Sheet 3

Huipeng Zeng 5156067
Annette Eschenbach 4887801

1. I used the contract example - MyToken from https://www.ethereum.org/token.
The functionality is that, the contract owner can put certain amount values into another account.
And it also allocates 5000 balance to contract owner.

2. Code of contract – Simple as follow:

```
pragma solidity >= 0.4.22 < 0.6.0;

contract Simple {

        mapping (address => uint256) public balanceOf;

        constructor() public {
                balanceOf[msg.sender]=5000;
                }

        function transfer(address _to, uint256 _value) public returns (bool success){
                require(balanceOf[msg.sender] >= _value);
                require(balanceOf[_to] + _value >= balanceOf[_to]);
                balanceOf[msg.sender]-=_value;
                balanceOf[_to]+=_value;
                return true;
        }
}
```

define following parameters to create the contract in block.

1)

```
var contractTemplate
=web3.eth.contract(JSON.parse(contractContent.contracts["simple.sol:Simple"].abi))
```

2)

```
var gasValue = eth.estimateGas({data:"0x"+contractContent.contracts["simple.sol:Simple"].bin})
```

3)

```
var contractInst = contractTemplate.new({from: eth.accounts[1],
data:"0x"+contractContent.contracts["simple.sol:Simple"].bin,gas:gasValue},function(e,contract)
{ if(typeof contract.address !== 'undefined'){console.log('Contract mined! address:'+
contract.address + 'transactionHash:'+contract.transactionHash);}});
```

3. I used eth.contract([contract name].address) to check the information.

```
> eth.contract(contractInst.address)
{
  abi: "0xdb79ffb797cc8f1806cf67cc7d010800a98dd2fe",
  eth: {
    accounts: ["0xd17f08773b79984b814e0922ed74048195ebfeda", "0x02b75772d8
0e8a57f7d96"],
    blockNumber: 66573,
    coinbase: "0xd17f08773b79984b814e0922ed74048195ebfeda",
    compile: {
      lll: function(),
      serpent: function(),
      solidity: function()
    },
    defaultAccount: undefined,
```

According to “blockNumer: 66573”, I think this contract is in block 66573.

4. a specication of how your contract can be used
1) to look up the balance.
Using [contract name].balanceOf([account])
2)
By using [contract name].transfer.call(eth.accounts[0], web3.toWei(1,"ether")) to run a transaction in local EVM, while by using [contract name].transfer.sendTransaction(eth.accounts[0], web3.toWei(1,"ether", {from:eth.accounts[1]}) , transaction will be recorded on blockchain. But our contract doesn’t work well somehow. (always return false, so means that cannot make the transaction)

5. a documentation of your usage in at least one transaction, such that your transaction can be found on the blockchain.