ASSIGNMENT:02

DAY:01

JAVA SCRIPT

1)Rewrite the following code using a ternary operator:

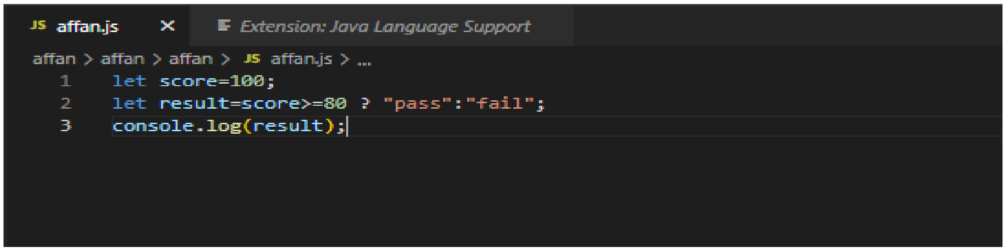
let result;

if (score >= 80) {

result = "Pass";} else {

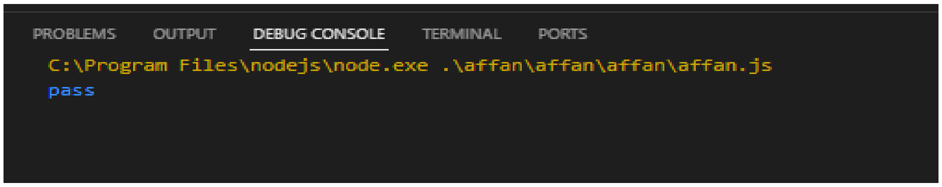
result = "Fail";}

SOURCE CODE:


affan.js
Extension: Java Language Support
affan > affan > affan > affan.js > ...

```
let score=100;
let result=score>=80 ? "pass":"fail";
console.log(result);
```

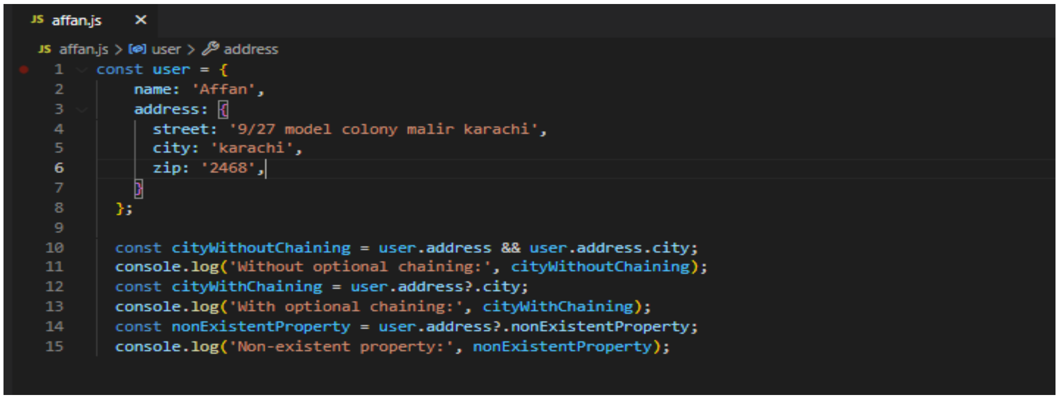
OUTPUT:



```
C:\Program Files\nodejs\node.exe .\affan\affan\affan\affan.js
pass
```

2) How does the optional chaining operator (?.) work, and how can it be used to access nested properties of an object?

SOURCE CODE:

