COMP20270

OOP in Python

# Assignment 1

# **Kitty**

Due Monday 23rd November (Wk10)

## 



## Objective

Traditionally, when a group of friends would go on holiday (😢), they’d manage a kitty to keep track of shared expenses. In recent years, a number of phone apps have emerged to digitise this process. Examples include, [Settle Up](https://settleup.io), [Splitwise](https://www.splitwise.com) and [Kittysplit](https://www.kittysplit.com/en/).

The objective for this assignment is to design and implement a basic Kitty application.

* The first part of the exercise is to decide what are the main objects in the application. This will be done as a group exercise.
* The second part of the exercise will be to implement a basic Kitty system that can process the scenarios described in the Appendix.

## Requirements

1. An individual can only participate in a single event at any given time.
2. Assume that all members share all transactions equally.
3. Set-up:
   1. Declare named individuals to participate in the event/activity.
   2. Setup the activity: indicating the participating individuals.
4. Code to add transactions. A transaction has (see Appendix):
   1. name: string
   2. amount: a positive number
   3. payee: member ID
5. Code to print the reconciliation, i.e. total cost, cost each, who owes who what? (see Appendix)
6. Submissions will be tested on the Samples in the Appendix and on similar test cases.

**Error Handling:**

1. Transactions should not be created for an event if the payee is not signed up for the event.
2. The code for inputting transactions should be robust against invalid data.

**How to proceed**: 

1. Start by creating basic versions of your classes.
2. Create the methods, etc. to allow you to record the transactions for Sample 1.
3. Write the code to track the total cost of the trip.
4. Tackle the code for performing the reconciliation for Sample 1:
   1. As a first step, calculate the balance (+,-) for each person.
   2. Then figure out who pays who. Write code for the scenario is Sample 1 first.
5. Extend the reconciliation code to handle Sample 2.

**Submission:** This is an individual (not group) project. Submission is through the Brightspace page. Your submission should comprise your notebook only. Clear all outputs in the notebook before saving for submission. You can use markdown cells in the notebook to explain any design decisions you have made.

## 

## 

## 

**A Note on Optimality:**

You might be tempted to try and find optimal reconciliations, i.e. minimising the number of transactions. Don’t give in to this temptation -- it’s a really hard problem, NP-Hard.

See: <https://www.win.tue.nl/~wstomv/publications/INFE023.pdf>

## 

## Appendix: Sample runs

**Sample 1**

* Annie, Sally & Bill are going to a concert.
* Annie paid for the tickets - €180.
* Sally paid for dinner - €75.
* Bill paid for drinks after - €19.
* Bill paid for the taxi - €16.
* Reconcile, who owes who what?

**Output**

Total: €290.00, that is €96.67 each.

Annie has balance €83.33

Bill has balance €-61.67

Sally has balance €-21.67

Bill pays Annie €61.67

Sally pays Annie €21.67

**Sample 2**

* Cathy, Robin & Jen are going to the cinema.
* Cathy paid for the tickets - €33.
* Robin paid for dinner - 60.
* Jen paid for drinks after - €21.
* Jen paid for the taxi - €27.
* Reconcile, who owes who what?

**Output**

Total: €141.00, that is €47.00 each.

Cathy has balance €-14.00

Robin has balance €13.00

Jen has balance €1.00

Cathy pays Robin €13.00

Cathy pays Jen €1.00

**Sample 3**

* Nora, Eva, Frankie & Harry go away for the weekend.
* Nora pays for dinner on Fri. €110.
* Eva pays for lunch on Sat. €60.
* Frankie paid for dinner €125.
* Harry paid for lunch on Sun €70.
* Reconcile, who owes who what?

**Output**

Total: €365.00, that is €91.25 each.

Nora has balance €18.75

Eva has balance €-31.25

Frankie has balance €33.75

Harry has balance €-21.25

Eva pays Nora €18.75

Eva pays Frankie €12.50

Harry pays Frankie €21.25