

# 10. Branching Statements

Java

**Fall 2009**

*Instructor: Dr. Masoud Yaghini*

## Outline

---

- The break Statement
- The continue Statement
- The return Statement
- References

# The break Statement



# The break Statement

- You saw the `break` in the `switch` statement.
- You can also use a break to terminate a `for`, `while`, or `do-while` loop
- Example:
  - [BreakDemo.java](#)

# The break Statement

- This program searches for the number 12 in an array.
- The `break` statement terminates the `for` loop when that value is found.
- Control flow then transfers to the print statement at the end of the program.
- This program's output is:  
Found 12 at index 4

# The continue Statement



# The continue Statement

- The `continue` statement skips the current iteration of a `for`, `while`, or `do-while` loop.
- The `continue` skips to the end of the innermost loop's body and evaluates the `boolean` expression that controls the loop.
- Example:
  - [ContinueDemo.java](#)

# The continue Statement

- `ContinueDemo` steps through a `String`, counting the occurrences of the letter "p".
- If the current character is not a `p`, the `continue` statement skips the rest of the loop and proceeds to the next character.
- If it is a `p`, the program increments the letter count.
- Here is the output of this program:  
`Found 9 p's in the string.`





# The return Statement



# The return Statement

- The last of the branching statements is the **return** statement.
- The **return** statement exits from the current method, and control flow returns to where the method was invoked.
- The **return** statement has two forms:
  - one that returns a value
  - one that doesn't returns a value
- To return a value, simply put the value (or an expression that calculates the value) after the return keyword.

# The return Statement

- Example:

```
return ++count;
```

- The data type of the returned value must match the type of the method's declared return value.
- When a method is declared `void`, use the form of return that doesn't return a value.

```
return;
```

- The calling an object's methods will be discussed later.



# References



### References

- S. Zakhour, S. Hommel, J. Royal, I. Rabinovitch, T. Risser, M. Hoeber, **The Java Tutorial: A Short Course on the Basics**, 4th Edition, Prentice Hall, 2006. (Chapter 3)

***The End***

