

Parsons Problems — Week 6

Functions and Libraries · Problem 1 of 2

Problem 1 — Rocket countdown

Rearrange the slips to form a complete Java program that reads a starting number from the command line and counts down to 0, printing "Blastoff!" once it reaches 0, using a helper function `announce` called once per number.

Example: `$ java Countdown 3` produces:

```
3...
2...
1...
Blastoff!
```

2 of the slips are distractors — they look plausible but do not belong in the solution.

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

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

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

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

```
int start = Integer.parseInt(args[0]);
```

```
public class Countdown {
```

```
public static void main(String[] args) {
```

```
static void announce(int n) {
```

```
}
```

```
}
```

```
}
```

Parsons Problems — Week 6

Functions and Libraries · Problem 2 of 2

Problem 2 — Cube a number

Rearrange the slips to form a complete Java program that reads an integer from the command line and prints its cube, computed by a cube function that reuses a square helper function.

Example: \$ java CubeNumber 3 prints 27.

2 of the slips are distractors — they look plausible but do not belong in the solution.

```
System.out.println(cube(n));
```

```
int n = Integer.parseInt(args[0]);
```

```
public class CubeNumber {
```

```
public static void main(String[] args) {
```

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

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

```
return square(n);
```

```
static int cube(int n) {
```

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

```
}
```

```
}
```

```
}
```