Name: _____ Session: AM / PM

Circle one

Chapter 4 - Day 2

for Loops, Totals, and Nested Loops

THE FOR LOOP

How a for loop works: A for loop puts three parts in its header. The initialization runs _____, before anything else. The test is checked before every iteration, and the loop ends when it is false. The update runs after each iteration. Like while, a for loop is a _____ loop.

Term	Definition / memory cue
for loop	
count-controlled loop	
initialization	
test	
update	

Two rules: Don't change the control variable inside the body; the update already does that. A variable declared in the header exists only _____.

F1 Counting up and counting down

`F1_ForBasics.java`

What changed: Day 1's SignalPing written as a for loop, then a countdown that steps by 2.

```
3         // Day 1's SignalPing, written as a for loop
4         for (int ping = 1; ping <= 3; ping++) {
5             System.out.println("Ping " + ping);
6         }
7         System.out.println("Signal check done.");
8
9         // Counting down by 2s
10        for (int t = 10; t >= 0; t -= 2) {
11            System.out.println("T-minus " + t);
12        }
13        System.out.println("Liftoff!");
```

1. Line 4's initialization is

2. Line 10's update subtracts _____ each iteration

3. The countdown prints _____ T-minus lines

F2 A bound the user types

F2_StageTable.java

What changed: The user types how many stages to show. That number becomes the loop's upper bound.

```
5         Scanner keyboard = new Scanner(System.in);
6         System.out.print("How many stages? ");
7         int stages = keyboard.nextInt();
8
9         // The number the user typed sets the upper bound
10        for (int stage = 1; stage <= stages; stage++) {
11            System.out.printf("Stage %d: %d kN%n", stage, stage * 250);
12        }
13        System.out.println("Table complete.");
```

1. The upper bound is the variable

2. Type 0, and the body runs _____ times

F3 Walking through a String

F3_CallsignLetters.java

What changed: The loop visits every index of a String, from 0 up to one less than its length.

```
3         String callsign = "ORION";
4
5         // Index 0 up to length() - 1
6         for (int i = 0; i < callsign.length(); i++) {
7             System.out.println(i + ": " + callsign.charAt(i));
8         }
9         System.out.println("Letters: " + callsign.length());
```

1. The first index is

2. The last index is _____, one less than length()

3. With <= on line 6, charAt(5) would

Finger vote. Write your vote (1-4), then what actually happened.

Version	The change	Vote	What happened
F4a	`ping++;` also inside the body (bound is 5)		
F4b	Prints `ping` after the loop		

RUNNING TOTALS AND SENTINELS

Accumulators: An accumulator keeps a running total. Declare it _____ the loop, start it at 0, add to it inside the loop with +=, and use it after the loop. A counter is similar, but it adds 1 each time (`count++`).

Term	Definition / memory cue
running total	
accumulator	
counter	
sentinel value	

R1 A total of three readings

`R1_FuelTotal.java`

What changed: A for loop reads three tank readings and adds each one to total, then prints the total and the average.

```
5         Scanner keyboard = new Scanner(System.in);
6         double total = 0; // accumulator starts at 0
7
8         for (int tank = 1; tank <= 3; tank++) {
9             System.out.print("Tank " + tank + " fuel (kg): ");
10            double fuel = keyboard.nextDouble();
11            total += fuel;
12        }
13
14        System.out.printf("Total fuel: %.1f kg%n", total);
15        System.out.printf("Average per tank: %.1f kg%n", total / 3);
```

1. The accumulator is

2. Line _____ adds each reading

3. Line 15 divides by _____, the number of readings

Sentinel values: A sentinel is a special value that ends the input. Read once before the loop (a priming read), loop while the value is not the sentinel, and read again at the end of the body. The sentinel itself is never _____. Choose a sentinel that can never be _____.

R2 Stop at -1 `R2_CrateSentinel.java`

What changed: Reads crate weights until the user types -1, counting the crates and totaling their weight.

```
6         int count = 0;    // counter
7         double total = 0; // accumulator
8
9         // Priming read: -1 is the sentinel
10        System.out.print("Crate weight (-1 to stop): ");
11        double weight = keyboard.nextDouble();
12
13        while (weight != -1) {
14            count++;
15            total += weight;
16            System.out.print("Crate weight (-1 to stop): ");
17            weight = keyboard.nextDouble();
18        }
19
20        System.out.println("Crates: " + count);
21        System.out.printf("Total weight: %.1f kg%n", total);
```

1. The sentinel is

2. Line _____ counts each crate

3. If the first input is -1, line 20 prints

Pair trace R2 with the scripted input 50, 75, 25, -1: one row per condition check.

weight	weight != -1	count after the body	total after the body
50			
		(body skipped)	(body skipped)

NESTED LOOPS

How nested loops work: In a nested loop, the inner loop runs _____ for every iteration of the outer loop, and its control variable starts over each time. The total is the outer count _____ the inner count: a 12-hour clock has 12 × 60 × 60 = 43,200 seconds.

Term	Definition / memory cue
nested loop	
outer loop	
inner loop	

N1 Days and shifts — N1_ShiftSchedule.java

What changed: The outer loop counts days and the inner loop counts shifts in each day. A counter adds up every shift.

```
3         int shifts = 0;
4
5         // Outer loop: days. Inner loop: shifts each day.
6         for (int day = 1; day <= 3; day++) {
7             for (int shift = 1; shift <= 2; shift++) {
8                 System.out.println("Day " + day + ", shift " + shift);
9                 shifts++;
10            }
11        }
12        System.out.println("Total shifts: " + shifts);
```

1. The inner loop runs _____ times each day

2. Line 8 runs _____ times in all

Rows and columns: When printing a pattern, the outer loop controls the rows and the inner loop controls the columns. `print` stays on the same line. One `println()` after the inner loop _____.

N2 A block of stars — N2_HullPanels.java

What changed: Three rows of five stars. The inner loop prints one row's stars; line 8 ends the row.

```
3         // Outer loop: rows. Inner loop: columns.
4         for (int row = 1; row <= 3; row++) {
5             for (int col = 1; col <= 5; col++) {
6                 System.out.print("*");
7             }
8             System.out.println();
9         }
```

1. The outer loop controls the

2. Each row has _____ stars

N3 One word changed `N3_HullPanelsBroken.java`

What changed: Line 6 now uses println instead of print. Sketch the output before it runs.

```
3         // Outer loop: rows. Inner loop: columns.
4         for (int row = 1; row <= 3; row++) {
5             for (int col = 1; col <= 5; col++) {
6                 System.out.println("*");
7             }
8             System.out.println();
9         }
```

1. Line 6 now ends the line after

2. Lines printed, counting blank ones:

Sketch N3's output before it runs.

BREAK, CONTINUE, AND CHOOSING A LOOP

break and continue: `break` ends a loop immediately. `continue` skips the rest of the body and goes on to the _____. You will see them in other people's code; in this course, the loop's condition says when the loop ends.

Term	Definition / memory cue
break	
continue	

1. Print stages 1 to 10. Best loop, and why?

2. Read crate weights until -1. Best loop, and why?

3. Show a menu, then ask whether to repeat. Best loop, and why?