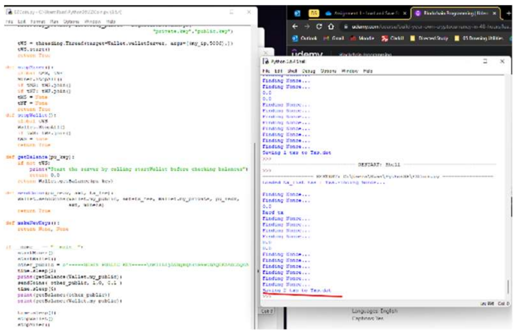
Recalibrating Wallet

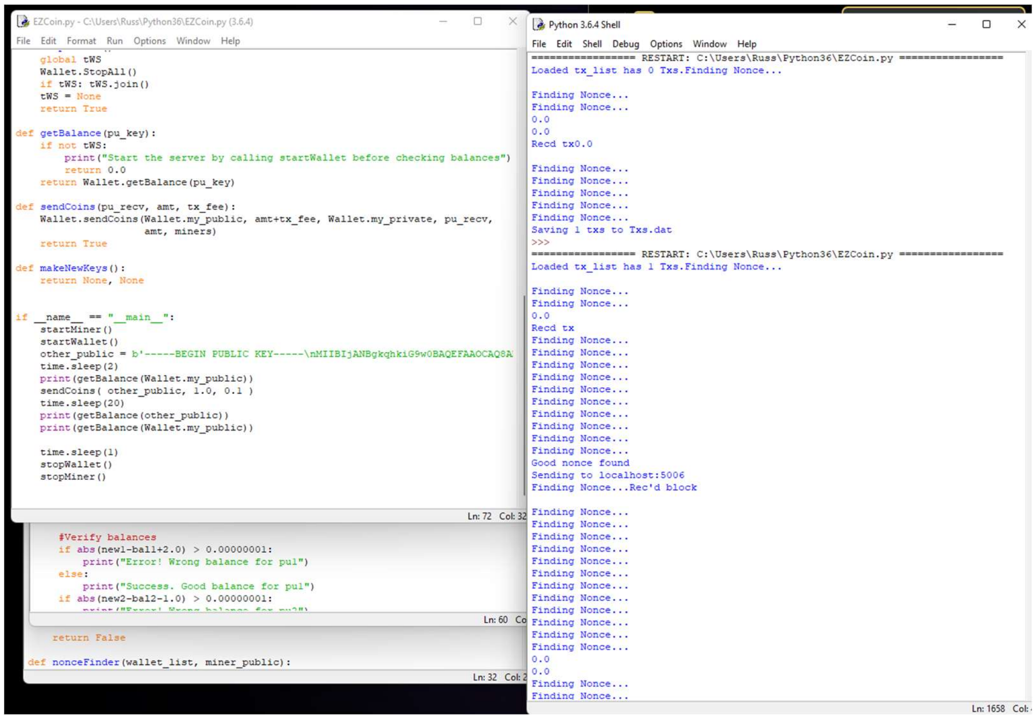
Vid 69

28:34 →



```
                                        "private.key","public.key")

    tWS = threading.Thread(target=Wallet.walletServer, args=((my_ip,5006),))
    tWS.start()
    return True

def stopMiner():
    [illegible]
    Miner.StopAll()
    if tMS: tMS.join()
    if tNF: tNF.join()
    tMS = None
    tNF = None
    return True
def stopWallet():
    global tWS
    Wallet.StopAll()
    if tWS: tWS.join()
    tWS = None
    return True

def getBalance(pu_key):
    if not tWS:
        print("Start the server by calling startWallet before checking balances")
        return 0.0
    return Wallet.getBalance(pu_key)

def sendCoins(pu_recv, amt, tx_fee):
    Wallet.sendCoins(Wallet.my_public, amt+tx_fee, Wallet.my_private, pu_recv,
                     amt, miners)
    return True

def makeNewKeys():
    return None, None


if __name__ == "__main__":
    startMiner()
    startWallet()
    other_public = b'-----BEGIN PUBLIC KEY-----\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8A
    time.sleep(2)
    print(getBalance(Wallet.my_public))
    sendCoins( other_public, 1.0, 0.1 )
    time.sleep(4)
    print(getBalance(other_public))
    print(getBalance(Wallet.my_public))

    time.sleep(1)
    stopWallet()
    stopMiner()
```

```
Finding Nonce...
Finding Nonce...
0.0
0.0
Finding Nonce...
Finding Nonce...
Finding Nonce...
Finding Nonce...
Finding Nonce...
Finding Nonce...
Finding Nonce...
Finding Nonce...
Saving 1 txs to Txs.dat
>>>
======================== RESTART: Shell ========================
>>>
================ RESTART: [illegible]\EZCoin.py ================
Loaded tx_list has 1 Txs.Finding Nonce...

Finding Nonce...
Finding Nonce...
0.0
Recd tx
Finding Nonce...
Finding Nonce...
Finding Nonce...
Finding Nonce...
0.0
0.0
Finding Nonce...
Finding Nonce...
Finding Nonce...
Finding Nonce...
Finding Nonce...
Finding Nonce...
Saving 1 txs to Txs.dat
>>>
```



