

Problem Statement

1. Simple Calculator Program

Design a simple calculator program that performs basic arithmetic operations.

Here's a breakdown of the program's functionality:

1. Display a welcome message: "Simple Calculator."
2. Prompt the user to enter the first numeric value.
3. Prompt the user to enter the desired arithmetic operation (+, -, *, /).
4. Prompt the user to enter the second numeric value.
5. Validate the user inputs to ensure they are valid numeric values.
6. Perform the specified arithmetic operation on the two numbers.
7. If the operation is division (/), check for division by zero and handle it by displaying an error message.
8. Display the result of the calculation.
9. If the entered operation is not one of the supported operations, displays an error message.

2. Discount Eligibility Checker

Create a program to check if a person is eligible for a discount based on their age and customer type. The program should:

1. Display a welcome message: "Discount Eligibility Checker."
2. Prompt the user for their age.
3. Prompt the user for their customer type, specifying whether they are a new or returning customer.
4. Check eligibility for a discount based on the following criteria:
 - If the person is 18 or older:

- New customers and returning customers are both eligible for a discount.
 - Handle invalid customer types and terminate the script if an invalid type is entered.
 - If the person is below 18:
 - Not eligible for a discount.
5. Display the result of discount eligibility:
- If eligible, congratulate the user.
 - If not eligible, inform the user.

3. Weather Clothing Advisor

Develop a program that provides clothing recommendations based on the current temperature in Celsius. The program should:

1. Display a welcome message: "Weather Clothing Advisor."
2. Prompt the user for the current temperature in Celsius.
3. Determine the clothing recommendation using the following criteria:
 - If the temperature is 30 degrees Celsius or higher: "It's hot! Wear light and breathable clothing."
 - If the temperature is between 20 and 30 degrees Celsius: "It's warm. T-shirts and shorts are suitable."
 - If the temperature is between 10 and 20 degrees Celsius: "It's cool. Consider wearing a light jacket."
 - If the temperature is below 10 degrees Celsius: "It's cold. Bundle up with a coat and warm layers."
4. Handle invalid temperature inputs by providing an error message and terminating the script.
5. Display the clothing recommendation based on the temperature.

4. Coffee Order Recommender

Design a program that suggests personalized coffee orders based on user preferences for sweetness and caffeine strength. The system should:

1. Display a welcome message: "Coffee Order Recommender."
2. Prompt the user to enter their preferred sweetness level for coffee, choosing from 'low', 'medium', or 'high'.
3. Prompt the user to enter their desired caffeine strength, choosing from 'low', 'medium', or 'high'.
4. Use if-else statements to recommend a specific coffee based on the entered preferences:
 - If the user prefers low sweetness and low caffeine: Suggest a classic black coffee.
 - If the user prefers medium sweetness and medium caffeine: Recommend a caramel macchiato for a balanced flavor.
 - If the user prefers high sweetness and high caffeine: Suggest indulging in a mocha for a sweet and strong experience.
 - Handle invalid preference inputs with an error message and terminate the script.
5. Display the recommended coffee based on the user's sweetness and caffeine preferences.

5. Fitness Goal Tracker

Create a program that provides personalized workout recommendations based on user fitness goals and available time. The system should:

1. Display a welcome message: "Fitness Goal Tracker."
2. Prompt the user to enter their fitness goal: 'lose weight', 'build muscle', or 'stay fit'.
3. Prompt the user to enter the available time for their workout in minutes.

4. Use if-else statements to recommend a specific workout based on the entered fitness goal and available time:
 - If the goal is to 'lose weight':
 - If the available time is 30 minutes or more, suggest a mix of cardio and strength training.
 - If the available time is less than 30 minutes, recommend starting with a brisk walk or a quick HIIT session.
 - If the goal is to 'build muscle':
 - If the available time is 45 minutes or more, recommend focusing on compound exercises like squats and deadlifts.
 - If the available time is less than 45 minutes, suggest incorporating bodyweight exercises like push-ups and squats.
 - If the goal is to 'stay fit':
 - If the available time is 20 minutes or more, advise opting for a mix of cardiovascular exercises and strength training.
 - If the available time is less than 20 minutes, recommend a short session of yoga or a brisk walk.
5. Handle invalid fitness goal inputs with an error message and terminate the script.
6. Display the personalized workout recommendation based on the user's fitness goal and available time.

