Loops

Syntax of the while statement

while (condition) {
  statement(s)
}

For while statements
- condition is a true/false (non-zero/zero) expression
- If condition is initially false, the statement is never executed
- If condition is true, statement(s) is executed and condition is re-evaluated
- One of the statements should eventually make the loop stop

A while Loop to Print Numbers

// Print the numbers 1 thru 10
int x = 1;
while (x <= 10){
  printf(“%d”, x);
  x = x + 1;
}

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

Infinite Loops

The following loop will never terminate:

int x = 0;
while (x < 10) {
  printf(“%d”, x);
}

Loop body does not change condition...
- ...so test is never false
- Common programming error that can be difficult to find

Sometimes you might start out with true case:

int quitKeyNotPressed = 1;
while (quitKeyNotPressed){
  statement(s)
  //remember that there is no boolean type in C
  //at some you make variable 0, so the loop terminates
}
For Loop

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

For loop:
- Executes loop body 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

Difference in Java vs. C
- Loop variable in C needs to be declared outside the for statement

Another example for Loop

```c
/* -- what does this one output? -- */
char letter = 'a';
int c;
for (c = 0; c < 26; c++) {
    printf("%c
", letter + c);
}
```

While vs. For

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

For vs. While

In general:

For loop is preferred for counter-based loops
- Explicit counter variable
- Easy to see how counter is modified each loop

While loop is preferred for sentinel-based loops
- E.g. eof = end of file or \n = newline character

Note: Either kind of loop can be expressed as other, so its really a matter of style and readability
Do-While

do{
    statement(s);
} while (condition);

Do-while
- Executes loop body as long as test evaluates to TRUE (non-zero).
- Note: Test is evaluated after executing loop body. Therefore the statements inside the loop body are executed at least once.