

Name(s) _____ Period _____ Date _____

Extra Practice - Preconditions and Postconditions



Check for Understanding

What is the purpose of the following method?

```
public boolean mystery(int[] nums) {  
    boolean result = true;  
  
    for (int index = 0; index < nums.length; index++) {  
        if ((nums[index] % 2) == 1) {  
            result = false;  
        }  
    }  
  
    return result;  
}
```

- A. The method determines whether the array nums contains all even numbers.
- B. The method determines whether the array nums contains no even numbers.
- C. The method determines whether the array nums has an even length.
- D. The method determines whether the first element of array nums is even.
- E. The method determines whether the last element of array nums is even.

AP Exam Prep

In the following code segment, assume that the `int` array `nums` has been properly declared with a length > 0 .

```
public boolean hasNegative(int[] nums) {
    for (int index = 0; index < nums.length; index++) {
        if (nums[index] < 0) {
            return true;
        }
        else {
            return false;
        }
    }
}
```

The method `hasNegative()` is intended to return `true` if an array has a negative value, `false` otherwise. Which of the following best describes a scenario under which the method `hasNegative()` will not work as intended?

- A. The array is of length one, and there is a negative value in `nums[0]`.
- B. The array is of length one, and there is a non-negative value in `nums[0]`.
- C. The array is of length two, the first value is non-negative, and the second value is negative.
- D. There are only non-negative values.
- E. There are only negative values.

Extra Practice

Do This:

1. Design a scenario for the Joyful Pastries food truck that could be solved by writing an algorithm using a one-dimensional (1D) array.
2. Write a method to solve it.

Student responses will vary. One possible solution:

The bakery never wants to run out of items, therefore any time the inventory for one item goes below 4, they begin baking more. The owner would like a method that will accept a list of quantities and determine whether they need to begin baking any items.

```
public int lowInventory( int[] quantities) {
    boolean status = false;

    for (int index = 0; index < quantities.length; index++) {
        if (quantities[index] < 4) {
            status = true;
        }
    }

    return status;
}
```