6. Budget Calculator

Develop a program for a budget calculator that assesses an individual's financial situation based on their monthly income and expenses. The system should:

1. Display a welcome message: "Budget Calculator."
2. Prompt the user to enter their monthly income.
3. Prompt the user to enter their total monthly expenses.
4. Calculate the remaining budget by subtracting monthly expenses from monthly income.
5. Categorize the user's spending level using if-else statements:
 - If the remaining budget is \$500 or more, advise that they are in a good financial position and suggest saving or investing some money.

- If the remaining budget is between \$200 and \$499, inform them of a moderate budget and recommend being mindful of discretionary spending.
 - If the remaining budget is between \$0 and \$199, indicate a tight budget and advise prioritizing essential expenses.
 - If the remaining budget is negative, warn the user that they have exceeded their budget and prompt them to review their expenses.
6. Display the spending level recommendation based on the user's financial situation.

7. Travel Expense Optimizer

Develop a program, "Travel Expense Optimizer," to assist users in planning and optimizing their travel expenses based on transportation, accommodation, and meal preferences. The system should:

1. Display a welcome message: "Travel Expense Optimizer."
2. Prompt the user to select their mode of transportation (e.g., train, bus, or car).
3. Prompt the user to choose their accommodation preference (e.g., budget, mid-range, or luxury).
4. Specify the user's meal preference (e.g., vegetarian, non-vegetarian).
5. Calculate expenses based on user preferences using if-else statements:
 - For transportation:
 - If the chosen mode is 'train,' set transportation expenses to INR 1500.
 - If the chosen mode is 'bus,' set transportation expenses to INR 1200.
 - If the chosen mode is 'car,' set transportation expenses to INR 2000.
 - Handle invalid transportation mode inputs.
 - For accommodation:
 - If the chosen preference is 'budget,' set accommodation expenses to INR 1000.

- If the chosen preference is 'mid-range,' set accommodation expenses to INR 2000.
 - If the chosen preference is 'luxury,' set accommodation expenses to INR 4000.
 - Handle invalid accommodation preference inputs.
- For meals:
 - If the chosen preference is 'vegetarian,' set meal expenses to INR 500.
 - If the chosen preference is 'non-vegetarian,' set meal expenses to INR 800.
 - Handle invalid meal preference inputs.
6. Calculate the total expense by summing up transportation, accommodation, and meal expenses.
 7. Display an optimized expense breakdown, including transportation, accommodation, meal expenses, and the total expense.

8. Movie Recommendation System

Develop a program that recommends movies based on user preferences. The system should:

1. Display a welcome message: "Movie Recommendation System."
2. Prompt the user to enter their preferred genre of movies.
3. Use if-else statements to suggest movie recommendations based on the entered genre:
 - If the genre is "Action": Recommend action-packed movies with thrilling sequences.
 - If the genre is "Comedy": Suggest light-hearted and humorous movies for entertainment.
 - If the genre is "Drama": Recommend emotionally engaging and thought-provoking dramas.
 - If the genre is "Science Fiction": Suggest futuristic and imaginative sci-fi movies.

- If the genre is not one of the specified options, handle it with an error message and terminate the script.
4. Display the movie recommendation based on the user's preferred genre.

1 Simple JavaScript Calculator

Objective:

- Create a simple JavaScript calculator that takes user input for two numbers and an operation (+, -, *, /) and performs the corresponding arithmetic operation using if-else statements.

Prerequisites:

- Basic knowledge of JavaScript syntax.
- Understanding of prompt and parseFloat functions.
- Familiarity with if-else statements.

