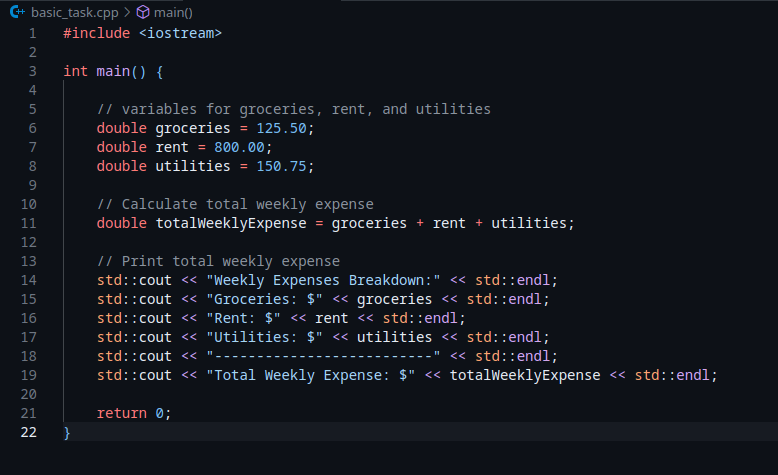
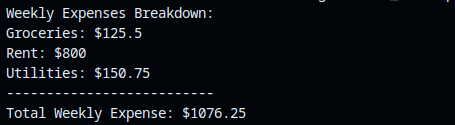
Assessment Task 1: Personal Budget Tracker

# Basic Expense Tracker:

This is a simple c++ code that have three (3) variables (Groceries, Rent and Utilities)

Each variable is assigned values after which the total weekly expense is calculated and the result is printed in the console.  
  
**Basic Expense Tracker code:**  


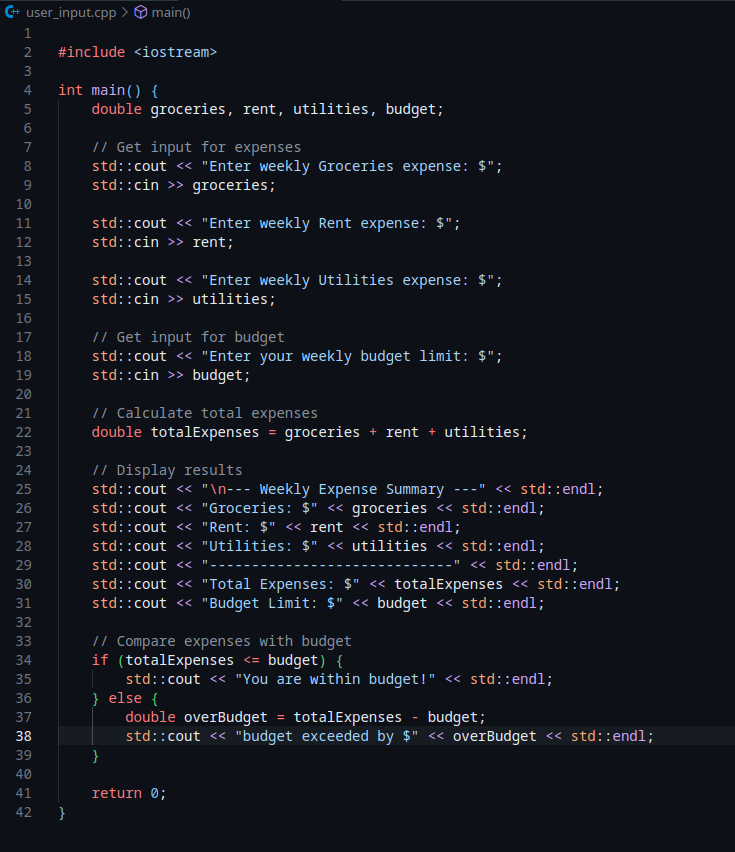
**Basic Expense Tracker example output:**



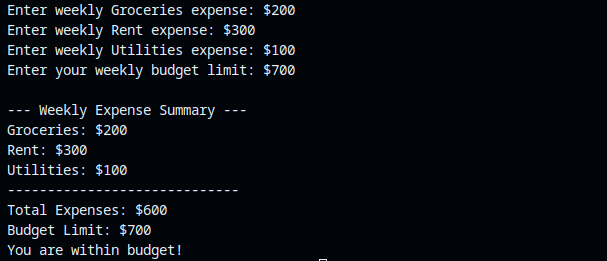
# User Input and Budget Check

This is an improvement of the Basic expense tracker. This expense tracker accepts user inputs for Groceries, Rent and Utilities and it also accepts the budget limit of the user as an input. After the user inputs the values for groceries,rent, utilities and budget, the tracker calculate the total expense against the user’s budget after which the expense tracker outputs in the console, two likely outcomes, whether the expense is above or within budget.

