

## Programming Exercise 02:

1. The original code has an issue in the following line:

```
$cost = calculateTotalCost(10.0, 3.0, 0.1);
```

### Explanation:

Here, the second argument (3.0) is passed as a float, but the function definition expects an integer (int \$quantity). This can lead to unexpected behavior or errors in some PHP configurations, as quantity is expected to be an integer value.

### Impact on the Code's Functionality

Passing a float for a parameter that is expected to be an integer may cause a type error depending on the PHP version and strictness settings. In this case, 3.0 should be an integer (3) to properly match the function definition.

2. Corrected Code:

```
<?php
function calculateTotalCost(float $price, int $quantity, float $discountRate): float
{
    // Calculate discount amount only if quantity is 2 or more
    $discount = $quantity >= 2 ? $price * $discountRate : 0;

    // Calculate total cost with discount applied
    $totalCost = $price * $quantity - $discount;

    return $totalCost;
}

// Example usage
$cost = calculateTotalCost(10.0, 3, 0.1); // Corrected the quantity to be an integer (3)
echo "Total cost: $" . number_format($cost, 2, '.', ''); // Format output for currency
?>
```

### Explanation of the Changes:

// The argument `3.0` was changed to `3`, as `quantity` should be an integer.

After the correction, the output will be:

Total cost: \$29.00

### Impact of the Change:

- The change ensures that the quantity is passed as an integer, which is more appropriate for calculations involving quantities.
- The rest of the code works as intended, where the discount is applied if the quantity is 2 or more. The function now correctly calculates and displays the total cost with the discount applied.