```
class ExpenseTracker:
    def _init_(self):
        """
        Initialize the ExpenseTracker with an empty expenses dictionary.
        """
        self.expenses = {}

    def add_expense(self, category, amount):
        """
        Add an expense to the tracker.

        Parameters:
        - category (str): Expense category.
        - amount (float): Expense amount.
        """
        if category in self.expenses:
            self.expenses[category] += amount
        else:
            self.expenses[category] = amount

    def view_expenses(self):
        """
        View all expenses in the tracker.
        """
        print("Expense Tracker:")
        for category, amount in self.expenses.items():
            print(f"{category}: ${amount}")

    def analyze_expenses(self):
        """
        Analyze and display basic statistics about expenses.
        """
        total_expenses = sum(self.expenses.values())
        num_categories = len(self.expenses)
        average_expense = total_expenses / num_categories if num_categories > 0 else 0

        print(f"Total Expenses: ${total_expenses}")
        print(f"Number of Categories: {num_categories}")
        print(f"Average Expense per Category: ${average_expense:.2f}")

# User interface and interaction
tracker = ExpenseTracker()

while True:
    print("\nExpense Tracker Menu:")
    print("1. Add Expense")
    print("2. View Expenses")
    print("3. Analyze Expenses")
    print("4. Exit")

    choice = input("Enter your choice (1-4): ")

    if choice == "1":
        category = input("Enter expense category: ")
        amount = float(input("Enter expense amount: "))
        tracker.add_expense(category, amount)

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

    elif choice == "3":
        tracker.analyze_expenses()
```

```
elif choice == "4":
    print("Exiting Expense Tracker. Goodbye!")
    break

else:
    print("Invalid choice. Please enter a number between 1 and 4.")
```