

PROGRAMMING ASSIGNMENT: EXPENSE TRACKER APPLICATION

Task: You are tasked with creating an Expense Tracker application that allows users to manage their expenses. The application should involve reading and writing from a file, obtaining input from the user, performing simple computations, displaying information based on user input and calculations, and formatting the output for readability.

Instructions:

- 1) Create a text file named "expenses.txt" to store the expense data. Each line in the file should contain an expense record in the format: "Category,Amount".
- 2) Write a program in C++ that accomplishes the following:
 - a) Reads the contents of the "expenses.txt" file.
 - b) Displays a menu to the user with the following options:
 - i) Add an expense
 - ii) View expenses by category
 - iii) Calculate total expenses
 - iv) Exit c.
 - c) Based on the user's choice, perform the appropriate actions:
 - i) If the user chooses to add an expense, prompt for the expense category and amount, and append the record to the "expenses.txt" file.
 - ii) If the user chooses to view expenses by category, prompt for a category name and display all expenses for that category.
 - iii) If the user chooses to calculate total expenses, compute and display the total expenses across all categories.
 - iv) If the user chooses to exit, terminate the program. d. Format the output for readability, such as aligning columns and adding appropriate headers.

Requirements:

1. Your program should handle any file-related errors appropriately (e.g., file not found, read/write errors).
2. Validate user input to ensure it is within the acceptable range or format.
3. Perform computations, such as calculating the total expenses or filtering expenses by category, as required by the user's choice.
4. Display the information in a formatted manner, such as using tables or a similar structure for better readability.
5. Maintain the state of the expense data across multiple programs run by reading from and writing to the "expenses.txt" file.

Example Output:

Menu:

1. Add an expense
2. View expenses by category
3. Calculate total expenses
4. Exit

Enter your choice: 1

Add an Expense

Expense Category: Grocery Expense Amount: \$50.00 Expense added successfully!

Menu:

1. Add an expense
2. View expenses by category
3. Calculate total expenses
4. Exit

Enter your choice: 2

View Expenses by Category

Enter the category name: Grocery

Expenses for Category: Grocery

Category	Amount
----------	--------

Grocery	\$50.00
---------	---------

Menu:

1. Add an expense
2. View expenses by category
3. Calculate total expenses
4. Exit

Enter your choice: 3

Total Expenses: \$50.00

Menu:

1. Add an expense
2. View expenses by category
3. Calculate total expenses
4. Exit

Enter your choice: 4

Exiting the Expense Tracker Application...

Submission: Submit your program's source code along with any necessary files, such as "expenses.txt" on Grade scope. Provide clear instructions on how to run the program, including any dependencies or setup required.

EXPENSE TRACKER APPLICATION IMPLEMENTATION GUIDELINE

Please check the starter code provided in the assignment and complete functions where you see **“ADD CODE HERE”** or **“UPDATE ME”** statement.

1. Implementing getMenu()

In the getMenu() function, you need to complete the code for reading the user's choice and updating the selectedChoice variable.

2. Implementing addExpense(string category, float expense)

In the addExpense() function, you need to complete the code for writing the expense to the expenses.txt file.

3. Implementing readExpenses()

In the readExpenses() function, you need to complete the code for reading all expenses from the expenses.txt file.

4. Implementing calculateExpenses()

In the calculateExpenses() function, you need to complete the code for calculating total expenses from the expenses.txt file.

5. Implementing sizeTextto10(string expenseCategory)

In the sizing category function, you need to complete the code for formatting the expense category string to a maximum length of 10 characters.

6. Implementing the sizeExpAmountTo9(float amount)

In the sizing Amount function, you need to complete the code for formatting the expense amount to a maximum length of 9 characters.

7. Implementing the printExpense() method (CategoryExpensePair expense)

In the printExpense() function, you need to complete the code for printing an expense.

8. Testing the Application

Once you have implemented all the missing functions, you can test the Expense Tracker application. Compile and run the code and interact with the application by selecting different menu options.

- Test adding expenses by entering valid and invalid expense amounts and categories.
- Test viewing expenses by category by entering valid and invalid category names.
- Test calculating total expenses and ensure the correct amount is displayed.
- Test the error handling for invalid user inputs.