EZCoin.py - C:\Users\Russ\Python36\EZCoin.py (3.6.4)

```
    global tWS
    Wallet.StopAll()
    if tWS: tWS.join()
    tWS = None
    return True

def getBalance(pu_key):
    if not tWS:
        print("Start the server by calling startWallet before checking balances")
        return 0.0
    return Wallet.getBalance(pu_key)

def sendCoins(pu_recv, amt, tx_fee):
    Wallet.sendCoins(Wallet.my_public, amt+tx_fee, Wallet.my_private, pu_recv,
                     amt, miners)
    return True

def makeNewKeys():
    return None, None


if __name__ == "__main__":
    startMiner()
    startWallet()
    other_public = b'-----BEGIN PUBLIC KEY-----\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8A
    time.sleep(2)
    print(getBalance(Wallet.my_public))
    sendCoins( other_public, 1.0, 0.1 )
    time.sleep(20)
    print(getBalance(other_public))
    print(getBalance(Wallet.my_public))

    time.sleep(1)
    stopWallet()
    stopMiner()
```

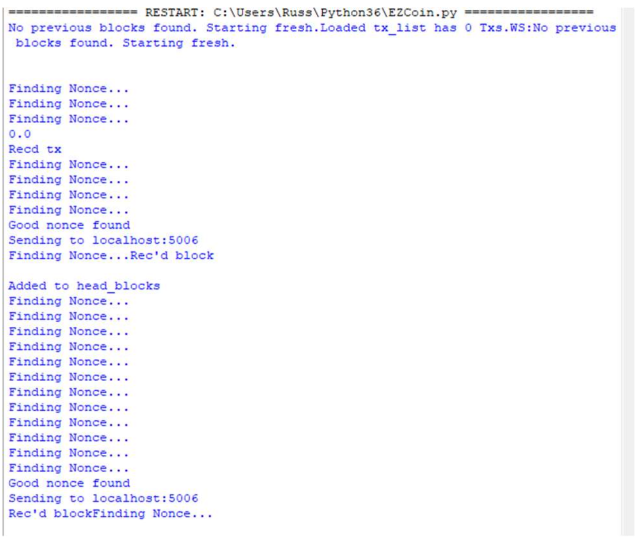
```
        #Verify balances
        if abs(new1-bal1+2.0) > 0.00000001:
            print("Error! Wrong balance for pu1")
        else:
            print("Success. Good balance for pu1")
        if abs(new2-bal2-1.0) > 0.00000001:
```

```
    return False

def nonceFinder(wallet_list, miner_public):
```

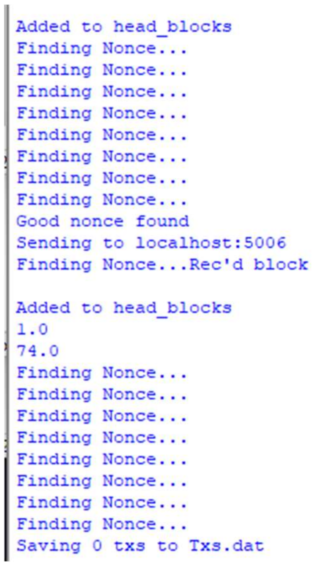
Python 3.6.4 Shell

```
================ RESTART: C:\Users\Russ\Python36\EZCoin.py ================
Loaded tx_list has 0 Txs.Finding Nonce...

Finding Nonce...
Finding Nonce...
0.0
0.0
Recd tx0.0

Finding Nonce...
Finding Nonce...
Finding Nonce...
Finding Nonce...
Finding Nonce...
Saving 1 txs to Txs.dat
>>>
================ RESTART: C:\Users\Russ\Python36\EZCoin.py ================
Loaded tx_list has 1 Txs.Finding Nonce...

Finding Nonce...
Finding Nonce...
0.0
Recd tx
Finding Nonce...
Finding Nonce...
Finding Nonce...
Finding Nonce...
Finding Nonce...
Finding Nonce...
Finding Nonce...
Finding Nonce...
Finding Nonce...
Finding Nonce...
Good nonce found
Sending to localhost:5006
Finding Nonce...Rec'd block

Finding Nonce...
Finding Nonce...
Finding Nonce...
Finding Nonce...
Finding Nonce...
Finding Nonce...
Finding Nonce...
Finding Nonce...
Finding Nonce...
Finding Nonce...
Finding Nonce...
Finding Nonce...
Finding Nonce...
0.0
0.0
Finding Nonce...
Finding Nonce...
```

