const object1 = {

message: 'hello',

};

object1.message = 'Good job!';

In JavaScript, when you declare an object using const, you're making the variable reference immutable, not the object itself.

**Immutable Reference**: You cannot reassign the variable to point to a different object. For example, the following would produce an error

const object1 = {

        message: 'hello'

      };

      object1 = {

        message: 'hi'

      };



**Mutable Object**: However, the properties of the object itself can still be modified. You can add, change, or delete properties within the object without any issues

 const object1 = {

        message: 'hello'

      };

      object1.message = 'Good job!'; //This is allowed

      object1.price=5000; //This is also allowed

question 2:

function display(ans) {

  if (ans==undefined) {  // == operator Checks for both null and undefined

    console.log('No value was passed');

  } else {

    console.log('Value passed: ', ans);

  }

}

display();      // == will Log: No value was passed    === will Log: No value was passed

display(null);  // == will Log: No value was passed    === Value passed: null

display(42);