



CMSC 131

Fall 2018

Other Announcements

- Project #1 (Orioles Baseball) has been posted.
 - Common errors
 - Style

Recall: “else if” style

One more example:

- `Cascading3.java`

Programming Errors

- Syntax Errors
- Semantic Errors
- Logical Errors

- Debugging
 - Later we'll learn the “debugger”
 - For now, lot's of “trace outputs”

- Advice: Write code incrementally and test, test, test!

Scope of Local Variables

- Local variables have “block scope”. What is this?
- Will the following work?

```
if (x < y) {  
    int z = 4;  
}  
  
if (y < 12) {  
    z = 8;  
}
```

Regarding Initialization of Variables

Examples:

- Initialization1.java
- Initialization2.java
- Initialization3.java

Java Identifiers

Rules

- Can only use: A–Z a–z 0–9 _ \$
- Cannot begin with a digit (0–9)
- Must avoid keywords

Style

- For variables, methods, and packages use “lower camel case”
numberOfCats, ambientTemperature, rows, centerOfGravity...
- For classes and interfaces use “upper camel case”
InputStream, Scanner, String, StringBuffer...

Choosing Meaningful Identifiers

Identifier should be made of English word(s) describing the role of the entity

- Can we abbreviate?
- Can we ever use single-letter identifiers (*i*, *j*, *x*, *y*)?
- Can we be too verbose?

Named Constants

- What is a “named constant”?
- When should we use them (and why is this important)?
- Style:
 MAX_WEIGHT, PI, SPEED_OF_LIGHT, ...
- `final` variables

Examples: WithoutNamedConstants.java,
 WithNamedConstants.java

See: www.whereisroadster.com

Style Requirements for Project #1

- Good variable names
 - Meaningful identifiers
 - Camel Case for conventional variables
- Symbolic constants
 - Well-chosen
 - Named correctly
- Proper braces
- Proper indentation
- No lines longer than 80 characters! (Sorry.)
- Try to eliminate duplicative code. (You will not be penalized for it yet, but try.)

Repetition

Why do we need this?

Two basic techniques:

- Loops (starting today)
- Recursion (later)