Added time.sleep(20) after 1st output, then cleared AllBlocks.dat and WalletBlocks.dat



```
================ RESTART: C:\Users\Russ\Python36\EZCoin.py ================
No previous blocks found. Starting fresh.Loaded tx_list has 0 Txs.WS:No previous
 blocks found. Starting fresh.


Finding Nonce...
Finding Nonce...
Finding Nonce...
0.0
Recd tx
Finding Nonce...
Finding Nonce...
Finding Nonce...
Finding Nonce...
Good nonce found
Sending to localhost:5006
Finding Nonce...Rec'd block

Added to head_blocks
Finding Nonce...
Finding Nonce...
Finding Nonce...
Finding Nonce...
Finding Nonce...
Finding Nonce...
Finding Nonce...
Finding Nonce...
Finding Nonce...
Finding Nonce...
Finding Nonce...
Finding Nonce...
Good nonce found
Sending to localhost:5006
Rec'd blockFinding Nonce...
```



```
Added to head_blocks
Finding Nonce...
Finding Nonce...
Finding Nonce...
Finding Nonce...
Finding Nonce...
Finding Nonce...
Finding Nonce...
Finding Nonce...
Good nonce found
Sending to localhost:5006
Finding Nonce...Rec'd block

Added to head_blocks
1.0
74.0
Finding Nonce...
Finding Nonce...
Finding Nonce...
Finding Nonce...
Finding Nonce...
Finding Nonce...
Finding Nonce...
Finding Nonce...
Saving 0 txs to Txs.dat
```

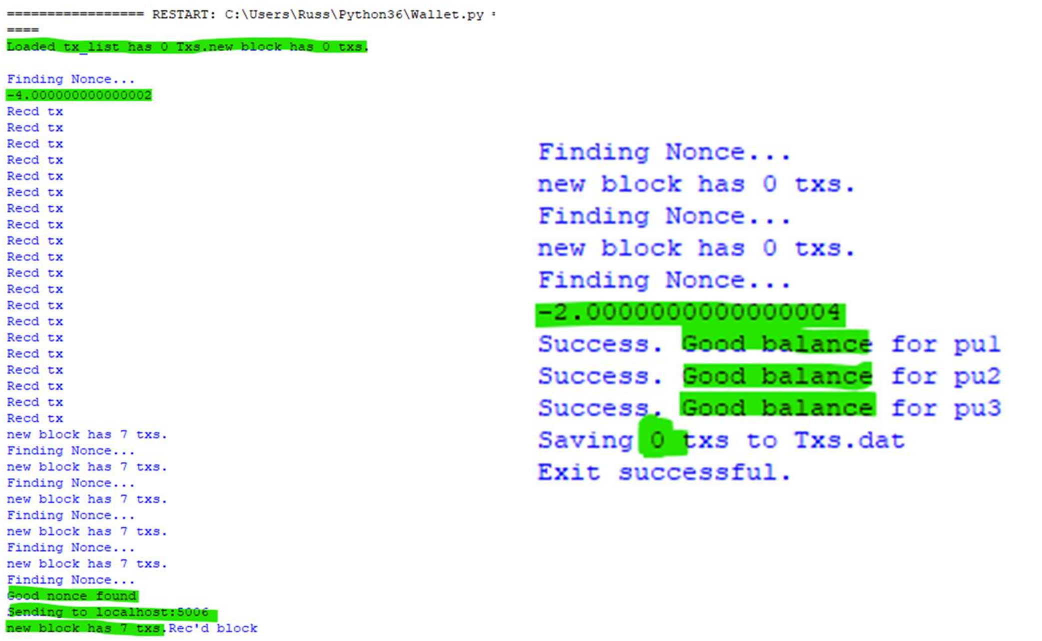
Vid 71

3:02 → everything passes (added check_size fn, and called in is_valid)

4:55 → this block wasn't valid, so it didn't get added to the blockchain that the Wallet is keeping hence the wrong balances

5:30 → added check_size to nonceFinder, added removeTx to TxBlock, (7:21) added placeholder to nonceFinder

9:46 → after Running GOOD BALANCES (remember to clear .dat files)



```
================= RESTART: C:\Users\Russ\Python36\Wallet.py =
====
Loaded tx_list has 0 Txs.new block has 0 txs.

Finding Nonce...
-4.000000000000002
Recd tx
Recd tx
Recd tx
Recd tx
Recd tx
Recd tx
Recd tx
Recd tx
Recd tx
Recd tx
Recd tx
Recd tx
Recd tx
Recd tx
Recd tx
Recd tx
Recd tx
Recd tx
Recd tx
Recd tx
new block has 7 txs.
Finding Nonce...
new block has 7 txs.
Finding Nonce...
new block has 7 txs.
Finding Nonce...
new block has 7 txs.
Finding Nonce...
new block has 7 txs.
Finding Nonce...
Good nonce found
Sending to localhost:5006
new block has 7 txs.Rec'd block
```

