Course: IS312 – Web Design and Programming

Quarter: Fall 2022

Trevor Sparks

**Task #1**

*// Upper and lower limit set*

const upperLimit: number = 21;

const lowerLimit: number = 0;

*// Initialization of userInput var*

let userInput:number;

*//Function to get user input*

function getUserInput(): number{

*// Use prompt to prompt the user for input, and the number function to*

*// convert the recived string to a numeber*

    userInput = Number(prompt('Please input a number between 1 and 20'));

*// call the validate user function to make sure the user input a valid number*

    validateUserInput(upperLimit, lowerLimit, userInput);

    return userInput;

}

*// function to validate user input. It checks to see if the users input is in between two*

*// pre-specified values*

function validateUserInput(upperLimit: number, lowerLimit: number, userInput: number){

    if(userInput > lowerLimit && userInput < upperLimit){

        return userInput;

    }else{

        console.log("The number is too high or too low");

        getUserInput();

    }

}

**Task #2:**

*// Task 2 function*

function someFunction(p1: number, p2: number):number{

    if(p1 <= 0){

        return p2;

    }else{

        p2 = p1 \* p2;

        p1--;

        return someFunction(p1,p2);

    }

}

**Task #3**

*// recursive function to calculate the Fibonacci sequence at the user specified index*

function calcFibb(input:number){

    if(input === 1){

        return 0;

    }else if(input === 2){

        return 1;

    }else{

        return calcFibb(input-2)+calcFibb(input-1);

    }

}

**Task #4:**

After talking with you on Monday to discuss this programming assignment, I felt like I started to make really good progress. Originally, I believed that the reason I was unable to complete the assignment was because I saved it to for last minute, but the truth was that I really did not understand recursion at all. After we met and talked through what recursion actually was, the code seemed to fall into place.

The biggest problem that I ran into this week was outputting the information that I received from the functions I wrote to a webpage. Originally my HTML page looked like this:

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>PE05 - Trevor Sparks</title>

    <script type="text/javascript" src="PE05.js"></script>

</head>

<body>

    <h1>Click the button to perform calculations and get user input</h1>

    <button onclick="whenTheButtonIsClicked();">Click me!</button>

    <p id="userInput">The user input:</p>

    <p id="task2FunctionAnswer">Task 2 function answer:</p>

    <p id="calcedFib">The Fibonacci value and the users input index: </p>

</body>

</html>

If you look in the body, the paragraph elements all have different ids, I tried to edit the inner text of these elements, but all that happened is that I removed the text that was there, and replaced it with the value of the function. For example, for the paragraph tag with the id userInput, the inner text was “The user input: ”, but after I tried to edit it, it simply became the value of the user input.

To fix this problem I changed the HTML code:

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>PE05 - Trevor Sparks</title>

    <script type="text/javascript" src="PE05.js"></script>

</head>

<body>

    <h1>Click the button to perform calculations and get user input</h1>

    <button onclick="whenTheButtonIsClicked();">Click me!</button>

   <div id="output"></div>

</body>

</html>

I created a div element and then used my TypeScript code to populate the inner HTML of that div element as follows:

let outputHTML = document.getElementById('output');

outputHTML.innerHTML = `<p>The user input: ${userInput}</p>

                        <p>Task 2 function answer: ${task2FunctionAnswer}</p>

                        <p>The value of the Fibonacci sequence at the

${userInput} position is ${calcedFib}</p>`;

I am not sure if this the best way to output data to an HTML page, but it work. My full HTML and TS code is below:

Index.html

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>PE05 - Trevor Sparks</title>

    <script type="text/javascript" src="PE05.js"></script>

</head>

<body>

    <h1>Click the button to perform calculations and get user input</h1>

    <button onclick="whenTheButtonIsClicked();">Click me!</button>

   <div id="output"></div>

</body>

</html>

PE05.ts

*// Upper and lower limit set*

const upperLimit: number = 21;

const lowerLimit: number = 0;

*// Initialization of userInput var*

let userInput:number;

*//Function to get user input*

function getUserInput(): number{

*// Use prompt to prompt the user for input, and the number function to*

*// convert the recived string to a numeber*

    userInput = Number(prompt('Please input a number between 1 and 20'));

*// call the validate user function to make sure the user input a valid number*

    validateUserInput(upperLimit, lowerLimit, userInput);

    return userInput;

}

*// function to validate user input. It checks to see if the users input is in between two*

*// pre-specified values*

function validateUserInput(upperLimit: number, lowerLimit: number, userInput: number){

    if(userInput > lowerLimit && userInput < upperLimit){

        return userInput;

    }else{

        console.log("The number is too high or too low");

        getUserInput();

    }

}

*// Task 2 function*

function someFunction(p1: number, p2: number):number{

    if(p1 <= 0){

        return p2;

    }else{

        p2 = p1 \* p2;

        p1--;

        return someFunction(p1,p2);

    }

}

*// recursive function to calculate the Fibonacci sequence at the user specified index*

function calcFibb(input:number){

    if(input === 1){

        return 0;

    }else if(input === 2){

        return 1;

    }else{

        return calcFibb(input-2)+calcFibb(input-1);

    }

}

*// funciton to call the get user function, calculate the other*

*// necessary numbers, and output HTML to the webpage*

function whenTheButtonIsClicked(){

    getUserInput();

    let task2FunctionAnswer = someFunction(userInput, userInput);

    let calcedFib = calcFibb(userInput);

    let outputHTML = document.getElementById('output');

    outputHTML.innerHTML = `<p>The user input: ${userInput}</p>

                            <p>Task 2 function answer: ${task2FunctionAnswer}</p>

                            <p>The value of the Fibonacci sequence at the ${userInput} position is ${calcedFib}</p>`;

}