//ASSIGNMENT 3

// SPDX-License-Identifier: GPL-3.0

pragma solidity ^0.8.0;

contract bank{

uint balance=0;

address public accholder;

constructor(){

accholder=msg.sender;

}

function deposit() public payable{

require(msg.value>0,"AMOUNT SHOULD BE GREATER THAN 0");

require(msg.sender==accholder,"YOU ARE NOT HTE ACCOUNT HOLDER");

balance += msg.value;

}

function withdrawl() public payable{

require(balance>0,"AMOUNT SHOULD BE GREATER THAN 0");

require(msg.sender==accholder,"YOU ARE NOT HTE ACCOUNT HOLDER");

payable(msg.sender).transfer(msg.value);

}

function showBalance() public view returns(uint){

return balance;

}

}

//ASSIGNMENT 4

// SPDX-License-Identifier: GPL-3.0

pragma solidity >=0.7.0 <0.9.0;

contract StudentManagement {

struct Student {

int studId;

string name;

string department;

}

Student[] private students;

function addStudent(int studId, string memory name, string memory department) public {

Student memory newStudent = Student(studId, name, department);

students.push(newStudent);

}

function getStudent(int studId) public view returns (string memory, string memory) {

for (uint i = 0; i < students.length; i++) {

Student memory student = students[i];

if (student.studId == studId) {

return (student.name, student.department);

}

}

return ("Name Not Found", "Department Not Found");

}

// Fallback function to add a default student if contract receives unexpected data

fallback() external {

students.push(Student(7, "XYZ", "Mechanical"));

}

}