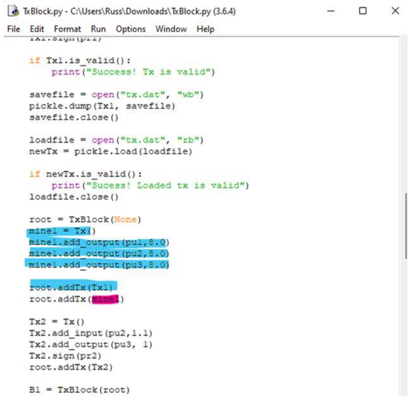
```
Finding Nonce...
new block has 0 txs.
Finding Nonce...
new block has 0 txs.
Finding Nonce...
-2.0000000000000004
Success. Good balance for pu1
Success. Good balance for pu2
Success. Good balance for pu3
Saving 0 txs to Txs.dat
Exit successful.
```

** See assignment 2 folder for Wallet, Miner and TxBlock re-calibrators **

** MinerRecalibrator2 just has verbosity added so we can watch transactions in real time

Vid 72

0:28 → adding mining txs to the root block

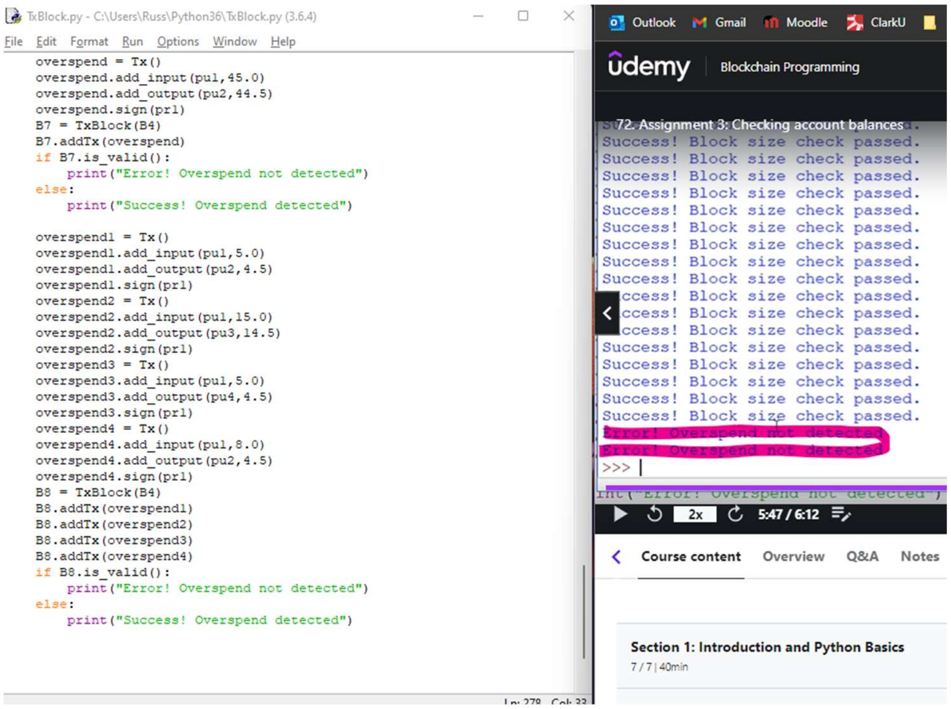


TxBlock.py - C:\Users\Russ\Downloads\TxBlock.py (3.6.4)

File Edit Format Run Options Window Help

```
    Tx1.sign(pr1)

    if Tx1.is_valid():
        print("Success! Tx is valid")

    savefile = open("tx.dat", "wb")
    pickle.dump(Tx1, savefile)
    savefile.close()

    loadfile = open("tx.dat", "rb")
    newTx = pickle.load(loadfile)

    if newTx.is_valid():
        print("Sucess! Loaded tx is valid")
    loadfile.close()

    root = TxBlock(None)
    mine1 = Tx()
    mine1.add_output(pu1,8.0)
    mine1.add_output(pu2,8.0)
    mine1.add_output(pu3,8.0)

    root.addTx(Tx1)
    root.addTx(mine1)

    Tx2 = Tx()
    Tx2.add_input(pu2,1.1)
    Tx2.add_output(pu3, 1)
    Tx2.sign(pr2)
    root.addTx(Tx2)

    B1 = TxBlock(root)
```

