//This is a banking application sample code to do normal banking transactions

pragma solidity ^0.4.19;

//Contract is the highest level in solidity ,Here we are defining the contract -- Infact it resembles as class in oops

contract gunturBank{

//state declaration

address client; //address is a datatype which is capable of storing the hashed account address

bool \_switch=false; //boolean data type which can store true or false

//Declaring the construtor for the contract , Name of the constructor should be same as contract name,Constructor will be called once during deploying

function gunturBank() public {

client =msg.sender; //defining owner of the contract, msg.sender is a builtin which contains the value of the address one who deploying the contract

}

//Defining the modifier , in simple terms we are defining custom visibility for the functions in addtion to public,private

modifier ifOwner(){

if (client!=msg.sender){ //Checking whether Owner of contract or not

throw; //stops the execution of urther instrucions in the contract

}

\_; //continues the execution of function code

}

//declaring the payable function which accepts the funds and stores in the contract address (i.e., contract address acts as account number of your bank account)

//payable is the keyword in solidity which makes the function to accept the ether

function depositFunds() public payable{

}

//function to withdraw funds , here we withdraw amount from the account number(contract address) to your wallet(Metamask address)

function withDrawFunds(uint amount) public ifOwner{

if(client.send(amount)){

\_switch=true;

}

else {

\_switch=false;

}

}

//function to retrieve the funds available in the contract

function getfunds() constant public ifOwner returns(uint){

return this.balance;//balance is the keyword in solidity which returns balance of the contract

}

}