Step-by-Step Solution:

Step 1: Welcome Message

```
// Welcome message
console.log("Simple Calculator");
```

Step 2: Get User Input for the First Number

```
// Get user input for the first number
let num1 = parseFloat(prompt("Enter the first number:"));
```

Step 3: Get User Input for the Operation

```
// Get user input for the operation
let operation = prompt("Enter the operation (+, -, *, /):");
```

Step 4: Get User Input for the Second Number

```
// Get user input for the second number
let num2 = parseFloat(prompt("Enter the second number:"));
```

Step 5: Validate Input and Perform Calculation

```
// Perform the calculation
let result;

if (operation === "+") {
  result = num1 + num2;
} else if (operation === "-") {
  result = num1 - num2;
} else if (operation === "*") {
  result = num1 * num2;
} else if (operation === "/") {
  // Check for division by zero
  if (num2 !== 0) {
    result = num1 / num2;
  } else {
    console.log("Error: Division by zero is not allowed.");
  }
}
```

```

    }
} else {
    // Handle invalid operation
    console.log("Invalid operation. Please enter +, -, *, or /.");
}

// Display the result
console.log(`Result: ${num1} ${operation} ${num2} = ${result}`);

```

Step 6: Final Code

```

// Welcome message
console.log("Simple Calculator");

// Get user input for the first number
let num1 = parseFloat(prompt("Enter the first number:"));

// Get user input for the operation
let operation = prompt("Enter the operation (+, -, *, /):");

// Get user input for the second number
let num2 = parseFloat(prompt("Enter the second number:"));

// Perform the calculation
let result;

if (operation === "+") {
    result = num1 + num2;
} else if (operation === "-") {
    result = num1 - num2;
} else if (operation === "*") {
    result = num1 * num2;
} else if (operation === "/") {
    // Check for division by zero
    if (num2 !== 0) {

```

```
    result = num1 / num2;
} else {
    console.log("Error: Division by zero is not allowed.");
}
} else {
// Handle invalid operation
console.log("Invalid operation. Please enter +, -, *, or ./.");
process.exit(); // Terminate the script
}

// Display the result
console.log(`Result: ${num1} ${operation} ${num2} = ${result}`);
```



PRIMATHON

2 Discount Eligibility Checker

Objective:

- Create a simple JavaScript program that checks whether a person is eligible for a discount based on their age and whether they are a new customer or a returning customer. Utilize let, const, operators, and if-else statements.

Prerequisites:

- Basic understanding of JavaScript syntax.
- Familiarity with prompt function for user input.
- Knowledge of let and const for variable declaration.
- Understanding of comparison operators (e.g., >=, ===).
- Ability to use if-else statements for decision-making.

Step-by-Step Solution:

Step 1: Welcome Message

```
// Welcome message
console.log("Discount Eligibility Checker");
```

Step 2: Get User Input for Age

```
// Get user input for age
const age = parseInt(prompt("Enter your age:"));
```

Step 3: Get User Input for Customer Type

```
// Get user input for customer type (new or returning)
const customerType = prompt("Are you a new customer or a
returning customer? (Type 'new' or 'returning'):");
```

Step 4: Check Eligibility for Discount

```
// Declare a variable to store discount eligibility
let discountEligible;

// Use if-else statements to determine discount eligibility
```

```
if (age >= 18) {  
    // Person is 18 or older  
    if (customerType === 'new') {  
        // New customers get a special discount  
        discountEligible = true;  
    } else if (customerType === 'returning') {  
        // Returning customers also get a discount  
        discountEligible = true;  
    } else {  
        // Handle invalid customer type  
        console.log("Invalid customer type. Please enter 'new' or  
'returning'.");  
        process.exit(); // Terminate the script  
    }  
} else {  
    // Person is below 18, not eligible for discount  
    discountEligible = false;  
}
```

Step 5: Display Discount Eligibility Result

```
// Display discount eligibility result  
if (discountEligible) {  
    console.log("Congratulations! You are eligible for a  
discount.");  
} else {  
    console.log("Sorry, you are not eligible for a discount.");  
}
```

