PYTHON PROGRAMMING PROJECT 3

Project: Expense tracker

Creating an expense tracker using Python involves designing a program that allows users to input their expenses, categorize them, and then view or analyze their spending. I'll provide a simple example using Python and explaining each part of the program.

Program:

```
class ExpenseTracker:
    def _init_(self):
        self.expenses = {}

    def add_expense(self, category, amount):
        if category in self.expenses:
            self.expenses[category] += amount
        else:
            self.expenses[category] = amount

    def view_expenses(self):
        print("Expense Summary:")
        for category, amount in self.expenses.items():
            print(f"{category}: ${amount}")

def main():
    tracker = ExpenseTracker()

    while True:
        print("\nExpense Tracker Menu:")
        print("1. Add Expense")
        print("2. View Expenses")
        print("3. Exit")

        choice = input("Enter your choice (1/2/3): ")

        if choice == '1':
            category = input("Enter expense category: ")
            amount = float(input("Enter expense amount: "))
            tracker.add_expense(category, amount)
            print("Expense added successfully!")

        elif choice == '2':
            tracker.view_expenses()

        elif choice == '3':
            print("Exiting Expense Tracker. Goodbye!")
            break

        else:
            print("Invalid choice. Please enter 1, 2, or 3.")

if _name_ == "_main_":
    main()
```

Explanation:

1. ExpenseTracker Class: This class is responsible for managing expenses. It has methods for adding expenses (add_expense) and viewing expenses (view_expenses).

The expenses are stored in a dictionary, where the keys are expense categories and the values are the total amount spent in each category.

2. Main Function (main): This function initializes an instance of ExpenseTracker and then enters into a loop where the user can choose different options:

Option 1: Add Expense - The user enters a category and amount, and the expense is

Option 2: View Expenses - The user can view a summary of all expenses added to the

Tracker.

Option 3: Exit - The program exits the loop and terminates.

3. Usage of the Program: When you run the program, it will repeatedly prompt you to choose an option. You can add expenses, view the summary, and exit the program when you're done.

This is a basic example, and you can extend it by adding features like date tracking, persisting data to a file, or incorporating graphical interfaces for better user interaction.