5:37 → Creating and Running overspend test cases SHOULD NOT DETECT OVERSPEND yet


TxBlock.py - C:\Users\Russ\Python36\TxBlock.py (3.6.4)

```
    overspend = Tx()
    overspend.add_input(pu1,45.0)
    overspend.add_output(pu2,44.5)
    overspend.sign(pr1)
    B7 = TxBlock(B4)
    B7.addTx(overspend)
    if B7.is_valid():
        print("Error! Overspend not detected")
    else:
        print("Success! Overspend detected")

    overspend1 = Tx()
    overspend1.add_input(pu1,5.0)
    overspend1.add_output(pu2,4.5)
    overspend1.sign(pr1)
    overspend2 = Tx()
    overspend2.add_input(pu1,15.0)
    overspend2.add_output(pu3,14.5)
    overspend2.sign(pr1)
    overspend3 = Tx()
    overspend3.add_input(pu1,5.0)
    overspend3.add_output(pu4,4.5)
    overspend3.sign(pr1)
    overspend4 = Tx()
    overspend4.add_input(pu1,8.0)
    overspend4.add_output(pu2,4.5)
    overspend4.sign(pr1)
    B8 = TxBlock(B4)
    B8.addTx(overspend1)
    B8.addTx(overspend2)
    B8.addTx(overspend3)
    B8.addTx(overspend4)
    if B8.is_valid():
        print("Error! Overspend not detected")
    else:
        print("Success! Overspend detected")
```

Blockchain Programming

72. Assignment 3: Checking account balances

```
Success! Block size check passed.
Success! Block size check passed.
Success! Block size check passed.
Success! Block size check passed.
Success! Block size check passed.
Success! Block size check passed.
Success! Block size check passed.
Success! Block size check passed.
Success! Block size check passed.
Success! Block size check passed.
Success! Block size check passed.
Success! Block size check passed.
Success! Block size check passed.
Success! Block size check passed.
Success! Block size check passed.
Success! Block size check passed.
Success! Block size check passed.
Error! Overspend not detected
Error! Overspend not detected
>>> |
```

5:47 / 6:12

Section 1: Introduction and Python Basics
7 / 7 | 40min

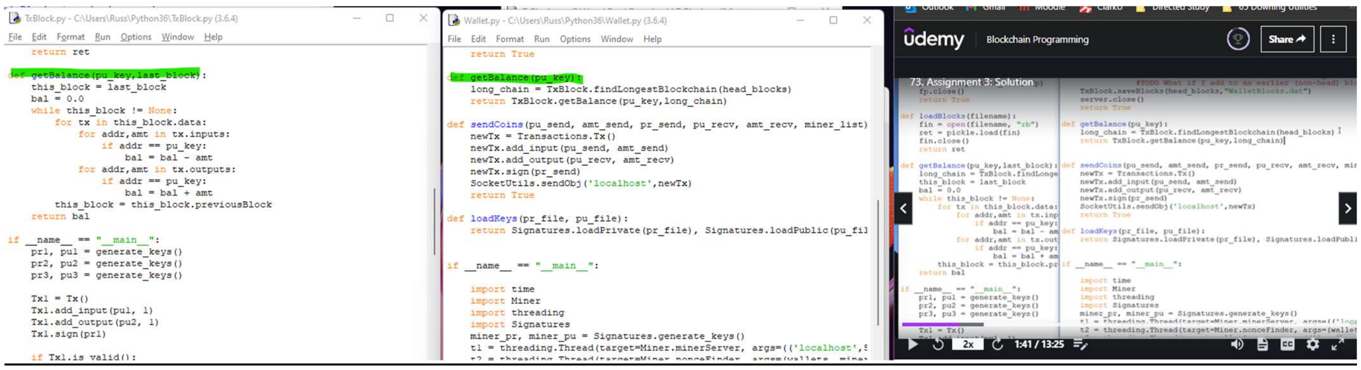
** See assignment 3 for Overspend code that doesn't work yet

Vid 73

implement a check so that that no **pu** block spends its entire balance

0:26 → moving getBalance into TxBlock

Editing getBalance in Wallet


TxBlock.py - C:\Users\Russ\Python36\TxBlock.py (3.6.4)

```
    return ret

def getBalance(pu_key,last_block):
    this_block = last_block
    bal = 0.0
    while this_block != None:
        for tx in this_block.data:
            for addr,amt in tx.inputs:
                if addr == pu_key:
                    bal = bal - amt
            for addr,amt in tx.outputs:
                if addr == pu_key:
                    bal = bal + amt
        this_block = this_block.previousBlock
    return bal

if __name__ == "__main__":
    pr1, pu1 = generate_keys()
    pr2, pu2 = generate_keys()
    pr3, pu3 = generate_keys()

    Tx1 = Tx()
    Tx1.add_input(pu1, 1)
    Tx1.add_output(pu2, 1)
    Tx1.sign(pr1)

    if Tx1.is_valid():
```

Wallet.py - C:\Users\Russ\Python36\Wallet.py (3.6.4)

```
    return True

def getBalance(pu_key):
    long_chain = TxBlock.findLongestBlockchain(head_blocks)
    return TxBlock.getBalance(pu_key,long_chain)

def sendCoins(pu_send, amt_send, pr_send, pu_recv, amt_recv, miner_list)
    newTx = Transactions.Tx()
    newTx.add_input(pu_send, amt_send)
    newTx.add_output(pu_recv, amt_recv)
    newTx.sign(pr_send)
    SocketUtils.sendObj('localhost',newTx)
    return True

def loadKeys(pr_file, pu_file):
    return Signatures.loadPrivate(pr_file), Signatures.loadPublic(pu_fil


if __name__ == "__main__":

    import time
    import Miner
    import threading
    import Signatures
    miner_pr, miner_pu = Signatures.generate_keys()
    t1 = threading.Thread(target=Miner.minerServer, args=(('localhost',5
```