Step 6: Final Code

```
// Welcome message  
console.log("Discount Eligibility Checker");
```

```
// Get user input for age
const age = parseInt(prompt("Enter your age:"));

// Get user input for customer type (new or returning)
const customerType = prompt("Are you a new customer or a
returning customer? (Type 'new' or 'returning'):");

// Check eligibility for discount
let discountEligible;

// Use if-else statements to determine discount eligibility
if (age >= 18) {
    // Person is 18 or older
    if (customerType === 'new') {
        // New customers get a special discount
        discountEligible = true;
    } else if (customerType === 'returning') {
        // Returning customers also get a discount
        discountEligible = true;
    } else {
        // Handle invalid customer type
        console.log("Invalid customer type. Please enter 'new' or
'returning'.");
        process.exit() // Terminate the script
    }
} else {
    // Person is below 18, not eligible for discount
    discountEligible = false;
}

// Display discount eligibility result
if (discountEligible) {
    console.log("Congratulations! You are eligible for a
discount.");
} else {
```

```
        console.log("Sorry, you are not eligible for a discount.");
    }
```

3 Weather Clothing Advisor

Objective:

- Create a simple JavaScript program that advises the user on what type of clothing to wear based on the current weather temperature. Utilize let, const, operators, and if-else statements.

Prerequisites:

- Basic understanding of JavaScript syntax.
- Familiarity with prompt function for user input.
- Knowledge of let and const for variable declaration.
- Understanding of comparison operators (e.g., >=, <=).
- Ability to use if-else statements for decision-making.

Step-by-Step Solution:

Step 1: Welcome Message

```
// Welcome message
console.log("Weather Clothing Advisor");
```

Step 2: Get User Input for Temperature

```
// Get user input for temperature
const temperature = parseFloat(prompt("Enter the current
temperature in Celsius:"));
```

Step 3: Determine Clothing Recommendation

```
// Declare a variable to store clothing recommendation
let clothingRecommendation;

// Use if-else statements to determine clothing recommendation
if (temperature >= 30) {
  clothingRecommendation = "It's hot! Wear light and breathable
clothing.";
} else if (temperature >= 20 && temperature < 30) {
  clothingRecommendation = "It's warm. T-shirts and shorts are
suitable.";
} else if (temperature >= 10 && temperature < 20) {
  clothingRecommendation = "It's cool. Consider wearing a light
jacket.";
} else if (temperature < 10) {
  clothingRecommendation = "It's cold. Bundle up with a coat and
warm layers.";
} else {
  // Handle invalid temperature input
  console.log("Invalid temperature input. Please enter a numeric
value.");
  process.exit(); // Terminate the script
}
```

Step 4: Display Clothing Recommendation

```
// Display clothing recommendation
```

```
console.log(clothingRecommendation);
```

Step 5: Final Code

```
// Welcome message
console.log("Weather Clothing Advisor");

// Get user input for temperature
const temperature = parseFloat(prompt("Enter the current
temperature in Celsius:"));

// Declare a variable to store clothing recommendation
let clothingRecommendation;

// Use if-else statements to determine clothing recommendation
if (temperature >= 30) {
  clothingRecommendation = "It's hot! Wear light and breathable
clothing.";
} else if (temperature >= 20 && temperature < 30) {
  clothingRecommendation = "It's warm. T-shirts and shorts are
suitable.";
} else if (temperature >= 10 && temperature < 20) {
  clothingRecommendation = "It's cool. Consider wearing a light
jacket.";
} else if (temperature < 10) {
  clothingRecommendation = "It's cold. Bundle up with a coat and
warm layers.";
} else {
  // Handle invalid temperature input
  console.log("Invalid temperature input. Please enter a numeric
value.");
  process.exit(); // Terminate the script
}

// Display clothing recommendation
```

```
console.log(clothingRecommendation);
```

4 Coffee Order Recommender

Objective:

- Create a simple JavaScript program that recommends the type of coffee based on user preferences such as sweetness and caffeine level. Utilize let, const, operators, and if-else statements.

