

```
1 import java.util.ArrayList;
2 import java.util.HashMap;
3 import java.util.List;
4 import java.util.Map;
5 import java.util.Scanner;
6
7 public class ExpenseTracker {
8     private Map<String, List<Expense>> expensesMap;
9
10    public ExpenseTracker() {
11        expensesMap = new HashMap<>();
12    }
13
14    public void addExpense(String category, String description, double amount)
15    {
16        Expense expense = new Expense(description, amount, category);
17        expensesMap.computeIfAbsent(category, k -> new ArrayList<>()).add
18            (expense);
19    }
20
21    public void viewExpenses() {
22        for (Map.Entry<String, List<Expense>> entry : expensesMap.entrySet())
23        {
24            String category = entry.getKey();
25            List<Expense> expenses = entry.getValue();
26
27            System.out.println("Category: " + category);
28            for (Expense expense : expenses) {
29                System.out.println("Description: " + expense.getDescription());
30            }
31        }
32    }
33}
```



Tomorrow's high
Near record



Search



PRE

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



Tomorrow's high
Near record



Search



```
52         switch (choice) {
53             case 1:
54                 System.out.println("Enter Category: ");
55                 String category = scanner.next();
56                 System.out.println("Enter Description: ");
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;
63             case 2:
64                 expenseTracker.viewExpenses();
65                 break;
66             case 3:
67                 System.out.println("Enter Category: ");
68                 String viewCategory = scanner.next();
69                 System.out.println("Total Expenses: " + expenseTracker
70                                     .getTotalExpenses(viewCategory));
71                 break;
72             case 4:
73                 System.out.println("Exiting Expense Tracker. Goodbye!");
74                 System.exit(0);
75                 break;
76             default:
77                 System.out.println("Invalid choice. Please try again.");
78         }
79     }
80 }
```

Execute | Beautify | Share | Source Code | Help

```
72         System.out.println("Exiting Expense Tracker. Goodbye!");
73         System.exit(0);
74         break;
75     default:
76         System.out.println("Invalid choice. Please try again.");
77     }
78 }
79 }
80 }
81
82 class Expense {
83     private String description;
84     private double amount;
85     private String category;
86
87     public Expense(String description, double amount, String category) {
88         this.description = description;
89         this.amount = amount;
90         this.category = category;
91     }
92
93     public String getDescription() {
94         return description;
95     }
96
97     public double getAmount() {
98         return amount;
99     }
100 }
```

Terminal

```
1. Add Expense
2. View Expenses
3. View Total Expenses for a Category
4. Exit
```

29°C
Mostly cloudy



Search