Blockchain Programming

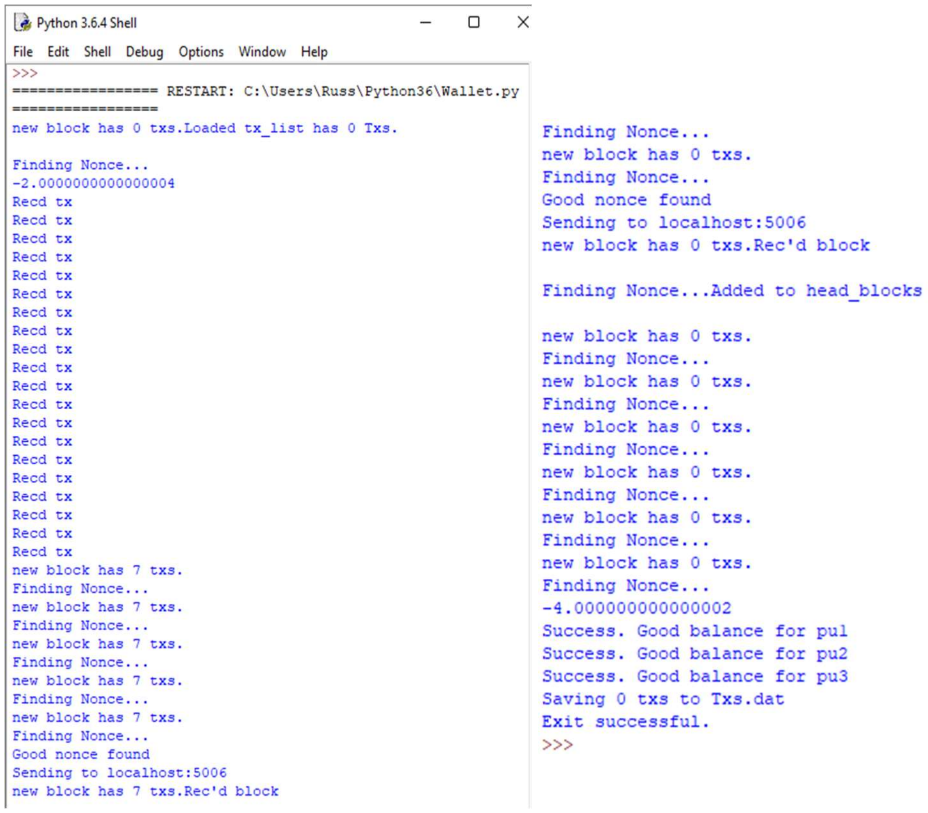
73. Assignment 3: Solution

```
    fp.close()
    return True

def loadBlocks(filename):
    fin = open(filename, "rb")
    ret = pickle.load(fin)
    fin.close()
    return ret

def getBalance(pu_key,last_block):
    long_chain = TxBlock.findLonge
    this_block = last_block
    bal = 0.0
    while this_block != None:
        for tx in this_block.data:
            for addr,amt in tx.inp
                if addr == pu_key:
                    bal = bal - am
            for addr,amt in tx.out
                if addr == pu_key:
                    bal = bal + am
        this_block = this_block.pr
    return bal

if __name__ == "__main__":
    pr1, pu1 = generate_keys()
    pr2, pu2 = generate_keys()
    pr3, pu3 = generate_keys()

    Tx1 = Tx()
```

```
    #TODO What if I add to an earlier (non-head) bl
    TxBlock.saveBlocks(head_blocks,"WalletBlocks.dat")
    server.close()
    return True

def getBalance(pu_key):
    long_chain = TxBlock.findLongestBlockchain(head_blocks)
    return TxBlock.getBalance(pu_key,long_chain)

def sendCoins(pu_send, amt_send, pr_send, pu_recv, amt_recv, mir
    newTx = Transactions.Tx()
    newTx.add_input(pu_send, amt_send)
    newTx.add_output(pu_recv, amt_recv)
    newTx.sign(pr_send)
    SocketUtils.sendObj('localhost',newTx)
    return True

def loadKeys(pr_file, pu_file):
    return Signatures.loadPrivate(pr_file), Signatures.loadPubli

if __name__ == "__main__":

    import time
    import Miner
    import threading
    import Signatures
    miner_pr, miner_pu = Signatures.generate_keys()
    t1 = threading.Thread(target=Miner.minerServer, args=(('loca
    t2 = threading.Thread(target=Miner.nonceFinder, args=(wallet
```