Prerequisites:

- Basic understanding of JavaScript syntax.
- Familiarity with prompt function for user input.
- Knowledge of let and const for variable declaration.
- Understanding of comparison operators (e.g., ===).
- Ability to use if-else statements for decision-making.

Step-by-Step Solution:

Step 1: Welcome Message

```
// Welcome message
console.log("Coffee Order Recommender");
```

Step 2: Get User Input for Sweetness Preference

```
// Get user input for sweetness preference
const sweetnessPreference = prompt("How sweet do you like your
coffee? (Enter 'low', 'medium', or 'high'):");
```

Step 3: Get User Input for Caffeine Preference

```
// Get user input for caffeine preference
const caffeinePreference = prompt("How strong do you like your
coffee? (Enter 'low', 'medium', or 'high'):");
```

Step 4: Determine Coffee Recommendation

```
// Declare a variable to store coffee recommendation
let coffeeRecommendation;

// Use if-else statements to determine coffee recommendation
if (sweetnessPreference === 'low' && caffeinePreference ===
'low') {
    coffeeRecommendation = "Try a black coffee for a classic
taste.";
} else if (sweetnessPreference === 'medium' &&
caffeinePreference === 'medium') {
    coffeeRecommendation = "A caramel macchiato might be the
perfect balance for you.";
} else if (sweetnessPreference === 'high' && caffeinePreference
=== 'high') {
    coffeeRecommendation = "Indulge in a mocha for a sweet and
strong flavor.";
} else {
    // Handle invalid preference input
    console.log("Invalid preference input. Please enter 'low',
'medium', or 'high'.");
```

```
    process.exit(); // Terminate the script
}
```

Step 5: Display Coffee Recommendation

```
// Display coffee recommendation
console.log(coffeeRecommendation);
```

Step 6: Final Code

```
// Welcome message
console.log("Coffee Order Recommender");

// Get user input for sweetness preference
const sweetnessPreference = prompt("How sweet do you like your
coffee? (Enter 'low', 'medium', or 'high'):").toLowerCase();

// Get user input for caffeine preference
const caffeinePreference = prompt("How strong do you like your
coffee? (Enter 'low', 'medium', or 'high'):").toLowerCase();

// Declare a variable to store coffee recommendation
let coffeeRecommendation;

// Use if-else statements to determine coffee recommendation
if (sweetnessPreference === 'low' && caffeinePreference ===
'low') {
  coffeeRecommendation = "Try a black coffee for a classic
taste.";
} else if (sweetnessPreference === 'medium' &&
caffeinePreference === 'medium') {
  coffeeRecommendation = "A caramel macchiato might be the
perfect balance for you.;"
```

```
    } else if (sweetnessPreference === 'high' && caffeinePreference === 'high') {
        coffeeRecommendation = "Indulge in a mocha for a sweet and strong flavor.";
    } else {
        // Handle invalid preference input
        console.log("Invalid preference input. Please enter 'low', 'medium', or 'high'.");
        process.exit(); // Terminate the script
    }

    // Display coffee recommendation
    console.log(coffeeRecommendation);
```

5 Fitness Goal Tracker

Objective:

- Create a simple JavaScript program that suggests a workout routine based on the user's fitness goal and available time. Utilize let, const, operators, and if-else statements.

Prerequisites:

- Basic understanding of JavaScript syntax.
- Familiarity with prompt function for user input.
- Knowledge of let and const for variable declaration.
- Understanding of comparison operators (e.g., ===, <=).
- Ability to use if-else statements for decision-making.