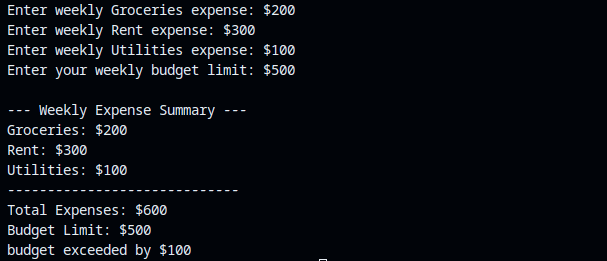
**Example code:**



**Example output within budget:**

****

**Example output above budget:**

****

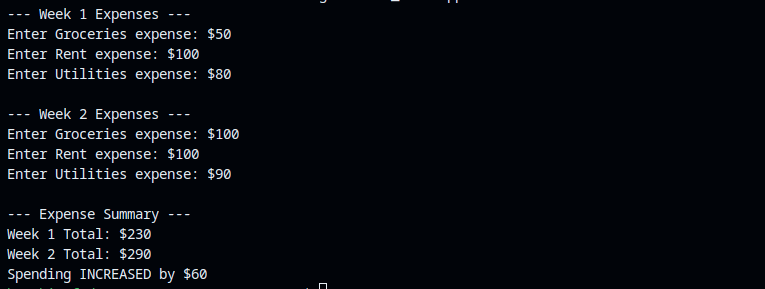
# Multi-Week Tracking

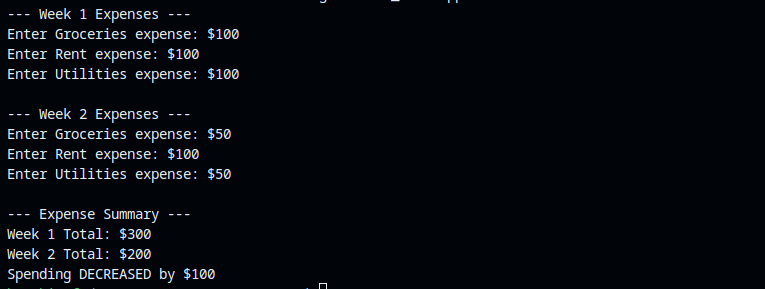
This expense tracking system allows the user to input expenses for two consecutive weeks. After the user have input the expenses, the expense is calculated on a weekly basis to determine whether the user spend more or less between first week and second week or spending remains the same across both weeks.

**Example code:**

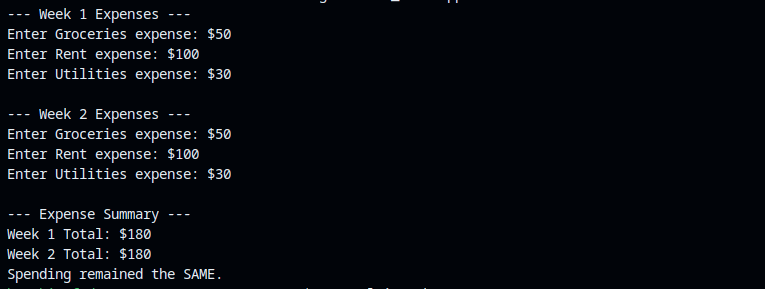
****

**Example output where the expense increases after two weeks:**

****

**Example output where the expense decreases after two weeks:  
**

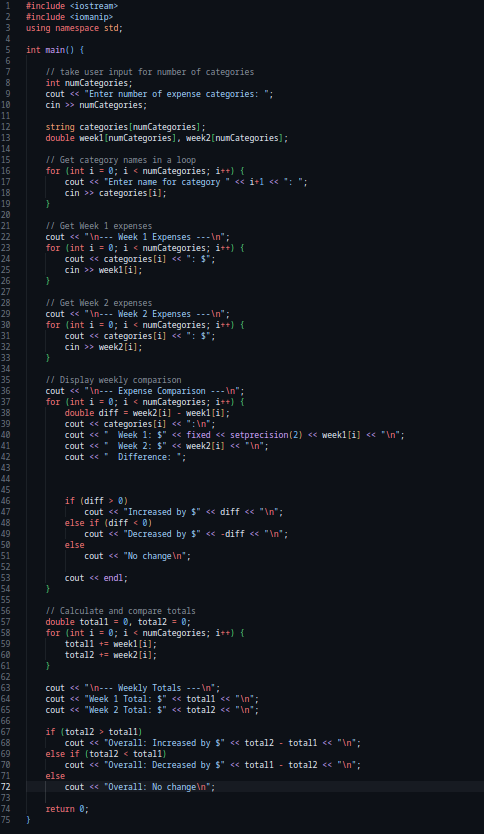
**Example output where the expense remains the same after two weeks:**

