

Recitation 04: Java

In today's syntax recitation, we'll cover some Java problems.

Problems

daysInMonth

Write a function named `daysInMonth` that accepts a month (an integer between 1 and 12) as a parameter and returns the number of days in that month. For example, the call `daysInMonth(9)` returns 30 because September has 30 days. Ignore leap years; assume that February always has 28 days.

JAN=31, FEB=28, MAR=31, APR=30, MAY=31, JUN=30, JUL=31, AUG=31, SEP=30, OCT=31, NOV=30, DEC=31

For example, `main` would call `daysInMonth` in the following way to print the number of days in February:

```
1 main() {
2     int feb = daysInMonth(2);
3     println("There are " + feb + " days in February.");
4 }
5
6 // your daysInMonth method would go here
```

While Loop Programming

- Write a function named `showTwos` that shows the factors of 2 in an integer.

```
1 void showTwos(int n) {
2     println(n + " = ");
3     while (n % 2 == 0) {
4         println("2 * ");
5         n /= 2;
6     }
7     println(n);
8 }
```

- Write a function named `showHailstone` that takes an integer parameter `n` and displays the hailstone sequence starting at `n` and ending with 1. In a hailstone sequence, each value x is followed either by $3x + 1$ if x is odd and $\frac{x}{2}$ if x is even.

```
1 void showHailstone(int n) {
2     print("sequence for " + n + ": " + n);
3     while (n != 1) {
4         if (n % 2 == 0) {
5             n /= 2;
6         }
7         else {
8             n = 3 * n + 1;
9         }
10        print(", " + n);
11    }
12    println();
13 }
```

Random Numbers

- Write a function named `rollSix` that simulates the repeated rolling of one six-sided die until a six is rolled. You should use a `Random` object to give an equal chance of rolling a one through six. Each time the die is rolled, you should display the number seen. When you roll a six, you should print the number of trials taken.

```
1 void rollSix() {
2     int roll = 0; // priming roll to execute the loop at least once
3     int count = 0;
4     Random rand = new Random();
5
6     while (roll != 6) {
7         roll = rand.nextInt(6) + 1;
8         count += 1;
9         println("Rolled: " + roll);
10    }
11    println("You got a six in " + count + " turns.");
12 }
```

- Write a function named `randomWalk` that performs a random one-dimensional walk, reporting each position reached and the maximum position reached during the walk. The random walk should begin at position 0. On each step, you should either increase or decrease the position by 1 (with equal probability). The walk stops when 3 or -3 is hit.

```
1 void randomWalk() {
2     int pos = 0;
3     int max = 0;
4     Random rand = new Random();
5
6     println("position = " + pos);
7     while (-3 < pos && pos < 3) {
8         int flip = rand.nextInt(2);
9         if (flip == 0) {
10             pos += 1;
11         }
12         else {
13             pos -= 1;
14         }
15         max = Math.max(pos, max);
16         println("position = " + pos);
17     }
18     println("max position = " + max);
19 }
```