



CMSC 131

Object-Oriented Programming I

break and continue

**Dept of Computer Science
University of Maryland College Park**

Slides based on material provided by Ben Bederson, Bonnie Dorr, Fawzi Emad, David Mount, Nelson Padua-Perez, Jan Plane, Pedram Sadeghian, and various TAs over the years

break Statement

- break can be used to **exit immediately** from the current loop (while, do-while, for loop),
- E.g., “Read numbers from input until negative number encountered”
 - **Example:** Break.java
 - Loop only terminates when break executed
 - This only happens with a value < 0
 - **break** always breaks to first enclosing loop
- How about breaking from nested loops?
 - You can have a flag in both loops that control when to stop

Warning about Break

- **Undisciplined use of break can make loops impossible to understand**
 - Termination of loops without a break can be understood by looking at the expressions associated with the iteration statement
- Rule of thumb: use break only when the loop condition is always true
 - i.e., **break** is the only way to terminate the loop
- When you use **break**, make sure it has a good comment explaining what is happening

continue Statement

- **continue** can also be used to affect loops (while, do-while, for loop)
- **In a while and do-while**
 - Control immediately jumps to the boolean expression
- In a **for** loop
 - Causes control to immediately jump to the update statement (the third component of the **for** loop)
- **Example:** ContinueAndDo.java, ContinueAndWhile.java, ContinueAndFor.java
- **continue** should be avoided
 - Confusing
 - Easy equivalents exist
 - Included in Java mainly for historical reasons
- When you use it, make sure it has a good comment explaining what is happening
- When a **for** loop is not the same as the corresponding while loop?