

## Homework 4

1. a) + b) Associating user's addresses with balance: mapping from address to value, i.e.

```
mapping(address => uint) balance;
```

2. a) + b) How to read balance from contract? Either: Getter method (i.e. `getBalance(...)`), or make the balance mapping public (i.e. `mapping(address => uint) public balance;` ).

3. All total supply to owner of contract: `balance[owner] = totalSupply;`

4. a) Sender's address not required as function is called by a user whose address is known when the function call is made. When the function is called, `msg.sender` is a given parameter which can be used without requiring manual input.

b) If sender's address could be given as a parameter, a third-party user could send an amount from a wallet which is not their own to another wallet, i.e. tokens could be stolen from wallets by using the transfer function.

5. `event Transfer(uint _amount, address _recipientAddress)`

6. 

```
struct Payment {  
    uint amount;  
    address recipientAddress;  
}
```

7. 

```
function getPayments(address _address) public view returns (Payment[]  
memory) {  
    return payments[_address];  
}
```