Practical 6

1. User defined modifiers

```
pragma solidity ^0.7.0;
//contract name is MyFirstContract
contract MyFirstContract {
//create two variables. A sting and an address
address owner;
string private name;
//constructor sets the creator of the contract to the owner variable
constructor() {
owner = msg.sender;
}
//modifier checks that the caller of the function is the owner
modifier onlyOwner() {
require(msg.sender == owner, 'Not Owner');
_;
}
//set name. Only the owner of the contract can call because a modifier is
specified
function setName(string memory newName) public onlyOwner{
name = newName;
}
//get the name
function getName () public view returns (string memory) {
return name;
}
}
```

2. Userdefined contract communication

```
pragma solidity ^0.8.0;
contract Calculator{
function add(int a, int b) external pure returns(int){
return a+b;
}
function multiply(int a, int b) external pure returns(int){
return a*b;
}
}
contract Foo{
Calculator calc = new Calculator();
function FourTimesSix() external view returns(int){
return calc.multiply(4,6);
}
function FourPlusSix() external view returns(int){
return calc.add(4,6);
}
}
```