Step-by-Step Solution:

Step 1: Welcome Message

```
// Welcome message  
console.log("Fitness Goal Tracker");
```

Step 2: Get User Input for Fitness Goal

```
// Get user input for fitness goal  
const fitnessGoal = prompt("What is your fitness goal? (Enter  
'lose weight', 'build muscle', or 'stay fit'):");
```

Step 3: Get User Input for Available Time

```
// Get user input for available time in minutes  
const availableTime = parseInt(prompt("How much time do you have  
for your workout? (Enter the time in minutes):"));
```

Step 4: Determine Workout Recommendation

```
// Declare a variable to store workout recommendation  
let workoutRecommendation;  
  
// Use if-else statements to determine workout recommendation  
if (fitnessGoal === 'lose weight') {  
    if (availableTime >= 30) {  
        workoutRecommendation = "Consider doing a mix of cardio and  
        strength training for an effective weight loss workout.";  
    } else {  
        workoutRecommendation = "You can start with a brisk walk or  
        a quick HIIT session to get started."  
    }  
}
```

```
    } else if (fitnessGoal === 'build muscle') {
        if (availableTime >= 45) {
            workoutRecommendation = "Focus on compound exercises like
squats and deadlifts to build muscle efficiently.";
        } else {
            workoutRecommendation = "Try incorporating some bodyweight
exercises like push-ups and squats for a quick muscle-building
session.";
        }
    } else if (fitnessGoal === 'stay fit') {
        if (availableTime >= 20) {
            workoutRecommendation = "Opt for a mix of cardiovascular
exercises and strength training to maintain overall fitness.";
        } else {
            workoutRecommendation = "Even a short session of yoga or a
brisk walk can help you stay active and fit.";
        }
    } else {
        // Handle invalid fitness goal input
        console.log("Invalid fitness goal input. Please enter 'lose
weight', 'build muscle', or 'stay fit'.");
        process.exit() // Terminate the script
    }
}
```

ACADEMY

Step 5: Display Workout Recommendation

```
// Display workout recommendation
console.log(workoutRecommendation);
```

Step 6: Final Code

```
// Welcome message
console.log("Fitness Goal Tracker");
```

```
// Get user input for fitness goal
const fitnessGoal = prompt("What is your fitness goal? (Enter 'lose weight', 'build muscle', or 'stay fit'):").toLowerCase();

// Get user input for available time in minutes
const availableTime = parseInt(prompt("How much time do you have for your workout? (Enter the time in minutes):"));

// Declare a variable to store workout recommendation
let workoutRecommendation;

// Use if-else statements to determine workout recommendation
if (fitnessGoal === 'lose weight') {
    if (availableTime >= 30) {
        workoutRecommendation = "Consider doing a mix of cardio and strength training for an effective weight loss workout.";
    } else {
        workoutRecommendation = "You can start with a brisk walk or a quick HIIT session to get started.";
    }
} else if (fitnessGoal === 'build muscle') {
    if (availableTime >= 45) {
        workoutRecommendation = "Focus on compound exercises like squats and deadlifts to build muscle efficiently.";
    } else {
        workoutRecommendation = "Try incorporating some bodyweight exercises like push-ups and squats for a quick muscle-building session.";
    }
} else if (fitnessGoal === 'stay fit') {
    if (availableTime >= 20) {
        workoutRecommendation = "Opt for a mix of cardiovascular exercises and strength training to maintain overall fitness.";
    } else {
        workoutRecommendation = "Even a short session of yoga or a
```

```
brisk walk can help you stay active and fit.";  
    }  
} else {  
    // Handle invalid fitness goal input  
    console.log("Invalid fitness goal input. Please enter 'lose  
    weight', 'build muscle', or 'stay fit'.");  
    process.exit(); // Terminate the script  
}  
  
// Display workout recommendation  
console.log(workoutRecommendation);
```

6 Budget Calculator

Objective:

- Create a simple JavaScript program that helps users manage their monthly budget by categorizing expenses into different spending levels. Utilize let, const, operators, and if-else statements.

Prerequisites:

- Basic understanding of JavaScript syntax.
- Familiarity with prompt function for user input.
- Knowledge of let and const for variable declaration.
- Understanding of comparison operators (e.g., >=, <=).
- Ability to use if-else statements for decision-making.

