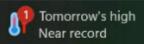
```
Execute | Beautify | Share | Source Code | (?) Help
   1 import java.util.ArrayList;
       import java.util.HashMap;
      import java.util.List;
      import java.util.Map;
   4
      import java.util.Scanner;
   6
      public class ExpenseTracker {
   8
           private Map<String, List<Expense>> expensesMap;
   9
  10 -
           public ExpenseTracker() {
               expensesMap = new HashMap<>();
  11
  12
           }
  13
           public void addExpense(String category, String description, double amount)
  14 -
               {
  15
               Expense expense = new Expense(description, amount, category);
               expensesMap.computeIfAbsent(category, k -> new ArrayList<>()).add
  16
                   (expense);
           }
  17
  18
           public void viewExpenses() {
  19 -
  20 -
               for (Map.Entry<String, List<Expense>>> entry : expensesMap.entrySet())
  21
                   String category = entry.getKey();
  22
                   List<Expense> expenses = entry.getValue();
  23
  24
                   System.out.println("Category: " + category);
  25
                   for (Expense expense : expenses) {
                       System.out.println("Description: " + expense.getDescription()
  26
```

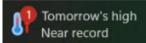








```
Source Code
                                            (?) Help
                      System.out.println("Description: " + expense.getDescription()
  26
                              ", Amount: " + expense.getAmount());
 27
  28
                 System.out.println("----");
  29
              }
  30
          }
 31
  32
         public double getTotalExpenses(String category) {
  33 -
              return expensesMap.getOrDefault(category, new ArrayList<>())
 34
 35
                      .stream()
  36
                      .mapToDouble(Expense::getAmount)
 37
                      .sum();
          }
 38
  39
         public static void main(String[] args) {
 40
              ExpenseTracker expenseTracker = new ExpenseTracker();
 41
 42
             Scanner scanner = new Scanner(System.in);
 43
 44
             while (true) {
                  System.out.println("1. Add Expense");
 45
                 System.out.println("2. View Expenses");
 46
                 System.out.println("3. View Total Expenses for a Category");
 47
                 System.out.println("4. Exit");
 48
 49
                  int choice = scanner.nextInt();
 50
 51
 52 -
                  switch (choice) {
                      case 1:
  53
```









```
Execute | 

Beautify | 

Share
                                            (?) Help
                               Source Code
53
                     case 1:
                         System.out.println("Enter Category: ");
54
55
                         String category = scanner.next();
                         System.out.println("Enter Description: ");
56
57
                         String description = scanner.next();
58
                         System.out.println("Enter Amount: ");
59
                         double amount = scanner.nextDouble();
60
61
                         expenseTracker.addExpense(category, description, amount);
62
                         break;
                    case 2:
63
64
                         expenseTracker.viewExpenses();
65
                         break;
66
                     case 3:
67
                         System.out.println("Enter Category: ");
                         String viewCategory = scanner.next();
68
                         System.out.println("Total Expenses: " + expenseTracker
69
                             .getTotalExpenses(viewCategory));
70
                         break;
71
                    case 4:
                         System.out.println("Exiting Expense Tracker. Goodbye!");
72
73
                         System.exit(0);
                         break;
74
                    default:
75
76
                         System.out.println("Invalid choice. Please try again.");
77
78
79
        }
80
```

