

Answer Key — Week 6 Parsons Problems

For TAs only — do not distribute

Problem 1 — Rocket countdown

```
public class Countdown {
    static void announce(int n) {
        System.out.println((n == 0) ? "Blastoff!" : n + "...");
    }

    public static void main(String[] args) {
        int start = Integer.parseInt(args[0]);
        for (int i = start; i >= 0; i--) { announce(i); }
    }
}
```

DISTRACTOR `System.out.println((n == 0) ? n + "..." : "Blastoff!");`

The two branches of the ternary are swapped — this prints `"Blastoff!"` for every non-zero number and the countdown text when `n` is 0.

DISTRACTOR `for (int i = start; i >= 0; i++) { announce(i); }`

Increments `i` instead of decrementing it — since `i` starts at `start` and only grows, the loop condition `i >= 0` never becomes false, so this is an infinite loop.

Problem 2 — Cube a number

```
public class CubeNumber {
    static int square(int n) { return n * n; }

    static int cube(int n) {
        return square(n) * n;
    }

    public static void main(String[] args) {
        int n = Integer.parseInt(args[0]);
        System.out.println(cube(n));
    }
}
```

DISTRACTOR `return square(n) * square(n);`

Multiplies `square(n)` by itself, giving `n` to the fourth power instead of the third — cube should combine `square(n)` with one more factor of `n`, not another `square(n)`.

DISTRACTOR `return square(n);`

Forgets the extra factor of `n` — returns `square(n)` alone, which is `n` squared, not cubed.