1:41 / 13:25

2:21 → Running Wallet, still working correctly and we are getting valid blocks so the balances are still good



Python 3.6.4 Shell

File Edit Shell Debug Options Window Help

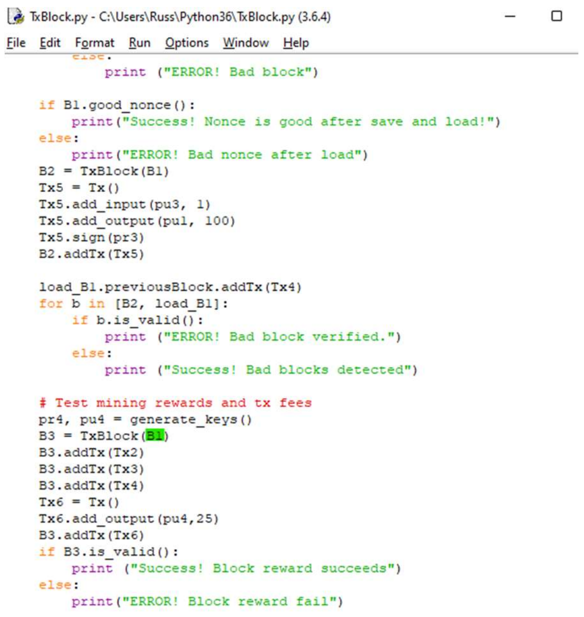
```
>>>
================ RESTART: C:\Users\Russ\Python36\Wallet.py
================
new block has 0 txs.Loaded tx_list has 0 Txs.

Finding Nonce...
-2.0000000000000004
Recd tx
Recd tx
Recd tx
Recd tx
Recd tx
Recd tx
Recd tx
Recd tx
Recd tx
Recd tx
Recd tx
Recd tx
Recd tx
Recd tx
Recd tx
Recd tx
Recd tx
Recd tx
Recd tx
Recd tx
new block has 7 txs.
Finding Nonce...
new block has 7 txs.
Finding Nonce...
new block has 7 txs.
Finding Nonce...
new block has 7 txs.
Finding Nonce...
new block has 7 txs.
Finding Nonce...
Good nonce found
Sending to localhost:5006
new block has 7 txs.Rec'd block
Finding Nonce...
new block has 0 txs.
Finding Nonce...
Good nonce found
Sending to localhost:5006
new block has 0 txs.Rec'd block

Finding Nonce...Added to head_blocks

new block has 0 txs.
Finding Nonce...
new block has 0 txs.
Finding Nonce...
new block has 0 txs.
Finding Nonce...
new block has 0 txs.
Finding Nonce...
new block has 0 txs.
Finding Nonce...
new block has 0 txs.
Finding Nonce...
-4.000000000000002
Success. Good balance for pu1
Success. Good balance for pu2
Success. Good balance for pu3
Saving 0 txs to Txs.dat
Exit successful.
>>>
```

2:38 → for TxBlock when we ask if something is_valid, need to be checking the spending that happens in there.

Making a Dict out of the addresses and the this that users have spent.

