CSC 4980 / 6980 Blockchain & Applications

Assignment 3

Due Date: 11:59 pm, March 13th, 2019

In this assignment the student is expect to code a Solidity contract for a simple capped crowd sale token:

Follow:

https://medium.com/crowdbotics/how-to-build-a-simple-capped-crowdsale-token-using-openzeppelin-library-part-1-2789ec642308
and
https://medium.com/crowdbotics/how-to-build-a-simple-capped-crowdsale-token-using-openzeppelin-library-part-2-cf96cb66c3d0

After you have successfully completed the steps and tried the contract with the transactions on (https://medium.com/crowdbotics/how-to-build-a-simple-capped-crowdsale-token-using-openzeppelin-library-part-2-cf96cb66c3d0) you are required to make the following changes to the contract:

1) Change the minimum contribution to 5 Ether. (20 points)
2) Add method, getTokensLeft, to report how many tokens are left. (30 points)
3) **Graduate Students question:** Add the needed functionality to not allow more than 1 purchase per account. (40 points)

After you have successfully implemented, deployed and tested. You are to evaluate your contract with the following parameters:

Token name: "CSC4980 Token" - "CSC6890 Token" for graduate students
Token symbol: "GSU"
Token decimals: 18
rate: 450 tokens x Ether
cap: 150 Ether

Now perform the following transactions:
1) Try to buy tokens with 2.5 ether (10 points)
2) Buy tokens with 15 ether (10 points)
3) Return how many tokens are left (10 points)
Graduate Students 4): Buy tokens (again) with 25 Ether (10 points)

Note: You need to show ALL the steps you take to define the given parameters and perform the given operations. Each one should be included in your README.MD file with its respective answer.

Your program should include a README.md file with detailed instructions on how to run the program **and the solution to all the exercises in this assignment, plus the testing transactions in the tutorial**. This file should be properly formatted using GitHub Markdown language.

Your program will be graded for correctness and completeness. If the program does not work, you will receive a 0 on the assignment. The program itself is worth 50 points. Answers to the questions will NOT count if your program cannot produce the solutions you are claiming, so be sure it works.

What to turn in:

Each student is responsible to turn in their own code. Plagiarism will be heavily prosecuted, so please do not risk it. No two people code alike and this should be reflected in the programs.

1) You should upload all running code and the README.md file to your own GitHub repository. Yes, only GitHub, no other services will be accepted.
2) The README.md file should contain instructions on how to run your code and the solution to the provided exercises. If you wrote the improved program, also add instructions on how to run that program and the solution to the exercises using that program as well.
3) The README.md file should be properly created using GitHub markdown (https://guides.github.com/features/mastering-markdown/) it should have the student's name clearly displayed somewhere.

Upload to iCollege the downloaded ZIP file of your GitHub repository. There is a clone/download button on the main page. You should also submit the URL of your GitHub repository in the submission comments.

IMPORTANT: Failure to use GitHub and provide the indicated README.md file with instructions and answers will lead to a zero grade in the assignment.