Introduction to Programming

with Java, for Beginners

Loops (while & for)

A “Loop”

- A simple but powerful mechanism for “making lots of things happen!”
- Performs a statement (or block) over & over
- Usually set up to repeat an action until some condition is satisfied
- Computing Scenarios Examples
  - Run an application until user hits “quit” button
  - Deal card hands until game over

Syntax of the while statement

```java
while (condition) {
    statement(s)
}
```

- `condition` is a true/false (boolean) expression
- If `condition` is initially false, the statement is never executed
- If `condition` is true, `statement` is executed and `condition` is re-evaluated
- The `statement` should eventually make the loop stop

How can we help Bart?

I will not surf in class
I will not surf in class
I will not surf in class
I will not surf in class
I will not surf in class
I will not surf in class
I will not surf in class
I will not surf in class
I will not surf in class
...
A *while* Loop to Print Numbers

```java
// Print the numbers 1 thru 10
int x = 1;
while (x <= 10) {
    System.out.println(x);
    x = x + 1;
}
```

- What happens if you forget the statement `x = x + 1`?
- We print value 1 forever
- Known as *infinite* loop

More *Infinite* Loops

```java
// Some infinite loops are intentional
boolean notQuitKey = true;
while (notQuitKey) {
    statement(s)
}
```

- Others are not
- ```java
  int x = 5;
  while (x < 10) {
      statement(s) which don’t change x
  }
```  

Compute Square of first 10 numbers

```java
//In Square.java
int num = 1;
int sqNum = 0;
while (num <= 10) {
    sqNum = num * num;
    System.out.println(num + " " + sqNum);
    num = num + 1;
}
```

**For Loop**

```java
for (init; end-test; re-init) {
    statement
}
```

- Executes loop body (statements within {}) as long as `end-test` evaluates to TRUE
- Initialization and re-initialization code included in loop statement
- Note: Test is evaluated **before** executing loop body
### While vs. For

<table>
<thead>
<tr>
<th>Code</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>int x = 1;</td>
<td></td>
</tr>
<tr>
<td>while (x &lt;= 10){</td>
<td></td>
</tr>
<tr>
<td>System.out.println(x);</td>
<td>An example of a while loop that has this pattern</td>
</tr>
<tr>
<td>x = x + 1;</td>
<td></td>
</tr>
<tr>
<td>}</td>
<td></td>
</tr>
<tr>
<td>for (int x = 1; x &lt;= 10; x = x + 1){</td>
<td></td>
</tr>
<tr>
<td>System.out.println(x);</td>
<td>A for loop that does the same thing</td>
</tr>
<tr>
<td>}</td>
<td></td>
</tr>
</tbody>
</table>

Note: *For* loops are used generally for bounded iteration

### Summary of Loops

<table>
<thead>
<tr>
<th>Type of Loop</th>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>while</td>
<td>`while (condition){</td>
</tr>
<tr>
<td></td>
<td>statement(s)</td>
</tr>
<tr>
<td></td>
<td>}</td>
</tr>
<tr>
<td>for</td>
<td>`for (expr1; condition; expr3){</td>
</tr>
<tr>
<td></td>
<td>statement(s)</td>
</tr>
<tr>
<td></td>
<td>}</td>
</tr>
</tbody>
</table>

*Note: For loops are used generally for bounded iteration*