Loops

April 28, 2004

Increment and decrement operators

- Shorthand for saying “add one to this variable” and “subtract one from this variable”

- Example:

```java
int i = 10;
i++;  // equivalent to i = i + 1;
System.out.println(i);  // i = 11
i--;  // equivalent to i = i - 1;
System.out.println(i);  // i = 10;
```

Looping gotchas

- Infinite loop
- Overflow
- Using real numbers
- Off-by-one
- Other logic mistakes

Last time....

- Conditionals: nested if's and switch statements
- Short-circuit evaluation
- Loops:
  - definition
  - while loop
  - the curse of the infinite loop
Overflow

- Usually occurs when you have an infinite loop and are using `double` or `float` variables
- The program will continue, but the variable will be set to infinity
- Integers do not overflow
  - instead, they “wrap around” and become negative.

Problem: Sometimes a 1 is not really 1....

- Computers store real numbers as an *approximate* value
  - real numbers are rounded at some point
- Numbers may not be what we think they are
  - e.g., “1” may actually be stored as “1.0000000000001” or “0.99999999999999”
- Lesson: avoid testing for equality with real numbers
  - in the previous example, use `while (num >= 0)`

Using real numbers

- Be careful when you are using real numbers in `<condition>`!
- Example:

```java
double num = 100.0;
while (num != 0.0) {
    num = num - 1.0000000000000001;
}
```

Off-by-one errors

Q: How would you write a loop that repeats 10 times?

```java
count = 0;
while (count < 10) {
    count++;
}
```

```java
count = 0;
while (count <= 10) {
    count++;
}
What do the previous examples show?

- The first loop executes 10 times
- The second loop executes 11 times
- Be careful when determining <condition>!

A previous example, revisited

```java
String rateStr = JOptionPane.showInputDialog(null, "Enter an interest rate: ");
rate = Double.parseDouble(rateStr);
while (rate < 0 || rate > 100) {
    rateStr = JOptionPane.showInputDialog(null, "You entered an invalid rate. Please enter a rate between 0 and 100:");
    rate = Double.parseDouble(rateStr);
}
...
```

What's the problem with the previous example?

- 2 lines are repeated:
  ```java
  rateStr = JOptionPane.showInputDialog(null, "Enter an interest rate: ");
  rate = Double.parseDouble(rateStr);
  ```
- This is because we have a pre-test loop
  - need to test <condition> before the loop is executed
  - need to initialize <condition> before we can test it
- Q: is there a way we can prevent having to repeat lines of code?
  - thought: we need a post-test loop here

do-while loop

- “Do this while some condition is true”
- The test follows the loop
- The loop will execute at least once
  - a while loop might never execute, if the initial condition evaluates to false
Rewrite the previous example using do-while

double rate = -1.0;
String rateStr;

do {
    rateStr = JOptionPane.showInputDialog(null, "Enter a rate between 0 and 100:");
    rate = Double.parseDouble(rateStr);
} while (rate < 0 || rate > 100)

...

Another example

```java
int count = 1, sum = 0;
double rate;
String rateStr;
do {
    rateStr = JOptionPane.showInputDialog(null, "Enter a rate between 0 and 100:");
    rate = Double.parseDouble(rateStr);
} while (rate < 0 || rate > 100)
```

```java
int num = 0, prod = 1; String numStr;
do {
    numStr = JOptionPane.showInputDialog(null, "Enter a number greater than 0:");
    num = Integer.parseInt(numStr);
    if (num < 0) {
        System.out.println("Thanks for playing. Bye!");
    } else if (num == 0) {
        System.out.println("Product is zero. Exiting.");
    } else { 
        prod = prod * num;
        if (prod > 5000) {
            System.out.println("Product exceeds 5000. Exiting.");
        }
    }
} while (num > 0 && num != 0 && prod <= 5000);
```