PRACTICAL-7

Demonstrate ES6 New Syntax and Test Browser Compatibility.

1) Arrow Function

Source code :

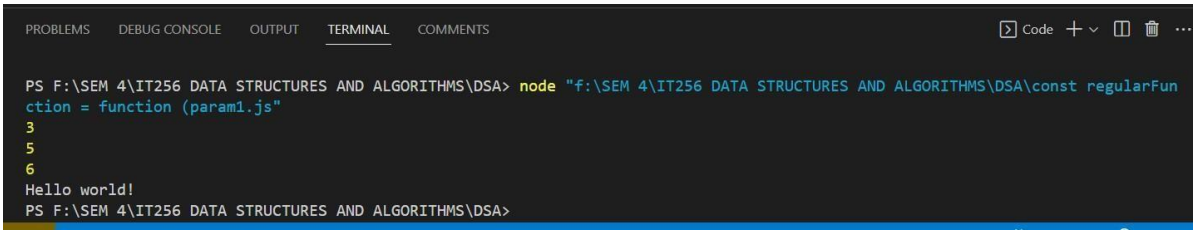
```
const regularFunction = function (param1, param2) {
    return param1 + param2;
  };

  const arrowFunction = (param1, param2) => param1 + param2;


  const arrowFunctionMultiParams = (param1, param2, param3) => {
    return param1 + param2 + param3;
  };


  const noParameters = () => "Hello world!";

  console.log(regularFunction(1,2));
  console.log(arrowFunction(2,3));
  console.log(arrowFunctionMultiParams(1,2,3));
  console.log(noParameters());
```



2) Destructuring

Source code :

```
let myNumbers = [1, 2, 3, 4];
let [firstNum, secondNum, thirdNum, fourthNum] = myNumbers;

console.log(firstNum);
console.log(secondNum);
console.log(thirdNum);
console.log(fourthNum);


let myObject = {
    name: 'John',
    age: 25,
    job: 'Software Engineer'
};

let {name, age, job} = myObject;

console.log(myObject.name);
console.log(age);
console.log(job); // Software Engineer
```

```
PROBLEMS  DEBUG CONSOLE  OUTPUT  TERMINAL  COMMENTS

2
3
4
John
25
Software Engineer
PS F:\SEM 4\IT256 DATA STRUCTURES AND ALGORITHMS\DSA>
```

3) Spread vs Rest Operator

Source code :

```
const numbers = [1,2,3,4,5];

function add(a, b, c, d, e) {
  return a + b + c + d + e;
}

const sum = add(...numbers);
console.log(sum);


const numbers2 = [6,7,8,9,10];

const combinedNumbers = [...numbers, ...numbers2];
console.log(combinedNumbers);
```

```
PROBLEMS  DEBUG CONSOLE  OUTPUT  TERMINAL  COMMENTS

le.js"
15
[
  1, 2, 3, 4,  5,
  6, 7, 8, 9, 10
]
PS F:\SEM 4\IT256 DATA STRUCTURES AND ALGORITHMS\DSA>
```