



Java 4: Loops

Repeating code with for and while, and catching loop bugs

NAME _____

DATE _____

CLASS _____

Repeating Code with Loops

A **loop** repeats a block of code. Each trip through the block is one **iteration**. A **while loop** checks its condition before every iteration and keeps going as long as the condition is `true`: `while (n < 10) { ... }`.

A **for loop** puts three parts in one line: `for (int i = 1; i <= 5; i++)`. The **initialization** `int i = 1` runs once, at the start. The **condition** `i <= 5` is checked before each iteration; as soon as it is `false`, the loop stops. The **update** `i++` runs after each iteration. `i++` adds 1 to `i`, `i--` subtracts 1, `i += 3` adds 3, and `i -= 3` subtracts 3.

An **accumulator** is a variable that collects a running total, like `sum = sum + i;`. An **off-by-one error** makes a loop run one time too many or too few, often from using `<` when you meant `<=`. If the condition never becomes `false`, the loop never stops: that is an **infinite loop**.

1 Parts of a for Loop

```
1 for (int i = 0; i < 4; i++) {  
2     System.out.print(i + " ");  
3 }
```

a. Copy each part of the loop onto its line.

initialization: _____

condition: _____

update: _____

b. Write the output.

c. How many **iterations** does this loop run?

d. What is the value of `i` when the condition becomes `false` and the loop stops?

2 Trace the Loop

a. Write one row each time the condition is checked: the value of `i`, whether `i <= 4` is true or false, and the **accumulator** `sum` after the body runs (a dash if the body does not run).

```
1 int sum = 0;  
2 for (int i = 1; i <= 4; i++) {  
3     sum = sum + i;  
4 }  
5 System.out.println("Sum: " + sum);
```

b. What does line 5 print?

i	i <= 4 ?	sum after body



Java 4: Loops (page 2)

3 Predict the Output

What prints? (Hint: `i -= 3`)

```
1 for (int i = 10; i > 0; i -= 3) {  
2     System.out.print(i + " ");  
3 }
```

5 Debug It!

a. Liam wants to print 1 to 10, but his loop stops at 9. What kind of error is this? Write the fixed first line.

```
1 for (int i = 1; i < 10; i++) {  
2     System.out.println(i);  
3 }
```

4 Count the Iterations

Each loop body prints `Hi` once. How many times does `Hi` print?

Loop	Times
a. <code>for (int i = 0; i < 5; i++)</code>	
b. <code>for (int i = 1; i <= 5; i++)</code>	
c. <code>for (int i = 0; i <= 5; i++)</code>	

b. Zoe's countdown prints `5` forever. Why? What line is missing?

```
1 int count = 5;  
2 while (count > 0) {  
3     System.out.println(count);  
4 }  
5 System.out.println("Liftoff!");
```

6 Write Your Own Loops

a. Fill in the blanks so the loop prints `2 4 6 8 10`. Write the three missing pieces.

```
for (int i = __A__; i __B__ 10; i __C__) {  
    System.out.print(i + " ");  
}
```

A: _____

B: _____

C: _____

b. Write a `while` loop that counts down from 10 to 1, printing one number per line, then prints `Go!`.