**JavaScript foundations: Coding Assignment**

**Instructions**

To complete this problem, you will do the following:

* Complete each of the following functions
* Use the skills and techniques you've learned including:
  + Using let and const
  + Using template strings
  + Using inline if statements, if it makes sense
  + Looping over objects with let/in
  + Destructuring and the rest operator
  + Using function parameter destructuring and defaults
  + Writing readable code

**Data**

There are a few parameters you will see throughout the functions.

**products**

A parameter of products refers to an array of objects. Each object is a product, as described below.

**product**

Each product is an object. It will have the following shape:

{

name: "Court Sneaker",

priceInCents: 8800,

availableSizes: [ 6, 8, 10, 11, 12 ]

}

**cart**

A cart is a collection of products, organized by product name. Each key points towards another object. It will look similar to the following:

{

"Court Sneaker": {

priceInCents: 8800,

quantity: 1

}

}

**printablePrice()**

This function has already been completed for you. It will take a number that represents a product price in cents.

Example:

printablePrice(8800); *//> $88.00*

**chooseItemByNameAndSize()**

This function has three parameters: products, name, and size.

When used correctly, the function will search through all of the products and find a product with a matching name that has the size value inside of the product's availableSizes array. If no match is found for either reason, null should be returned.

Example:

chooseItemByNameAndSize(products, "Court Sneaker", 8); *//> { name: "Court Sneaker", ... }*

chooseItemByNameAndSize(products, "Court Sneaker", 9); *//> null*

chooseItemByNameAndSize(products, "Travel Flip-Flops", 8); *//> null*

**addProductToCart()**

This function has two parameters: product and cart. The cart parameter is optional.

If addProductToCart() is only called with a product, it will create a new cart object and reformat the product, as described above. It will also set the quantity to 1.

console.log(product); *//> { name: "Court Sneaker", ... }*

addProductToCart(product); *//> { "Court Sneaker": { quantity: 1, priceInCents: 8800 } }*

If a cart *is given* and *a new item is being added*, it will add the item and set the quantity to 1.

console.log(cart); *//> { "Black Hat": { quantity: 1, priceInCents: 2300 } }*

console.log(product); *//> { name: "Court Sneaker", ... }*

addProductToCart(product); *//> { "Black Hat": { ... }, Court Sneaker": { ... } }*

If a cart *is given* and *the product being added is already in the cart*, just increase the quantity by 1.

console.log(cart); *//> { "Court Sneaker": { quantity: 1, priceInCents: 8800 } }*

console.log(product); *//> { name: "Court Sneaker", ... }*

addProductToCart(product); *//> { "Court Sneaker": { quantity: 2, priceInCents: 8800 } }*

**calculateTotal()**

This function takes a single parameter of cart. It should loop through the cart and return the total price for everything in the cart. For each distinct item, multiply the quantity by the priceInCents.

console.log(cart); *//> { "Court Sneaker": { quantity: 2, priceInCents: 8800 } }*

calculateTotal(cart); *//> 17600*

If the cart is empty, return 0.

calculateTotal({}); *//> 0*

**printReceipt()**

This function takes a single parameter of cart. If the cart is empty, return null.

printReceipt({}); *//> null*

Otherwise, print out each item in the cart with its quantity, name, and total. Each item should be connected by a \n. Finally, include a total at the end of the receipt.

For example, assume this is the *input* into printReceipt():

{

"Court Sneaker": { quantity: 2, priceInCents: 8800 },

"Black Hat": { quantity: 1, priceInCents: 2300 },

}

This would then be the output:

"2xCourt Sneaker - $176.00\n1xBlack Hat - $23.00\nTotal: $199.00"

**Hint:** You will need to use calculateTotal() and printablePrice() in this function.

**Tips**

* You may complete this challenge on your own machine before uploading it to Qualified.
* Reference the related checkpoint for help on completing this Practice problem.
* If you need help, contact your mentor or speak with your peers in Slack.