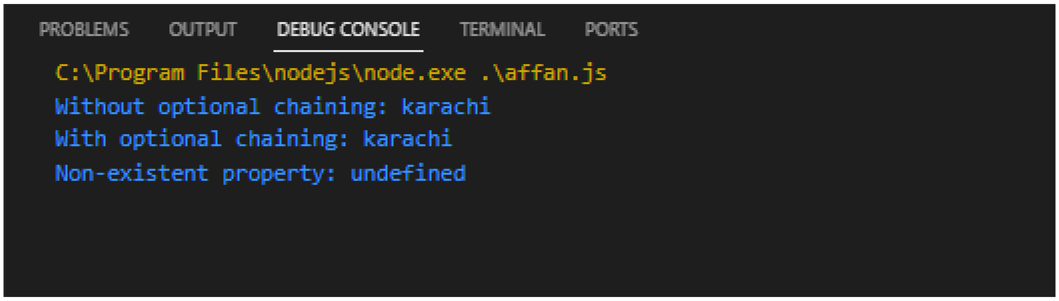


```
const user = {
    name: 'Affan',
    address: {
      street: '9/27 model colony malir karachi',
      city: 'karachi',
      zip: '2468',
    }
  };

  const cityWithoutChaining = user.address && user.address.city;
  console.log('Without optional chaining:', cityWithoutChaining);
  const cityWithChaining = user.address?.city;
  console.log('With optional chaining:', cityWithChaining);
  const nonExistentProperty = user.address?.nonExistentProperty;
  console.log('Non-existent property:', nonExistentProperty);
```

ASSIGNMENT:02
DAY:01

OUTPUT:

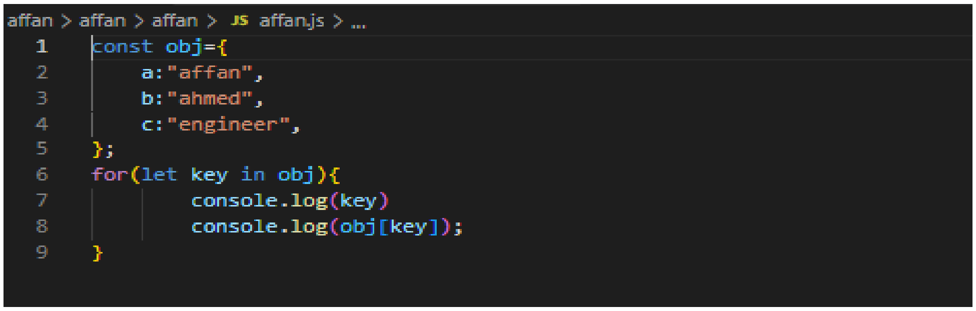


```
C:\Program Files\nodejs\node.exe .\affan.js
Without optional chaining: karachi
With optional chaining: karachi
Non-existent property: undefined
```

3) Compare the for...in loop and the for...of loop in terms of their use cases and the types of values they iterate over.

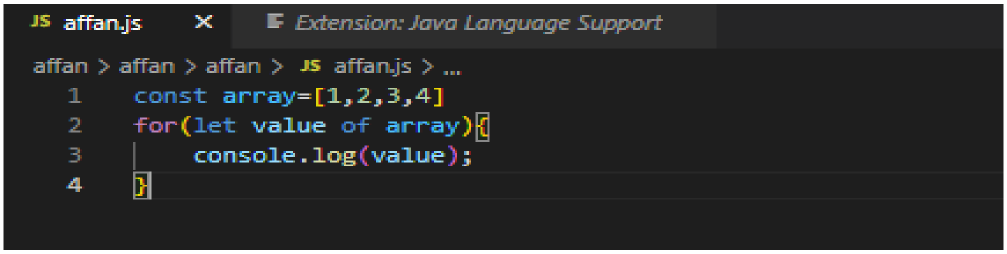
SOURCE CODE:

For in loop



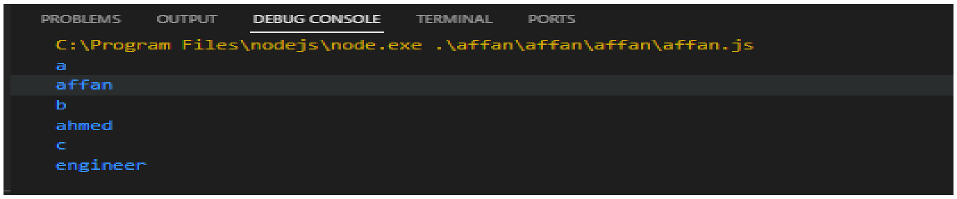
```
const obj={
    a:"affan",
    b:"ahmed",
    c:"engineer",
};
for(let key in obj){
        console.log(key)
        console.log(obj[key]);
}
```

For of loop


JS affan.js ×
Extension: Java Language Support
affan > affan > affan > JS affan.js > ...

```
const array=[1,2,3,4]
for(let value of array){
    console.log(value);
}
```

