Homework 5

- Honor Code: You must work completely independently on this assignment. Do not discuss the questions or answers with each other before the assignment is due. Any breach of the honor code will be handled per the University's policy on academic honesty.
- Follow the instructions very carefully. Answers that do not conform to the instructions will not be given credit.
- Submit your solutions through Blackboard as individual separate files, and not as zip archive.
- Understand thoroughly all the code given to you in this lab. Search for documentation online if there is a primitive or API you have not encountered before.
- Use Java 8.
- Only use the external Java libraries provided to you in the lab, if any.
- The term "mainnet" refers to the actual Bitcoin blockchain network. The term "testnet" is an alternative to mainnet to be used for testing. Testnet coins are separate and distinct from actual bitcoins, and are never supposed to have any value. This allows application developers or bitcoin testers to experiment, without having to use real bitcoins or worrying about breaking the main bitcoin chain. You can obtain testnet coins for free from the faucet at https://testnet.coinfaucet.eu/en/. It is courteous to send the testnet coins back to the faucet after you are done experimenting with them.

1. Read:
    a. Bitcoin addresses:
        i. https://en.bitcoin.it/wiki/Address
        ii. https://en.bitcoin.it/wiki/Technical_background_of_version_1_Bitcoin_addresses
        iii. https://en.bitcoin.it/wiki/List_of_address_prefixes
    b. Base58 encoding: https://en.bitcoin.it/wiki/Base58Check_encoding
    c. Bitcoin address reuse and privacy: https://en.bitcoin.it/wiki/Address_reuse
    d. Testnet: https://en.bitcoin.it/wiki/Testnet
    e. BitcoinJ
        i. https://bitcoinj.github.io/#getting-started
        ii. https://bitcoinj.github.io/working-with-transactions
        iii. https://bitcoinj.github.io/working-with-the-wallet
2. Set the hw5.WalletInit.WALLETS_DIR constant to a directory that stores all the wallets.
3. Read and thoroughly understand hw5.WalletInit.
4. Run the hw5.WalletInitTest class to create a wallet on testnet. This class will download and sync the testnet block chain on to your computer and thus may take a few minutes. Skim through the logging output to get a basic understanding of what's happening. Submit the current receive address of your wallet as your solution to question 4 in a text file called homework5solutions.txt.
5. Use the testnet faucet to send testnet coins to the testnet wallet's receive address. Look up the transaction ID at the testnet blockchain explorer at https://www.blocktrail.com/tBTC to see if the transaction was submitted to the network. Be sure to receive at least one confirmation in case a double-spend attack prevented you from receiving the coins. The faucet may send the same coin to you in multiple different transactions, so the blockchain explorer may tell you that the coin has been double-spent. To find the actual transaction that successfully sent you coins, run the WalletInitTest and watch the WalletInitTest.monitor method's periodic reporting of the transactions it receives from its peers. The output will tell you which transaction succeeded and which is dead. Submit the transaction ID and the amount of testnet coins sent to your receive address as your solution to question 5 in homework5solutions.txt. Answer the following questions: how many outputs does the transaction have? What is the purpose of the output that locks coins to the address that's not yours?
6. Run the hw5.WalletInitTest class again to update your wallet's blockchain and confirm that it contains the amount given to you in question 3.

7. Use BitcoinJ to write a custom Bitcoin address generator for mainnet. Place your implementation in the method CustomAddressGenerator.get(String). This method has one parameter which is a string called `prefix` encoded in base58. The method returns a bitcoin address on mainnet that begins with 1 followed by `prefix`. Submit your CustomAddressGenerator class.
8. Use the CustomAddressGenerator.get(String) method to generate a mainnet address that begins with 1 followed by the first four letters of your last name in uppercase. If your last name is shorter than four letters, append as many "X" characters as necessary. If the first four letters of your last name in uppercase includes a letter not allowed in a Bitcoin address, then use "X" instead of the letter. Submit your custom bitcoin address as your solution to question 8 in a text file called homework5solutions.txt.