pragma solidity ^0.6.0;

contract Escrow {

// Indicates the current state of the transaction

enum State {AWAITING\_PAYMENT, AWAITING\_DELIVERY, COMPLETE}

// variable to hold the state

State public currentState;

// buyer address

address public buyer;

// seller address

address payable public seller;

// Ensure that only the buyer can call functions the modifier is attached to

modifier onlyBuyer() {

require(msg.sender == buyer, "Only the buyer can call this function!");

// do not forget the underscore character, which is where the function body will be attached

// the underscore can be positioned wherever you need it to be in the modifier

\_;

}

// Set up the buyer and seller addresses

constructor(address \_buyer, address payable \_seller) public {

buyer = \_buyer;

seller = \_seller;

}

// The buyer calls this function to deposit Ether to the contract

function deposit() onlyBuyer external payable {

require(currentState == State.AWAITING\_PAYMENT, "Buyer has already paid!");

currentState = State.AWAITING\_DELIVERY;

}

// The buyer calls this function to confirm that they have received the item for sale

function confirmDelivery() onlyBuyer external {

require(currentState == State.AWAITING\_DELIVERY, "Cannot confirm delivery!");

seller.transfer(address(this).balance);

currentState = State.COMPLETE;

}

}