FinTech Unit 19 Blockchain-Python Homework Grading Rubric

Criteria	Ratings			
Project Setup • Project Directory created. • Constants setup in separate file. • Mnemonic Generated. • Wallet keys derived.	**35 Points Mastery** • Completed 4 out of 4 requirements • Code runs without error and produces the assigned results • Code accounts for all possible scenario • Code is free of bugs	**34 > 28 Points Approaching Mastery** • Completed 3 out of 4 of requirements • Code runs without error • Code produces results as expected 80% of the time	**28 > 23 Points Progressing** • Completed fewer than 2 out of 4 requirements • Code runs without error • Code produces results, but not necessarily the correct results	**23 > 0 Emerging** • Completed 1 or none out of the 4 requirements • No submission • Code runs with error
Transactions • Transactions linked with signing libraries • Wallet funded and transactions sent. • **BTCTEST** address funded and screenshots of the transactions added. • **Local PoA Ethereum** transactions added and screenshots added.	**35 Points Mastery** • Completed 4 out of 4 requirements • Code runs without error and produces the assigned results • Code accounts for all possible scenario • Code is free of bugs	**34 > 28 Points Approaching Mastery** • Completed 3 out of 4 of requirements • Code runs without error • Code produces results as expected 80% of the time	**28 > 23 Points Progressing** • Completed 2 out of 4 requirements • Code runs without error • Code produces results, but not necessarily the correct results	**23 > 0 Emerging** • Completed 1 or none of the 4 requirements • No submission • Code runs with error
Optional Challenge • Support for **BTC** added. • Support for **LTC** using **lit** added. • **Function** added to track transaction status by **txid.**	**30 Points Mastery** • Completed 3 out of 3 requirements • Code runs without error and produces the assigned results • Code accounts for all possible scenario • Code is free of bugs	**20 Points Approaching Mastery** • Completed 2 out of 3 of requirements • Code runs without error • Code produces results as expected 80% of the time	**10 Points Progressing** • Completed 1 out of 3 requirements • Code runs without error • Code produces results, but not necessarily the correct results	**0** • Completed none of the 3 requirements • No submission • Code runs with error

Coding Conventions/Formatting • 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**	**9 Points Approaching Mastery**	**8 Points Progressing**	**8 > 0 Emerging**
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**	**9 Points Approaching Mastery**	**8 Points Progressing**	**8 > 0 Emerging**
Documentation/Comments • Code is well commented with concise, relevant comments	**10 Points Mastery**	**9 Points Approaching Mastery**	**8 Points Progressing**	**8 > 0 Emerging**