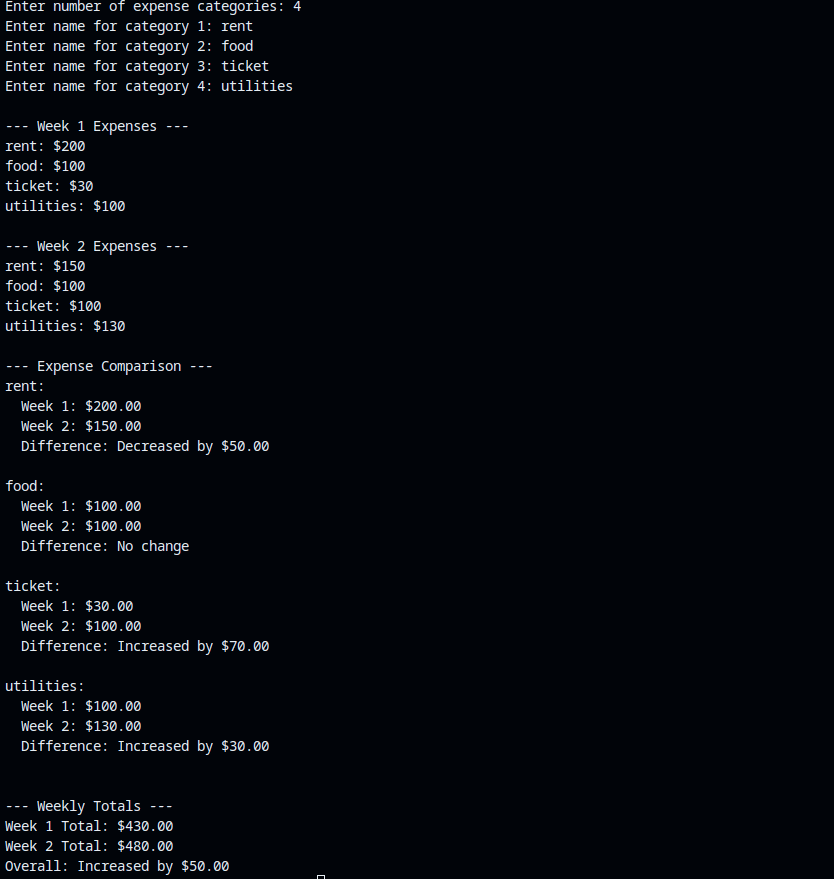
****

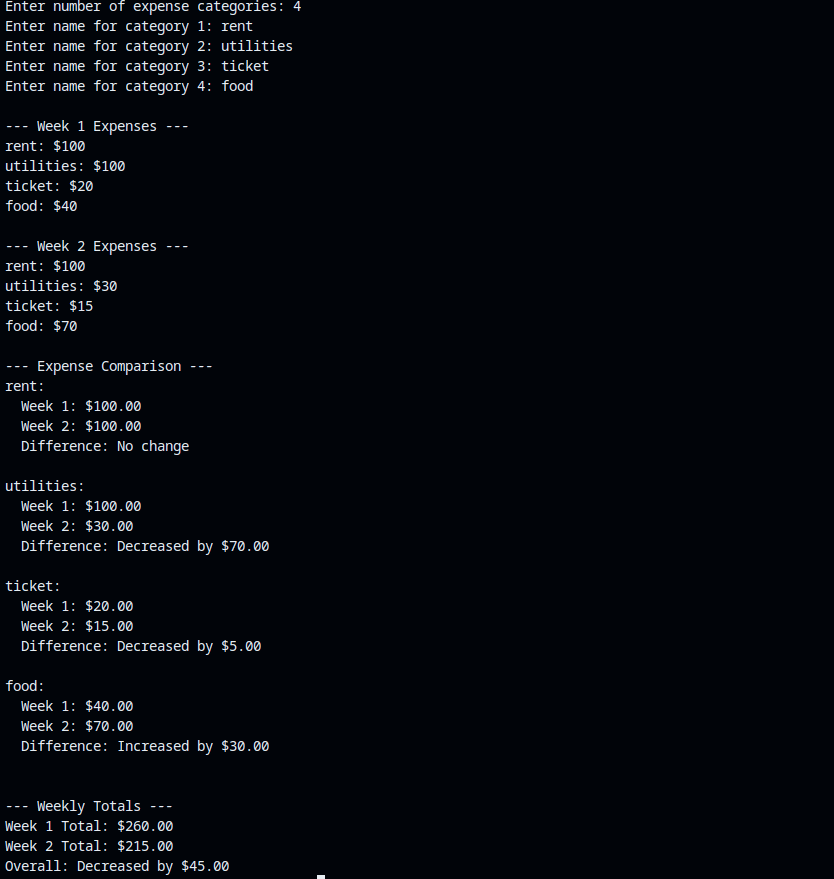
# Dynamic Input and Enhanced Comparison

This advanced expense tracker system allows the user to dynamically specify the number of categories by taking an input on the number of expense categories, the user then sets the value for each category with another input and the expense tracking system calculates the user’s expense over two weeks, with each week having its own separate calculations. After the calculation of each week’s expense, there is an output summary that determines whether the expense among categories and total expense increases, decreases or remains the same over the course of the two weeks.

**Example code:**



**Example output increase in category expense:**  


**Example output decrease in category expense:  
**

**Example output no in category expense:**

