Variables, Expressions & Strings

Name: _____ Period: ____ Date: _____

Objectives

- Represent a value with a variable and track its value through assignments (AAP-1.A, AAP-1.B).
- Evaluate arithmetic expressions, including `MOD` (AAP-2.C).
- Evaluate expressions that manipulate strings (AAP-2.D).
- Translate code between App Lab (JavaScript) and AP pseudocode.

Directions: App Lab (JavaScript) is on the left; AP Exam Reference Sheet pseudocode is on the right. On the exam, assignment is written with the ← arrow.

Section 1: Variables and Assignment

1. The same assignment, two notations. Fill in what each stores.

```
var score = 10;    // JavaScript
```

```
score ← 10         (AP pseudocode)
```

After this runs, `score` holds _____.

2. Trace this pseudocode. What is displayed, and why?

```
a ← 1
b ← a
a ← 2
DISPLAY(b)
```

Answer: _____

3. Why do meaningful variable names (like `totalScore`) matter, even though the computer does not care what a variable is called?

Section 2: Arithmetic and MOD

`a MOD b` evaluates to the remainder when `a` is divided by `b`. MOD has the same precedence as × and /.

4. Evaluate each expression.

(a) `17 MOD 5` _____ (b) `20 MOD 4` _____

(c) `7 MOD 10` _____ (d) `2 + 3 * 4` _____

5. A program uses `n MOD 2` to test a number. What does the result tell you about `n` when it equals 0? When it equals 1?

 CC BY-NC 4.0

Section 3: Strings

String **concatenation** joins strings end-to-end. In App Lab, + concatenates strings.

6. Evaluate. Give the exact resulting string, including spaces.

```
var first = "Ada";
var last = "Lovelace";
var full = first + " " + last;
```

`full` holds: _____

7. What is a **substring**? Give an example of a substring of `"computer"`.

Section 4: Translate: JavaScript ↔ Pseudocode

8. Rewrite this App Lab code as AP pseudocode.

```
var x = 7;
var y = x * 2;
console.log(y);
```

---

9. Rewrite this pseudocode as App Lab (JavaScript).

```
count ← 0
count ← count + 1
DISPLAY(count)
```

CC BY-NC 4.0