1. What is the return value of .filter()?

A) A single number (the sum).

B) Boolean (true/false).

C) The first matching element.

D) A new array containing only the elements that passed the test (returned true).

2. You have an array of prices: [10, 20, 30]. You want to discount them by 50%. Which code is best?

A) prices.map(price => price * 0.5)

B) prices.filter(price => price * 0.5)

C) prices.reduce(price => price * 0.5)

D) for (p in prices) { p * 0.5 }

3. Why is the following function NOT a "Pure Function"?

JavaScript

```
let tax = 0.2;

function calculateTotal(price) {

    return price + (price * tax);

}
```

A) It returns a number.

B) It relies on an external variable (tax) that can change, making the output unpredictable.

C) It uses math.

D) tax should be inside the function.

4. You have a list of students. You need to (1) Filter only those who passed, and (2) Map them to just their names. How do you chain this?

A) students.filter(s => s.passed).map(s => s.name)

B) students.map(s => s.name).filter(s => s.passed) (This would fail because 'passed' property is lost after mapping).

C) students.reduce(...)

D) students.forEach(...)

5. Compare .reduce() to a for loop for summing numbers. numbers.reduce((acc, curr) => acc + curr, 0).

A) The loop is shorter.

B) Reduce is slower.

C) The .reduce() method is more "Declarative" (tells *what* to do) vs "Imperative" (tells *how* to do it), making it cleaner for data transformations.

D) Loops are illegal in functional programming.