

Teaching Unit 3: Loops and Iteration

3.1 For Loops

Solutions

3.1.1 - Trace

Determine the exact output. Explain the Java rule or state changes that produce it.

```
for (int i = 5; i > 0; i -= 2) {  
    System.out.print(i + " ");  
}
```

Output

5 3 1

The body runs for 5, 3, and 1; the next value -1 fails the condition.

3.1.2 - Write code

Write public static int sumEven(int n). Return the sum of even integers from 2 through n inclusive. Assume n is nonnegative.

One valid solution is shown below. Other implementations satisfying the contract are also valid.

```
public static int sumEven(int n) {  
    int sum = 0;  
    for (int i = 2; i <= n; i += 2) sum += i;  
    return sum;  
}
```

Check your code

```
System.out.println(sumEven(0));  
System.out.println(sumEven(7));  
System.out.println(sumEven(8));
```

Expected output

0
12
20

Teaching Unit 3: Loops and Iteration

3.2 While Loops

Solutions

3.2.1 - Trace

Determine the exact output. Explain the Java rule or state changes that produce it.

```
int n = 3;
while (n < 20) n *= 2;
System.out.println(n);
```

Output

24

The updates produce 6, 12, and 24; the condition is checked again after each update.

3.2.2 - Write code

Write public static int digitSum(int n) for $n \geq 0$. Return the sum of its decimal digits, including a return value of 0 for input 0.

One valid solution is shown below. Other implementations satisfying the contract are also valid.

```
public static int digitSum(int n) {
    int sum = 0;
    while (n > 0) {
        sum += n % 10;
        n /= 10;
    }
    return sum;
}
```

Check your code

```
System.out.println(digitSum(0));
System.out.println(digitSum(508));
```

Expected output

0
13

Teaching Unit 3: Loops and Iteration

3.3 Nested Loops

Solutions

3.3.1 - Trace

Determine the exact output. Explain the Java rule or state changes that produce it.

```
for (int r = 0; r < 2; r++) {  
    for (int c = 0; c < 3; c++) System.out.print(r + c);  
    System.out.println();  
}
```

Output

```
012  
123
```

Each row resets c to zero; r + c is numeric addition before printing.

3.3.2 - Write code

Write nested loops that print three rows of stars: one star, then two, then three. Use print inside the inner loop and println after it.

One valid solution is shown below. Other implementations satisfying the contract are also valid.

```
for (int row = 1; row <= 3; row++) {  
    for (int col = 0; col < row; col++) System.out.print("*");  
    System.out.println();  
}
```

Teaching Unit 3: Loops and Iteration

3.4 Accumulators and Counters

Solutions

3.4.1 - Trace

Determine the exact output. Explain the Java rule or state changes that produce it.

```
int product = 1;
for (int n = 2; n <= 4; n++) product *= n;
System.out.println(product);
```

Output

24

A product accumulator starts at the multiplicative identity 1, not 0.

3.4.2 - Write code

Given {3, -2, 0, 7, -1}, count and sum the positive values. Print the count followed by a colon and the sum.

One valid solution is shown below. Other implementations satisfying the contract are also valid.

```
int[] values = {3, -2, 0, 7, -1};
int count = 0;
int sum = 0;
for (int value : values) {
    if (value > 0) { count++; sum += value; }
}
System.out.println(count + ":" + sum);
```