

Class 07 Assignment - April 21, 2025

✓ Assignment:

One Dollar (\$1) Challenge

Create a unique application by applying OOP principles. It could be anything from a CLI app to an API or anything in between.

If there are multiple qualified submissions, there will be two winners: one male and one female.

```
# personal_expense_tracker.py

import datetime

# Class to represent a single expense
class Expense:
    def __init__(self, amount: float, category: str, description: str):
        self.amount = amount
        self.category = category
        self.description = description
        self.date = datetime.date.today() # Automatically captures today's date

    def __str__(self):
        # Format the expense for clean printing
        return f"{self.date} - {self.category} - Rs. {self.amount} - {self.description}"

# Class to manage multiple expenses
class ExpenseManager:
    def __init__(self):
        self.expenses = [] # Stores all Expense objects

    def add_expense(self, amount, category, description):
        # Creates and stores a new expense
        expense = Expense(amount, category, description)
        self.expenses.append(expense)
        print("✓ Expense added successfully.")

    def show_all_expenses(self):
        # Displays all recorded expenses
        if not self.expenses:
            print("⊘ No expenses recorded yet.")
        else:
            print("\n📁 All Expenses:")
            for exp in self.expenses:
                print(exp)
```

```

def generate_monthly_report(self):
    # Generates a summary of total expenses by category
    report = {}
    for exp in self.expenses:
        report[exp.category] = report.get(exp.category, 0) + exp.amount

    print("\n📊 Monthly Expense Report:")
    for category, total in report.items():
        print(f"{category}: Rs. {total}")

# Entry point for the CLI app
def main():
    manager = ExpenseManager() # Create an instance of the manager

    while True:
        # Display menu options to the user
        print("\n--- Personal Expense Tracker ---")
        print("1. Add Expense")
        print("2. Show All Expenses")
        print("3. Monthly Report")
        print("4. Exit")
        choice = input("Choose an option (1-4): ")

        if choice == "1":
            try:
                # Gather user input for the new expense
                amount = float(input("Enter amount (Rs): "))
                category = input("Enter category (e.g., Food, Travel): ")
                description = input("Enter description: ")
                manager.add_expense(amount, category, description)
            except ValueError:
                # Handle invalid amount input
                print("❌ Invalid input. Amount must be a number.")
        elif choice == "2":
            # Show all expenses
            manager.show_all_expenses()
        elif choice == "3":
            # Generate and show the monthly report
            manager.generate_monthly_report()
        elif choice == "4":
            # Exit the app
            print("👋 Exiting... Thank you!")
            break
        else:
            # Handle invalid menu choices
            print("❌ Invalid choice. Try again.")

```

```
# Run the application
if __name__ == "__main__":
    main()
```