Nesting

✴ remember that conditional statements and loops contain single statements (or single block statements)
✴ this statement can itself be a conditional or loop
✴ a conditional or loop that occurs inside another conditional or loop is called a nested conditional or loop

An Example

✴ code that outputs which of two numbers is smaller:

```java
if (x < y)
    Output.showMessage("x is smaller");
else
    Output.showMessage("y is smaller");
```

✴ problem: what if x and y are equal?

Pseudocode

✴ pseudocode for what we want:

```java
if x is equal to y
    output "equal"
else
    output which is smaller
```
Building the Code

✴ start with:
if (x == y)
   Output.showMessage("equal");
else
   output which is smaller

Example Code

✴ becomes:
if (x == y)
   Output.showMessage("equal");
else
   if (x < y)
      Output.showMessage("x is smaller");
   else
      Output.showMessage("y is smaller");

✴ note no {}’s (could use but not needed)

Abstraction

✴ abstraction is the selective hiding of details
✴ can write pseudocode at different levels of abstraction
• what matters is that you can get from the pseudocode to Java source code

Abstraction

✴ EX:
if x is less than y
   output "x is smaller"
else
   output "y is smaller"

✴ can be abstracted as:
output which is smaller
Abstraction & Nesting

✴ abstraction is the key to writing complicated nesting constructs
✴ view entire nested conditional/loop as a single “chunk” relative to what it is nested in

Formatting Nested If-Else

✴ EX: given a numeric grade (0.0 - 10.0), output the corresponding letter grade

```java
if (grade >= 9.0)
    Output.showMessage("A");
else
    if (grade >= 8.0)
        Output.showMessage("B");
    else
        if (grade >= 7.0)
            Output.showMessage("C");
        else
            if (grade >= 6.0)
                Output.showMessage("D");
            else
                Output.showMessage("F");
```

Formatting Nested If-Else

```java
if (grade >= 9.0)
    Output.showMessage("A");
else
    if (grade >= 8.0)
        Output.showMessage("B");
    else
        if (grade >= 7.0)
            Output.showMessage("C");
        else
            if (grade >= 6.0)
                Output.showMessage("D");
            else
                Output.showMessage("F");
```
### Formatting Nested If-Else

```java
if (grade >= 9.0)
    Output.showMessage("A");
else if (grade >= 8.0)
    Output.showMessage("B");
else if (grade >= 7.0)
    Output.showMessage("C");
else if (grade >= 6.0)
    Output.showMessage("D");
else
    Output.showMessage("F");
```

### Formatting Nested If-Else

```java
if (grade >= 9.0)
    Output.showMessage("A");
else if (grade >= 8.0)
    Output.showMessage("B");
else if (grade >= 7.0)
    Output.showMessage("C");
else if (grade >= 6.0)
    Output.showMessage("D");
else
    Output.showMessage("F");
```

### Formatting Nested If-Else

```java
if (grade >= 9.0)
    Output.showMessage("A");
else if (grade >= 8.0)
    Output.showMessage("B");
else if (grade >= 7.0)
    Output.showMessage("C");
else if (grade >= 6.0)
    Output.showMessage("D");
else
    Output.showMessage("F");
```

### Formatting Nested If-Else

```java
EX: our code from earlier
if (x == y)
    Output.showMessage("equal");
else
    if (x < y)
        Output.showMessage("x is smaller");
    else
        Output.showMessage("y is smaller");
```
Formatting Nested If-Else

Could be formatted:

```java
if (x == y)
    Output.showMessage("equal");
else if (x < y)
    Output.showMessage("x is smaller");
else
    Output.showMessage("y is smaller");
```

More Abstraction

**EX:**

- declare a radius variable
- get radius and assign to radius variable
- can be abstracted as:
  - input radius

More Abstraction

**EX:**

steps in getting a valid value into a variable:

- declare variable
- get value and assign to variable
- validate value
- echo value

- can be abstracted as:
  - input valid variable

Example

**EX abstract pseudocode:**

- input positive # of cats

- would translate into the following Java:

```java
int numCats;
numCats = Input.readInt("Enter # of cats: ");
while (numCats < 0)
    numCats = Input.readInt("NEGATIVE; try again: ");
Output.showValue("# of cats: ", numCats);
```
Another Nesting Example

Write a program that calculates and outputs the areas of circles with user-input radii. Classify each area as “big” (greater than 100), “medium” (less than or equal to 100 and greater than 50), or “small” (less than or equal to 50).

Pseudocode

input positive # of circles
count = 0
while count less than # of circles
    input positive radius
    calculate and output area
    classify area
    count = count + 1

see Circles2.java for the source code

Module 13 Vocabulary

nested conditional or loop
abstraction

Questions?

Email Kevin at sahrk@sou.edu