

```

// SPDX-License-Identifier: MIT
pragma solidity >=0.6.2 <0.9.0;

contract SimpleBank {
    uint8 private clientCount;
    mapping (address => uint) private balances;
    address public owner;

    // Log the event about a deposit being made by an address and its
    amount
    event LogDepositMade(address indexed accountAddress, uint amount);

    // Constructor is "payable" so it can receive the initial funding of
    30,
    // required to reward the first 3 clients
    constructor() public payable {
        require(msg.value == 30 ether, "30 ether initial funding
required");
        /* Set the owner to the creator of this contract */
        owner = msg.sender;
        clientCount = 0;
    }

    /// @notice Enroll a customer with the bank,
    /// giving the first 3 of them 10 ether as reward
    /// @return The balance of the user after enrolling
    function enroll() public returns (uint) {
        if (clientCount < 3) {
            clientCount++;
            balances[msg.sender] = 10 ether;
        }
        return balances[msg.sender];
    }

    /// @notice Deposit ether into bank, requires method is "payable"
    /// @return The balance of the user after the deposit is made
    function deposit() public payable returns (uint) {
        balances[msg.sender] += msg.value;
        emit LogDepositMade(msg.sender, msg.value);
        return balances[msg.sender];
    }

    /// @notice Withdraw ether from bank
    /// @return remainingBal The balance remaining for the user
    function withdraw(uint withdrawAmount) public returns (uint
remainingBal) {
        // Check enough balance available, otherwise just return balance
        if (withdrawAmount <= balances[msg.sender]) {
            balances[msg.sender] -= withdrawAmount;
            payable(msg.sender).transfer(withdrawAmount);
        }
        return balances[msg.sender];
    }

    /// @notice Just reads balance of the account requesting, so
    "constant"
    /// @return The balance of the user
    function balance() public view returns (uint) {

```

```
        return balances[msg.sender];
    }

    /// @return The balance of the Simple Bank contract
    function depositsBalance() public view returns (uint) {
        return address(this).balance;
    }
}
```