```
pragma solidity ^0.4.19;

contract TipJar {

   address owner;    // current owner of the contract

   function TipJar() public {
      owner = msg.sender;
   }

   function withdraw() public {
      require(owner == msg.sender);
      msg.sender.transfer(address(this).balance);
   }

   function deposit(uint256 amount) public payable {
      require(msg.value == amount);
   }

   function getBalance() public view returns (uint256) {
      return address(this).balance;
   }
}
```

- **code to accept deposits and track account balances:**

```
pragma solidity ^0.4.19;

contract Bank {

   mapping(address => uint256) public balanceOf;   // balances, indexed
by addresses

   function deposit(uint256 amount) public payable {
      require(msg.value == amount);

      balanceOf[msg.sender] += amount;     // adjust the account's balance
   }
}
```

- **Code for Withdrawals and Account Balances**

```
pragma solidity ^0.4.19;

contract Bank {

  mapping(address => uint256) public balanceOf;   // balances, indexed by addresses

  function deposit(uint256 amount) public payable {
    require(msg.value == amount);
    balanceOf[msg.sender] += amount;     // adjust the account's balance
  }

  function withdraw(uint256 amount) public {
    require(amount <= balanceOf[msg.sender]);
    balanceOf[msg.sender] -= amount;
    msg.sender.transfer(amount);
  }
}
```