2807/7001ICT Programming Principles (I), Trimester 3, 2019
Workshop 10

School of Information and Communication Technology
Griffith University

November 7, 2019

Module	4
When	Day 10
Goals	This workshop's exercise is a longer problem where you define a class.
Marks	5
Due	All problems are due by the *beginning* of the next workshop.

1 Preparation

Before your workshop class:

- Read all of this document.
- Review the lecture notes sections 1 to 25.
- Bring some paper (a print-out of this document is best) and writing implements.
- Bring a storage device, such as a portable hard drive and cable, or a USB drive.

2 Pre-workshop questions

There are no pre-workshop questions for this workshop.

3 Workshop activities

3.1 Class design

Case study: A Go Card account maintains a balance that may be spent on public transport. Users may request a statement that shows all transactions. The only transactions are to top up the account with some positive number of dollars, and to take a ride costing some dollars and cents.

The goal for this exercise is to develop a class for a Go Card Account. The class will be tested by a program that simulates transactions, like this:

```
Creating account. Input initial balance: 100
? r 3.50
? r 10.90
? b
Balance = $85.60
? t 20
? x gghhg
Bad command.
? t
Bad command.
? q
```

```
Statement:
event              amount ($)  balance ($)
Initial balance                      100.00
Ride                     3.50         96.50
Ride                    10.90         85.60
Top up                  20.00        105.60
Final balance                        105.60
```

Where:

- `r` *`number`* simulates a ride costing *`number`* dollars;
- `t` *`number`* simulates a top up of *`number`* dollars;
- `b` requests the currrent balance; and
- `q` ends input and prints a statement.

Bad inputs are to be reported and ignored.

Let us consider the design for a class that represents a Go Card account. To design a class, we consider what services the object(s) must provide (its methods), and what data needs to be stored in the object(s) to support those services.

Questions:

- What is a good name for a class that represents a Go Card account?
  - Be descriptive of the what the class represents. Don't include the word "class" in the name.
- What services should be provided?
  - A constructor (`__init__`) is required to set up the account with an initial balance.
  - It needs to record the amount each ride costs. A method that accepts the amount as a parameter is required.
  - It needs to record the amount for each top-up. A method that accepts the amount as a parameter is required.
  - It needs to be able to report the current balance at any time. A method that returns this is required.
  - A method is required print out a statement of all of the transactions.

  We can see from the output of the proposed program that the class needs to store the details of every transaction in order. It also needs to
- What data is required to be stored in the object to enable those services?
  - So that a method can return the current balance at any time, it would be useful have a field for the current balance.
  - So that the full statement can be printed, the object must store the amount of each transaction, in order. What data type can grow and keep multiple values in the order they are added?

3.2 Problem 1 (3 marks)

Problem: Implement the program descibed above, leaving out the printing of a full statement at the end.

3.3 Problem 2 (2 marks)

Problem: Implement the program descibed above, including the printing of a full statement at the end.

4 After the workshop

- You have created programs that might be useful to refer back to in future workshops. Make sure that you will have that work in the future. One copy is not enough for at IT professional. You should have at least 2 copies:

  1. on your Griffith network storage drive; and
  2. on your portable storage device.