Step-by-Step Solution:

Step 1: Welcome Message

```
// Welcome message  
console.log("Budget Calculator");
```

Step 2: Get User Input for Monthly Income

```
// Get user input for monthly income  
const monthlyIncome = parseFloat(prompt("Enter your monthly  
income:"));
```

Step 3: Get User Input for Monthly Expenses

```
// Get user input for monthly expenses  
const monthlyExpenses = parseFloat(prompt("Enter your total  
monthly expenses:"));
```

Step 4: Calculate Remaining Budget

```
// Calculate remaining budget  
const remainingBudget = monthlyIncome - monthlyExpenses;
```

Step 5: Categorize Spending Level

```
// Declare a variable to store spending level  
let spendingLevel;  
  
// Use if-else statements to categorize spending level
```

```
if (remainingBudget >= 500) {  
    spendingLevel = "You are in a good financial position.  
    Consider saving or investing some money.";  
} else if (remainingBudget >= 200 && remainingBudget < 500) {  
    spendingLevel = "You have a moderate budget. Be mindful of  
    discretionary spending.";  
} else if (remainingBudget >= 0 && remainingBudget < 200) {  
    spendingLevel = "Your budget is tight. Prioritize essential  
    expenses.";  
} else {  
    // Handle negative budget (overspending)  
    console.log("Warning: You have exceeded your budget. Review  
    your expenses.");  
}
```

Step 6: Display Spending Level

```
// Display spending level  
console.log(spendingLevel);
```

Step 7: Final code

```
// Welcome message  
console.log("Budget Calculator");  
  
// Get user input for monthly income  
const monthlyIncome = parseFloat(prompt("Enter your monthly  
income:"));  
  
// Get user input for monthly expenses  
const monthlyExpenses = parseFloat(prompt("Enter your total  
monthly expenses:"));
```

```
// Calculate remaining budget
const remainingBudget = monthlyIncome - monthlyExpenses;

// Declare a variable to store spending level
let spendingLevel;

// Use if-else statements to categorize spending level
if (remainingBudget >= 500) {
    spendingLevel = "You are in a good financial position.
Consider saving or investing some money.";
} else if (remainingBudget >= 200 && remainingBudget < 500) {
    spendingLevel = "You have a moderate budget. Be mindful of
discretionary spending.";
} else if (remainingBudget >= 0 && remainingBudget < 200) {
    spendingLevel = "Your budget is tight. Prioritize essential
expenses.";
} else {
    // Handle negative budget (overspending)
    console.log("Warning: You have exceeded your budget. Review
your expenses.");
}

// Display spending level
console.log(spendingLevel);
```

7 Travel Expense Optimizer

Objective:

- Create a JavaScript program that helps users plan and optimize their travel expenses for a trip within India. The program will consider factors such as

mode of transportation, accommodation, and meal preferences. Utilize let, const, operators, and if-else statements.

Prerequisites:

- Intermediate understanding of JavaScript syntax.
- Familiarity with prompt function for user input.
- Knowledge of let and const for variable declaration.
- Understanding of comparison operators (e.g., `>=`, `<=`, `==`).
- Ability to use complex if-else statements for decision-making.

Step-by-Step Solution:

Step 1: Welcome Message

```
// Welcome message
console.log("Travel Expense Optimizer");
```

Step 2: Get User Input for Mode of Transportation

```
// Get user input for mode of transportation
const transportationMode = prompt("Select your mode of
transportation (e.g., train, bus, or car)").toLowerCase();
```

Step 3: Get User Input for Accommodation

```
// Get user input for accommodation preference
const accommodationPreference = prompt("Choose your
accommodation preference (e.g., budget, mid-range, or
luxury)").toLowerCase();
```

Step 4: Get User Input for Meal Preferences

```
// Get user input for meal preferences
const mealPreference = prompt("Specify your meal preference
(e.g., vegetarian, non-vegetarian)").toLowerCase();
```

