

Method

Control statement



Demo



Bug reports

Find and fix errors in code



Accounting has detected an
unusual drop in shipping
charges collected.



Orders should only have one tag applied to them, but several have had multiple tags attached.



Reports for top orders are missing items that should be on the list according to inventory.



All of the top order lists are the same length. For example, requests for lists of 2 or 5 items both return 3 items.



Brian Kernighan

“Debugging is twice as hard as writing the code in the first place. Therefore, if you write the code as cleverly as possible, you are, by definition, not smart enough to debug it.”





Organization

- Small focused methods vs. one giant method
- Group with packages and classes

Whitespace

- Blank lines, indentation, spaces
- See structure and flow of code

Descriptive variable names

- Represent what is being stored
 - orders vs. array
- Avoid excessive abbreviations



Descriptive Variable Names

```
public double calcTot(double prc, double rt, double dr){  
    double disc = prc * dr;  
    double subTot = prc - disc;  
    double tx = subTot * rt;  
    double tot = subTot + tx;  
    return tot;  
}
```

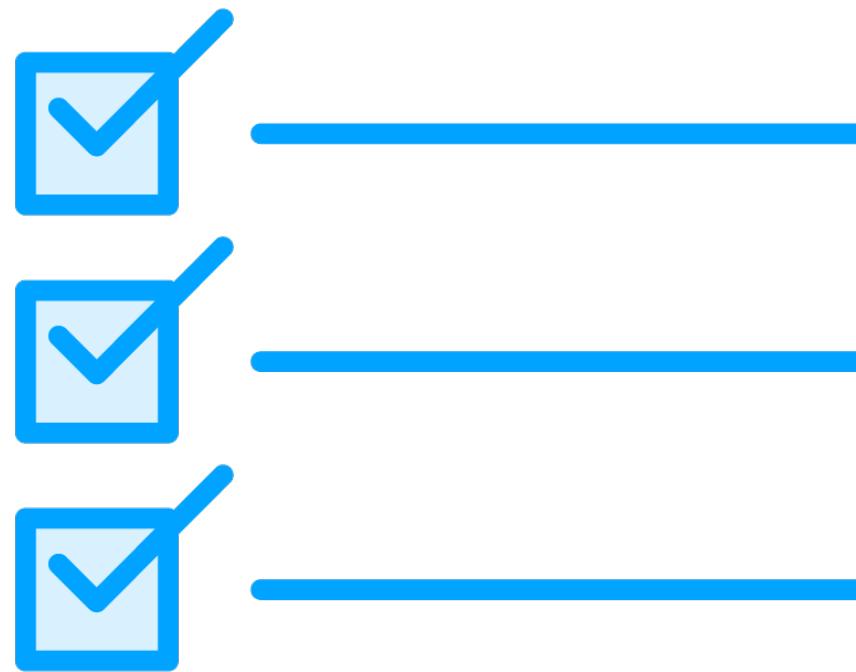
**Extra mental burden to figure
out what this is doing**

```
public double calculateTotal(double price, double taxRate, double discountRate){  
    double discount = price * discountRate;  
    double subTotal = price - discount;  
    double tax = subTotal * taxRate;  
    double total = subTotal + tax;  
    return total;  
}
```

Easy to read = easy to debug



Coding Standards



Conventions and practices for code

- Maintainability

Consistent experience throughout codebase

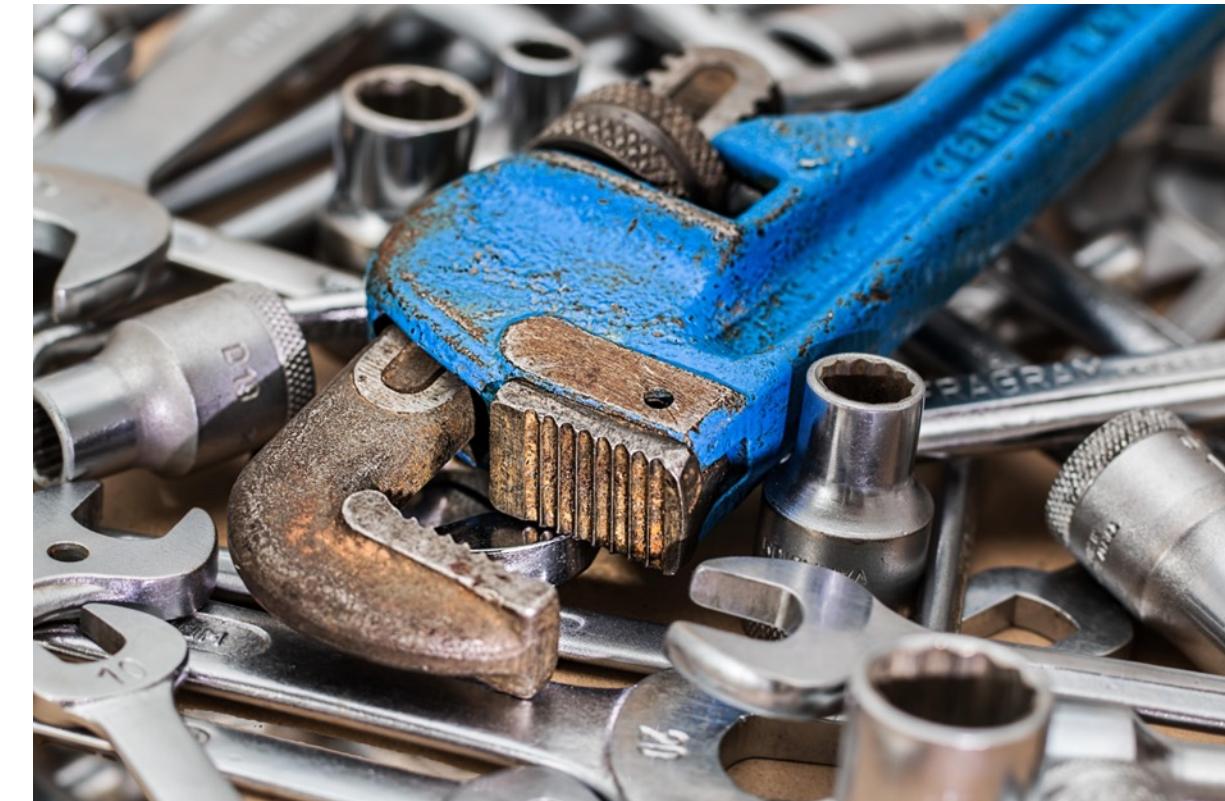
Leverage tools to define and auto-format



Standard Libraries



Build your own
Longer to write
More potential bugs



Java libraries
Faster to implement
Wide use = already debugged





More Information

[Collections in Java](#)

[Refactoring to SOLID Java](#)

[Java Design Patterns](#)

[Java Best Practices](#)



Summary



Common mistakes

- if, switch, loops

Consistent, easy to understand code

- Fewer bugs, faster bug fixes



Up Next:

Find the Problem, Fix the Problem

