Execute | Beautify | Share | Source Code | Help

```java
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import java.util.Scanner;

public class ExpenseTracker {
    private Map<String, List<Expense>> expensesMap;

    public ExpenseTracker() {
        expensesMap = new HashMap<>();
    }

    public void addExpense(String category, String description, double amount)
        {
        Expense expense = new Expense(description, amount, category);
        expensesMap.computeIfAbsent(category, k -> new ArrayList<>()).add
            (expense);
    }

    public void viewExpenses() {
        for (Map.Entry<String, List<Expense>> entry : expensesMap.entrySet())
            {
            String category = entry.getKey();
            List<Expense> expenses = entry.getValue();

            System.out.println("Category: " + category);
            for (Expense expense : expenses) {
                System.out.println("Description: " + expense.getDescription()
```

Execute | Beautify | Share Source Code Help

```
            for (Expense expense : expenses) {
                System.out.println("Description: " + expense.getDescription()
                    +
                        ", Amount: " + expense.getAmount());
            }
            System.out.println("--------");
        }
    }

    public double getTotalExpenses(String category) {
        return expensesMap.getOrDefault(category, new ArrayList<>())
                .stream()
                .mapToDouble(Expense::getAmount)
                .sum();
    }

    public static void main(String[] args) {
        ExpenseTracker expenseTracker = new ExpenseTracker();
        Scanner scanner = new Scanner(System.in);

        while (true) {
            System.out.println("1. Add Expense");
            System.out.println("2. View Expenses");
            System.out.println("3. View Total Expenses for a Category");
            System.out.println("4. Exit");

            int choice = scanner.nextInt();

            switch (choice) {
                case 1:
```

Execute | Beautify | Share Source Code Help

```
                switch (choice) {
                case 1:
                    System.out.println("Enter Category: ");
                    String category = scanner.next();
                    System.out.println("Enter Description: ");
                    String description = scanner.next();
                    System.out.println("Enter Amount: ");
                    double amount = scanner.nextDouble();

                    expenseTracker.addExpense(category, description, amount);
                    break;
                case 2:
                    expenseTracker.viewExpenses();
                    break;
                case 3:
                    System.out.println("Enter Category: ");
                    String viewCategory = scanner.next();
                    System.out.println("Total Expenses: " + expenseTracker
                        .getTotalExpenses(viewCategory));
                    break;
                case 4:
                    System.out.println("Exiting Expense Tracker. Goodbye!");
                    System.exit(0);
                    break;
                default:
                    System.out.println("Invalid choice. Please try again.");
                }
            }
        }
    }
```