FinTech Unit 2 Homework PyBank: Grading Rubric

Criteria	Ratings				Points	Feedback
Set up Project • GitHub repo created - named: python-homework. • Directory created with appropriate activity. • *main.ipynb* added to main folder. • Repo updated with latest changes.	**35 Points Mastery** • Completed 4 out of 4 requirements • Code Runs without error and produces the assigned results • Code is creatively engineered where possible • Code uses concise logic	**34 > 28 Points Approaching Mastery** • Completed 3 out of 4 requirements • Code runs without error and produces the assigned results • Code accounts for all possible scenerio • Code is free of bugs	**28 > 23 Points Progressing** • Completed 2 out of 4 requirements • Code runs without error • Code produces results as expected 80% of the time	**23 > 0 Emerging** • Completed 1 or fewer out of the 4 requirements • No submission • Code runs with error	0	
Analysis and Calculations • Total number of months in dataset included. • Net total amount of Profit/Losses calculated over entire period. • Average of the changes in Profit/Losses calculated over entire period. • Greatest increase in Profits calculated over entire period (Date and Amount). • Greatest decrease in Profits calculated over entire period (Date and Amount). • Analysis printed and text file exported with final results.	**35 Points Mastery** • Completed 6 out of 6 requirements • Code Runs without error and produces the assigned results • Code is creatively engineered where possible • Code uses concise logic	**34 > 28 Points Approaching Mastery** • Completed 5 out of the 6 requirements • Code runs without error and produces the assigned results • Code accounts for all possible scenerio • Code is free of bugs	**28 > 23 Points Progressing** • Completed 4 out of the 6 requirements • Code runs without error • Code produces results as expected 80% of the time	**23 > 0 Emerging** • Completed 3 or fewer out of the 6 requirements • No submission • Code runs with error	0	
Coding Conventions/Formating • Appropriate header, name, short description at top of the notebook • Imports are at the top of the file, just after any headers or subheads. • Files read in from relative file path • Functions and variable names are descriptive, lowercase, with words separated by underscores • Clean code, no repetition, maintainable and highly reusable code. • Appropriate code wrapping and cell sizes • Appropriate subheads as needed	**10 Points Mastery** • Imports are at the top of the file, just after any module comments and docstrings, and before module globals and constants. • Function names are lowercase, with words separated by underscores • Variable names follow the same convention as function names. • Code follows (DRY) principals, no repetition, maintainable and highly reusable code.	**9 Points Approaching Mastery** • Variable names are specific and descriptive of the information held by the variable • Imports are within the top of file	**8 Points Progressing** • Code lacks proper indentation and length convention - Limit all lines to a maximum of 79 characters. • Variable names are generic and not descriptive of the information held by the variable • Imports and files are located in a non-standard location	**8 > 0 Emerging** • Code is excessively lengthy • Variable names are missing or lacking any descriptive information • Import and files are not loaded	0	
Deployment/Submission • Files submitted in personal repo • Appropriate directory structure with correct files needed to run scripts • Appropriate commit messages • Appropriate README	**10 Points Mastery** • Repository cloned to local machine • Files added to the repo via the command line • Appropriate commit messages	**9 Points Approaching Mastery** • Repository cloned to local machine • Files added to repo via the command line	**8 Points Progressing** • Repository created on GitHub • Files added manually on GitHub	**8 > 0 Emerging** • No Submission • Submission via incorrect format	0	
Documentation/Comments • Code is well commented with concise, relevant comments	**10 Points Mastery** • Code is well commented with concise, relevant notes	**9 Points Approaching Mastery** • Code is commented and mostly understandable to an outside user	**8 Points Progressing** • Code has comments, but they are not understandable to an outside user	**8 > 0 Emerging** • Code is not commented	0	

LETTER GRADE

TOTAL POINTS 0   **F**