FinTech Unit 20 Solidity Homework Grading Rubric

All Levels

Coding Conventions/Formatting	10 Points Mastery	9 Points Approaching Mastery	8 Points Progressing	8 > 0 Emerging
• 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				
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

Level 1

Criteria	Ratings			
Contract Setup • *public* variables defined per assigned criteria. • Constructor function created accepting *address payable _one.* • Constructor function created accepting *address payable _two.* • Constructor function created accepting *address payable _three.*	**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
Contract Functionality • *balance* function created with assigned criteria. • *deposit* function created with assigned criteria. • *fallback* function created with assigned criteria. • *Contract tested*; *screenshots* provided.	**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

Level 2

Criteria	Ratings			
Contract Setup • Number of points calculated by dividing *msg.value.* • *uint amount* set to points for each employee. • *amount* added to *total* used to calculate the msg.value distribution*.* • *amount* transfer to each *employee*, *amount* set to the *points* and multiplied by given *percentage.*	**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
Contract Functionality • *remainder* sent employee with highest percentage. • *Contract Deployed.* • *Contract Tested, screenshots* provided.	**35 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	**34 > 28 Points Approaching Mastery** • Completed 2 out of 3 of requirements • Code runs without error • Code produces results as expected 80% of the time	**28 > 23 Points Progressing** • Completed 1 out of 3 requirements • Code runs without error • Code produces results, but not necessarily the correct results	**23 > 0 Emerging** • Completed fewer than 1 or none of the 3 requirements • No submission • Code runs with error

Level 3

Criteria	Ratings			
Contract Setup • *Human Resources* set as the constructor *msg.sender.* • *employee initiation variables* created. • *uint start_time* and *uint public distributed_shares* defined to assigned criteria. • *distribute function* created with *defined criteria.*	**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
Contract Functionality • *Contract deployed locally, screenshots provided.* • *Contract Tested* with *fakenow* testing logic*.* • *Contract deployed to a live Testnet, screenshots* provided.	**35 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	**34 > 28 Points Approaching Mastery** • Completed 2 out of 3 of requirements • Code runs without error • Code produces results as expected 80% of the time	**28 > 23 Points Progressing** • Completed 1 out of 3 requirements • Code runs without error • Code produces results, but not necessarily the correct results	**23 > 0 Emerging** • Completed fewer than 1 or none of the 3 requirements • No submission • Code runs with error