Step 5: Calculate and Optimize Expenses

```
// Declare variables to store expense details
let transportationExpense;
let accommodationExpense;
let mealExpense;
let totalExpense;

// Use if-else statements to calculate expenses based on user
preferences
if (transportationMode === 'train') {
  transportationExpense = 1500;
} else if (transportationMode === 'bus') {
  transportationExpense = 1200;
} else if (transportationMode === 'car') {
  transportationExpense = 2000;
} else {
  // Handle invalid transportation mode input
  console.log("Invalid transportation mode input. Please enter
'train', 'bus', or 'car'.");
  process.exit() // Terminate the script
}

if (accommodationPreference === 'budget') {
  accommodationExpense = 1000;
} else if (accommodationPreference === 'mid-range') {
```

```

    accommodationExpense = 2000;
} else if (accommodationPreference === 'luxury') {
    accommodationExpense = 4000;
} else {
    // Handle invalid accommodation preference input
    console.log("Invalid accommodation preference input. Please
enter 'budget', 'mid-range', or 'luxury'.");
    process.exit(); // Terminate the script
}

if (mealPreference === 'vegetarian') {
    mealExpense = 500;
} else if (mealPreference === 'non-vegetarian') {
    mealExpense = 800;
} else {
    // Handle invalid meal preference input
    console.log("Invalid meal preference input. Please enter
'vegetarian' or 'non-vegetarian'.");
    process.exit(); // Terminate the script
}

// Calculate total expense
totalExpense = transportationExpense + accommodationExpense +
mealExpense;

```

Step 6: Display Optimized Expense Breakdown

```

// Display optimized expense breakdown
console.log(`Expense Breakdown:
- Transportation: INR ${transportationExpense}
- Accommodation: INR ${accommodationExpense}
- Meals: INR ${mealExpense}
-----
Total Expense: INR ${totalExpense}`);

```

Step 7: Final code

```
// Welcome message
console.log("Travel Expense Optimizer");

// Get user input for mode of transportation
const transportationMode = prompt("Select your mode of
transportation (e.g., train, bus, or car)").toLowerCase();

// Get user input for accommodation preference
const accommodationPreference = prompt("Choose your
accommodation preference (e.g., budget, mid-range, or
luxury)").toLowerCase();

// Get user input for meal preferences
const mealPreference = prompt("Specify your meal preference
(e.g., vegetarian, non-vegetarian)").toLowerCase();

// Declare variables to store expense details
let transportationExpense;
let accommodationExpense;
let mealExpense;
let totalExpense;

// Use if-else statements to calculate expenses based on user
preferences
if (transportationMode === 'train') {
  transportationExpense = 1500;
} else if (transportationMode === 'bus') {
  transportationExpense = 1200;
} else if (transportationMode === 'car') {
  transportationExpense = 2000;
} else {
  // Handle invalid transportation mode input
}
```

```
    console.log("Invalid transportation mode input. Please enter
'train', 'bus', or 'car'.");
    process.exit() // Terminate the script
}

if (accommodationPreference === 'budget') {
    accommodationExpense = 1000;
} else if (accommodationPreference === 'mid-range') {
    accommodationExpense = 2000;
} else if (accommodationPreference === 'luxury') {
    accommodationExpense = 4000;
} else {
    // Handle invalid accommodation preference input
    console.log("Invalid accommodation preference input. Please
enter 'budget', 'mid-range', or 'luxury'.");
    process.exit() // Terminate the script
}

if (mealPreference === 'vegetarian') {
    mealExpense = 500;
} else if (mealPreference === 'non-vegetarian') {
    mealExpense = 800;
} else {
    // Handle invalid meal preference input
    console.log("Invalid meal preference input. Please enter
'vegetarian' or 'non-vegetarian'.");
    process.exit() // Terminate the script
}

// Calculate total expense
totalExpense = transportationExpense + accommodationExpense +
mealExpense;

// Display optimized expense breakdown
console.log(`Expense Breakdown:
- Transportation: INR ${transportationExpense}`)
```

```
- Accommodation: INR ${accommodationExpense}  
- Meals: INR ${mealExpense}  
-----  
Total Expense: INR ${totalExpense}` );
```



PRIMATHON
ACADEMY