**IS312 – Web Design and Programming**

**Student Name:** Trevor Sparks

**Quarter:** Fall 2022

**PE07 Lab Report**

Please complete this document per the lab instructions.

**Task #1 - 2 Source Code:**

interface Identity <T,U>{

    value:  T;

    message: U;

}

**Task #1 - 4 Source Code:**

const var1:Identity<number, string> = {value: 1, message: "Hello World"};

const var2:Identity<string, number> = {value: "Hello World", message: 5};

**Task #2 - 1 Source Code:**

interface ProcessIdentity{

    <T,U>(value: T, message: U): T

}

**Task #2 - 3 Source Code:**

function processIdentity<T,U>(value: T, message:U): T{

    console.log(message);

    return value;

}

**Task #2 – 7 Response to prompts:**

This allows us to pass variables to the variable processor which is of type processIdentity. TypeScript handles this by making things more strict. You must pass processor a number and then a string, or else TypeScript throughs an error during compilation.

**Task #2 – 9 Response to prompts:**

The results when running the code is an error. However, if you comment out the line with returnNumber2 the code returns 100. We get an error because of the reason I sated in the last question. We cannot pass a string and then a number to the processIdentity function (in this case is is the variable processor which is of type processIdentity).

**Task #3 – 2 Source Code:**

*// Task #3*

interface IProcessIdentity<T, U> {

    value: T;

    message: U;

    process() : T;

}

**Task #3 – 6 Response to prompts:**

We are able to do this because the T,U,X, and Y are just place holder values. What the letters are do not matter, it could be P and Q if you wanted it to be.

**Task #3 – 7 Source Code:**

*// Task #3*

interface IProcessIdentity<T, U> {

    value: T;

    message: U;

    process() : T;

}

class processIdentityClass<X,Y> implements IProcessIdentity<X, Y> {

    value: X;

    message: Y;

    process() : X {

        return this.value;

    }

    constructor(value: X, message: Y) {

        this.value = value;

        this.message = message;

    }

}

let processInstance : processIdentityClass<number, string> = new processIdentityClass(0, "hello")

let returnval = processInstance.process();

console.log(returnval)

processInstance.value = '100';

**Task #3 – 10 Response to prompts:**

In this case when the function is run I will receive an output of 0. This is because I set 0 as the value. Also, I get an error when I try to set processInstance.value to =’100’ because ‘100’ is a string and the code is expecting to get a number.

**Task #4 – 1 Source Code:**

*// Task #4*

class ProcessIdentityClass<X,Y> {

    private value: X;

    private message: Y;

    constructor(value: X, message: Y) {

        this.value = value;

        this.message = message;

    }

    getIdentity() : X {

        console.log(this.message);

        return this.value;

    }

}

let processorInstance1 : ProcessIdentityClass<number, string>;

processorInstance1 = new ProcessIdentityClass(0, "hello");

let output = processorInstance1.getIdentity();

console.log(output)

**Task #4 – 5 Response to prompts:**

The output for this code is:

hello

0

Once you have completed this document please submit per instructions.