6:26 → after Running detected overspend

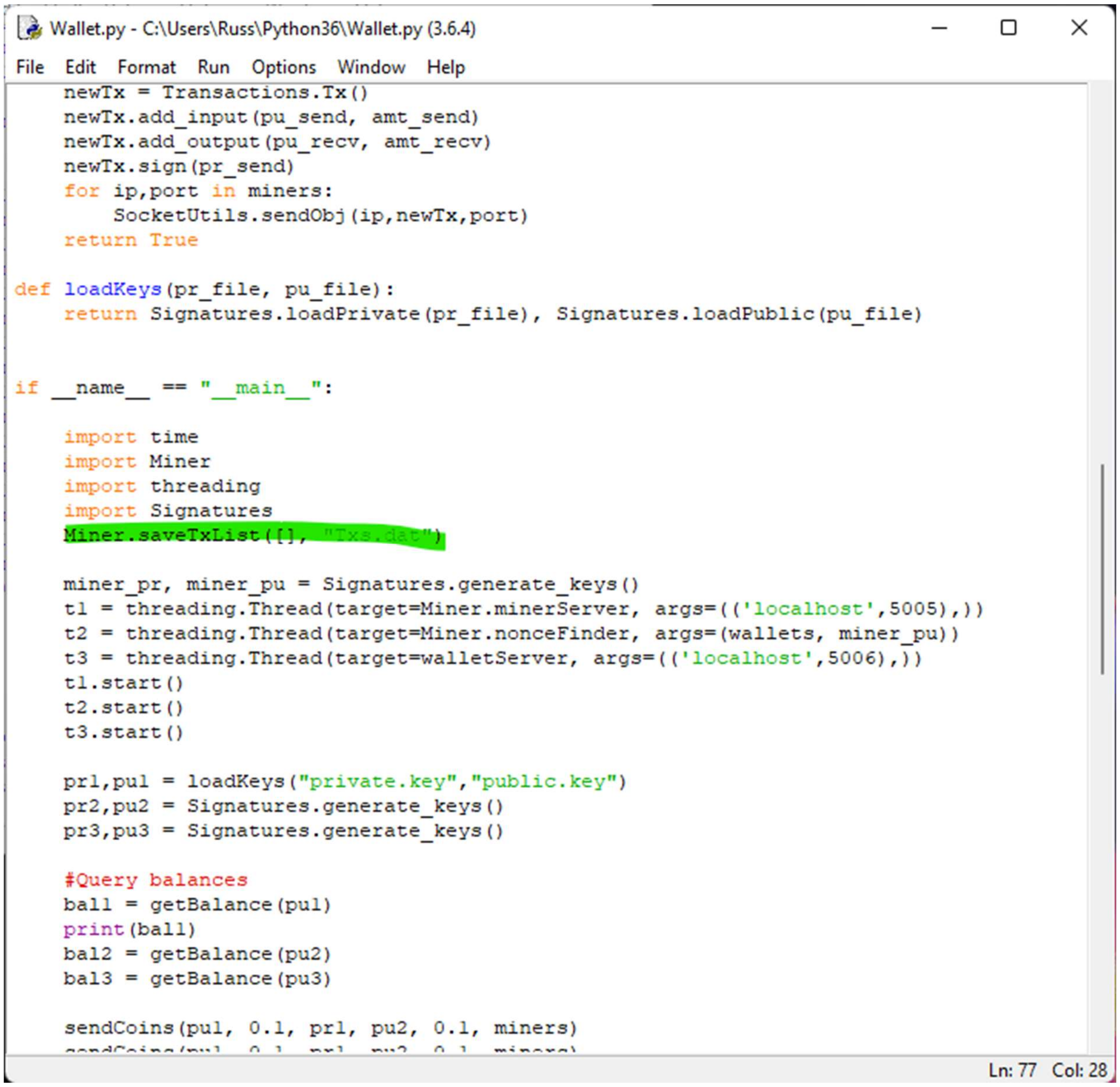
Change B3 inheritance to B1 because B2 is a bad block



TxBlock.py - C:\Users\Russ\Python36\TxBlock.py (3.6.4)

File Edit Format Run Options Window Help

```
            print ("ERROR! Bad block")

    if B1.good_nonce():
        print("Success! Nonce is good after save and load!")
    else:
        print("ERROR! Bad nonce after load")
    B2 = TxBlock(B1)
    Tx5 = Tx()
    Tx5.add_input(pu3, 1)
    Tx5.add_output(pu1, 100)
    Tx5.sign(pr3)
    B2.addTx(Tx5)

    load_B1.previousBlock.addTx(Tx4)
    for b in [B2, load_B1]:
        if b.is_valid():
            print ("ERROR! Bad block verified.")
        else:
            print ("Success! Bad blocks detected")

    # Test mining rewards and tx fees
    pr4, pu4 = generate_keys()
    B3 = TxBlock(B1)
    B3.addTx(Tx2)
    B3.addTx(Tx3)
    B3.addTx(Tx4)
    Tx6 = Tx()
    Tx6.add_output(pu4,25)
    B3.addTx(Tx6)
    if B3.is_valid():
        print ("Success! Block reward succeeds")
    else:
        print("ERROR! Block reward fail")
```


Wallet.py - C:\Users\Russ\Python36\Wallet.py (3.6.4)
File Edit Format Run Options Window Help
Ln: 77 Col: 28

```
    newTx = Transactions.Tx()
    newTx.add_input(pu_send, amt_send)
    newTx.add_output(pu_recv, amt_recv)
    newTx.sign(pr_send)
    for ip,port in miners:
        SocketUtils.sendObj(ip,newTx,port)
    return True

def loadKeys(pr_file, pu_file):
    return Signatures.loadPrivate(pr_file), Signatures.loadPublic(pu_file)


if __name__ == "__main__":

    import time
    import Miner
    import threading
    import Signatures
    Miner.saveTxList([], "Txs.dat")

    miner_pr, miner_pu = Signatures.generate_keys()
    t1 = threading.Thread(target=Miner.minerServer, args=(('localhost',5005),))
    t2 = threading.Thread(target=Miner.nonceFinder, args=(wallets, miner_pu))
    t3 = threading.Thread(target=walletServer, args=(('localhost',5006),))
    t1.start()
    t2.start()
    t3.start()

    pr1,pu1 = loadKeys("private.key","public.key")
    pr2,pu2 = Signatures.generate_keys()
    pr3,pu3 = Signatures.generate_keys()

    #Query balances
    bal1 = getBalance(pu1)
    print(bal1)
    bal2 = getBalance(pu2)
    bal3 = getBalance(pu3)

    sendCoins(pu1, 0.1, pr1, pu2, 0.1, miners)
```