Python – Interim Assessment 2

Overview

The passing grade for this exercise is 80%.

To receive a passing grade, you should ensure that your programs adhere to the following points:

- Meet the requirements outlined in the question in an optimal manner
- Adhere to proper naming convention rules
- Use comments to explain your rationale, where felt necessary
- Ensure that code is correctly indented and is professionally presented

To maintain the integrity of this assessment, we ask that you do not seek assistance on the course forums in respect of these exercises. Any such posts made by learners will be immediately removed.

On receipt of your submission, an eTutor will provide you with constructive feedback.

Please turn over to begin the assessment.

You have saved up an amount of money, €500. Being a kind person, you like to help your friends by giving them small loans.

Your friends have asked if they can borrow the following amounts from you: 60,20,100,80,40,300,200,100. These requests are stored in the provided text file called loan_requests.txt, which should be read into your application and stored in a list.

Loans will be granted on a first come, first served basis. Write a Python program, which processes these loan request amounts.

If there is enough money in the kitty, the full amount of the loan request should be granted.

If, however, the amount of the loan request exceeds the amount of money remaining, whatever amount of money is left in the kitty should be given out. For example, if the balance remaining is €200 and the loan request is for €300, the balance remaining of €200 should be given out.

Loan requests go unpaid if the kitty is empty.

Just to recap, the following steps should be carried out:

- Store the available cash in a variable called kitty
- Create an empty list called requests
- Read the data from the text file line-by-line, storing each line as a different element in the requests list
- Loop through all of the requests in the list. For each request, check the following:
    - If the payment can be made in full
        - Print to the console stating that the request was paid
        - Write an additional line to the end of the loan_requests.txt file
    - If the payment can be partially paid
        - Print to the console stating that a partial payment was made
        - Write an additional line to the end of the loan_requests.txt file
    - If the payment cannot be paid (kitty is empty)
        - Print to the console stating that the request went unpaid
        - Write an additional line to the end of the loan_requests.txt file

The output in the console should look like the following:

```
60 - Paid!
20 - Paid!
100 - Paid!
80 - Paid!
40 - Paid!
300 request cannot be processed in full (Insufficient funds available). Amount paid:  200
Request of 200 is UNPAID!
Request of 100 is UNPAID!
```

The contents of the loan_requests.txt file should look like the following:

```
60
20
100
80
40
300
200
100

Request of 60 paid in full.
Request of 20 paid in full.
Request of 100 paid in full.
Request of 80 paid in full.
Request of 40 paid in full.
Request of 300 could not be paid in full.Partial payment of 200 made.
Outstanding Request:200
Outstanding Request:100
```

End of Assessment