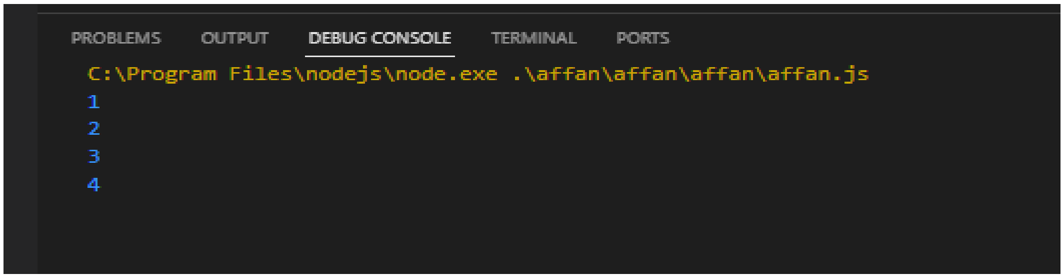
OUTPUT: (for in loop)



```
C:\Program Files\nodejs\node.exe .\affan\affan\affan\affan.js
a
affan
b
ahmed
c
engineer
```

ASSIGNMENT:02
DAY:01

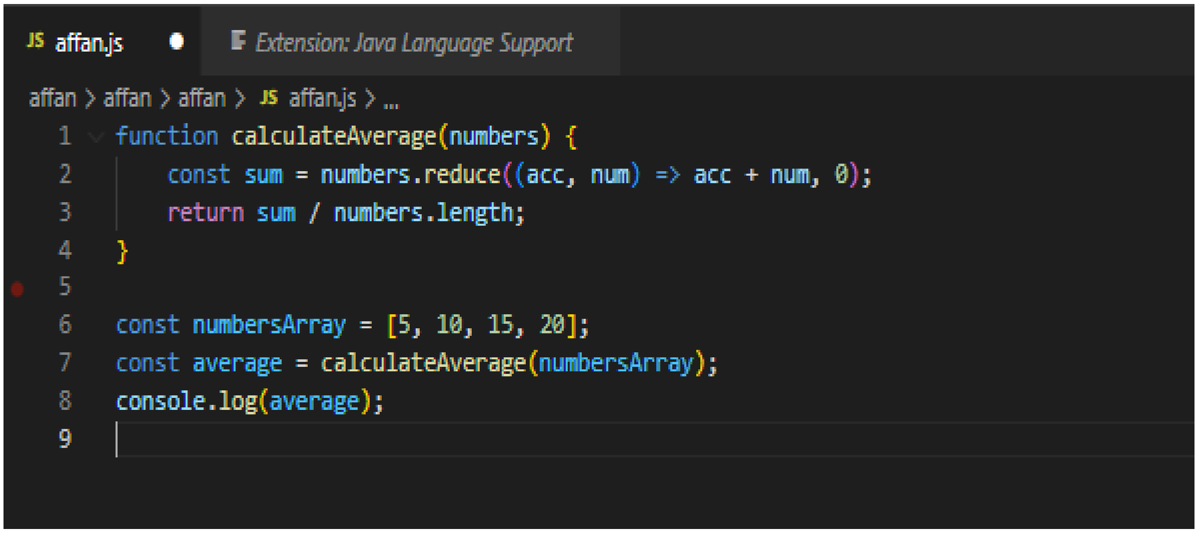
OUTPUT: (for of loop)



```
C:\Program Files\nodejs\node.exe .\affan\affan\affan\affan.js
1
2
3
4
```

4) 4. Define a function calculateAverage that takes an array of numbers as an argument and returns the average value.

SOURCE CODE:

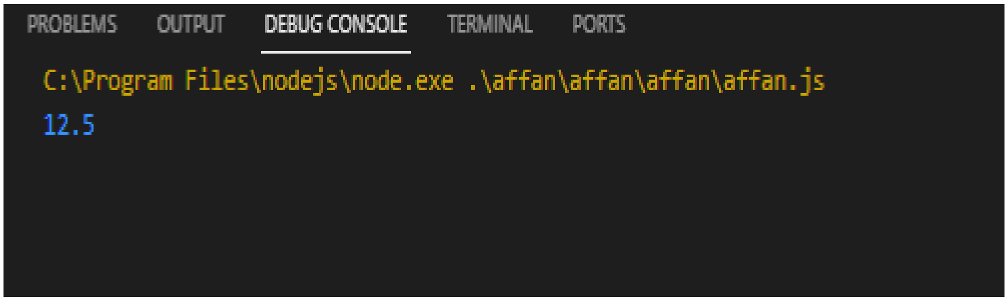

affan.js
Extension: Java Language Support
affan > affan > affan > affan.js > ...

```javascript
function calculateAverage(numbers) {
    const sum = numbers.reduce((acc, num) => acc + num, 0);
    return sum / numbers.length;
}

const numbersArray = [5, 10, 15, 20];
const average = calculateAverage(numbersArray);
console.log(average);

```

OUTPUT:

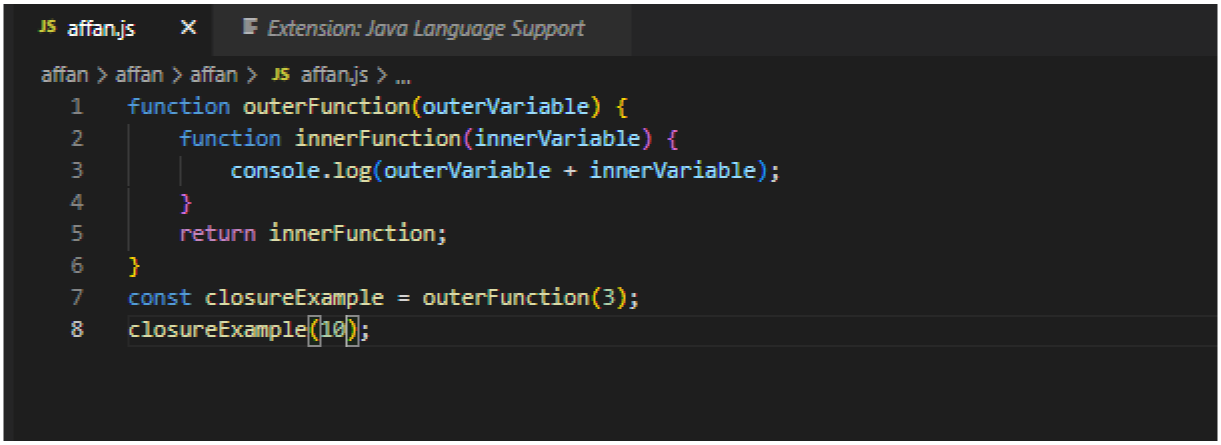


```
C:\Program Files\nodejs\node.exe .\affan\affan\affan\affan.js
12.5
```

ASSIGNMENT:02
DAY:01

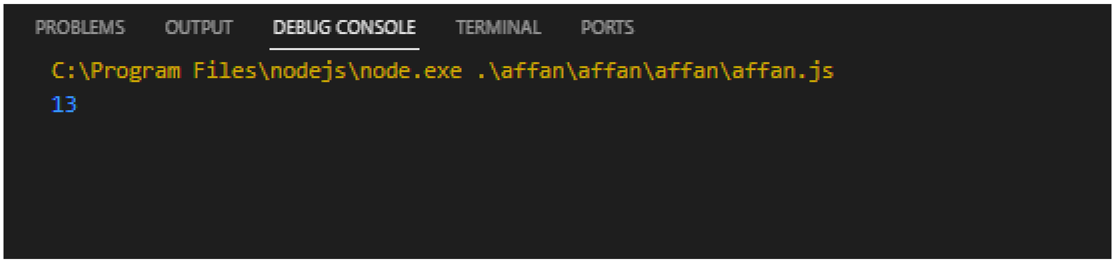
5) Explain the concept of "closures" in JavaScript and provide an example of their practical use.

SOURCE CODE:


JS affan.js ×
Extension: Java Language Support
affan > affan > affan > JS affan.js > ...

```javascript
function outerFunction(outerVariable) {
    function innerFunction(innerVariable) {
        console.log(outerVariable + innerVariable);
    }
    return innerFunction;
}
const closureExample = outerFunction(3);
closureExample(10);
```

OUTPUT:



```
C:\Program Files\nodejs\node.exe .\affan\affan\affan\affan.js
13
```