

The Brief

Automation Objectives

Write a business automation program to:

1. Extract and summarise data from the finance dashboard in the final round of business stimulation game in MAB module. Refer to *PFB - Instructions to download CSV files for Integrated Group Project pdf* to download the files.
2. The automation will perform the tasks from the following csv files:
 - a. **Profit & Loss csv** : The program will compute the **difference** in the **net profit** column if net profit on the current day is lower than the previous day.
 - b. **Cash-On-Hand csv**: The program will compute the **difference** in Cash-on-Hand if the current day is lower than the previous day.
 - c. **Overheads csv**: The program will find the **highest** overhead category.
 - d. Write the computed amount from **a to c** to a text file and name it as `summary_report.txt`.

Figure 1.0 included two scenarios to illustrate the automation objectives and the expected output in `summary_report.txt`.

Files and Project Directory

You should organise your program and csv files into the following folder structure:

```
Folder : project_group
| cash_on_hand.py
| overheads.py
| profit_loss.py
└─ Folder: csv_reports
   Cash on Hand.csv
   Overheads.csv
   Profits and Loss.csv
```

The Brief

Dedicate each python file to achieve specific tasks. For example, the `cash_on_hand.py` should only contain codes that compute the difference in Cash-on-Hand, while `overheads.py` should only contain codes that find the highest overhead category.

Organizing code this way makes the overall program more manageable, easier to maintain and debug errors.

Coding Skills

To complete the assignment successfully, you need to use only the programming topics learn from PFB, unless given the permission to do so.

The use of external modules not taught will severely affect the grade. External module refers to additional module installed with pip install command.

However, you may use any **built-in** functions or/and modules.

Standard Criteria

The project will be evaluated based on:

1. Program Correctness
2. Code Readability
3. Code Elegance/ Efficiency
4. Code Documentation
5. Assignment Specification

Bonus Marks

Bonus marks will be awarded based on the group's ability to:

1. Collaborate on coding
2. Modularized the python files

How to Collaborate?

1. As this is a group project, you are expected to collaborate with each other and each member is expected to contribute to the project. To collaborate coding better, you can make of use GitHub, a leading collaboration platform used by major tech companies and programmers worldwide.
2. A set of instructional slides on how to collaborate on GitHub using Visual Studio Code are available. (Refer to Collaborate with GitHub.pdf)
3. Each member should be assigned to work on a specific part of the program. For example, a team member can work on the `cash_on_hand.py`, while another member can work on the `profit_loss.py`.

The Brief

What is a modular program?

1. Modularization is the technique of splitting a large programming task into smaller, separate, and manageable subtasks.
2. To achieve modularization, you can further organized the code in each python file as a function.
3. A main python file (main.py) will import these functions, to coordinate and execute the functions.
4. In this way the overall program becomes even more manageable, easier to maintain and debug errors.
5. Refer to Figure 2.0 for an example of modularizing a complex program.

Figure 1.0 Automation Objectives

SCENARIO 1

1. Salary Expense is the highest overheads in "Overheads.csv"
2. Each value on the current day is higher than the previous day in "Cash on Hand.csv" and "Profit & Loss.csv"

Overheads.csv

Category	Overheads
Salary Expense	28.77
Interest Expense	0.23
Rental Expense	20.64
Penalty Expense	12.88
Depreciation Expense	20.83
Human Resource Expense	16.66

Cash on Hand.csv

Day	Cash On Hand
35	6823839
36	6956180
37	7683145
38	8212180
39	8379000
40	8401232

Profit & Loss.csv

Day	Sales	Trading Profit	Operating Expense	Net Profit
35	24303924	8866263	2605390	6260273
36	24471890	8953446	2661675	6291771
37	25233785	9345165	2716605	6628560
38	25797345	9635457	2771130	6864327
39	26020982	9748900	2825655	6874707
40	26034115	9755787	2881080	6923245

Output: Summary report based on the scenario 1

```

summary_report - Notepad
File Edit Format View Help
[HIGHEST OVERHEADS] SALARY EXPENSE: 28.77%
[CASH SURPLUS] CASH ON EACH DAY IS HIGHER THAN THE PREVIOUS DAY
[NET PROFIT SURPLUS] NET PROFIT ON EACH DAY IS HIGHER THAN THE PREVIOUS DAY
Ln 4, Col 1 100% Windows (CRLF) UTF-8

```

SCENARIO 2

1. Depreciation Expense is the highest overheads in "Overheads.csv"
2. Value on day 36 is lower than day 35 and day 40 is lower than day 39 in "Cash on Hand.csv"
3. Value on day 38 is lower than day 37 in "Profit & Loss.csv"

Overheads.csv

Category	Overheads
Salary Expense	28.77
Interest Expense	0.23
Rental Expense	20.64
Penalty Expense	12.88
Depreciation Expense	40.83
Human Resource Expense	16.66

Cash on Hand.csv

Day	Cash On Hand
35	6956180
36	6823839
37	7683145
38	8212180
39	8401232
40	8379000

Profit & Loss.csv

Day	Sales	Trading Profit	Operating Expense	Net Profit
35	24303924	8866263	2605390	6260273
36	24471890	8953446	2661675	6291771
37	25233785	9345165	2716605	6628560
38	25797345	9635457	2771130	6628560
39	26020982	9748900	2825655	6874707
40	26034115	9755787	2881080	6923245

Output: Summary report based on scenario 2

```

summary_report - Notepad
File Edit Format View Help
[HIGHEST OVERHEADS] DEPRECIATION EXPENSE: 40.83%
[CASH DEFICIT] DAY: 36.6, AMOUNT: USD132281
[CASH DEFICIT] DAY: 40.8, AMOUNT: USD22292
[PROFIT DEFICIT] DAY: 38.0, AMOUNT: USD235767
Ln 5, Col 1 100% Windows (CRLF) UTF-8

```

Figure 2.0 Modularizing the program

