Alanis Irizarry
CSC271

Assignment 10: Modularization

A. Reusable code:

The recommendation logic in recommendProgram() chooses a fitness package.
The display section in displayRecommendation() shows the program details on the page.
The payment calculation in payment_calculator.js divides the total price by months.
The input validation checks if all form selections are complete before running calculations.

B. Plan Your Functions:

I plan to turn these sections into smaller functions:

-calculatePayment(price, months) : will take two parameters and return the monthly payment.

-displayRecommendation(packageName, price, duration, features, goal) : will take parameters and show the selected program on the page.

-validateInputs(goal, experience, budget) : will take parameters and return true or false.

-enhancePricingTable() : will not take parameters or return values, but will add interactivity to the table.

C. Document your plan:

1. Each function focuses on one clear job. For example, calculatePayment() does the math, displayRecommendation() handles the visuals, and validateInputs() checks user input. Some functions will return results, while others just update the webpage.
2. I chose to modularize these sections to make the scripts easier to read, update, and fix. It also reduces repeated code and helps keep logic and visuals separate, so future changes will be easier to make.
3. New Functions:
    a. initializeEventListeners() : sets up all event listeners when the page loads (no parameters or return value).
    b. calculateDiscountedPrice(price, discountRate) : returns a discounted price for promotions.
    c. resetCalculator() : clears results on the payment page (no parameters or return value).