The following exercises are related to the use of Typescript.

1. Configure your Development Environment: Install Visual Studio Code Install Node.js Install Git Source control

Solution:

```
Follow the installation wizard on all required software.
You will need to add git installation bin directory to your path
You will need to run this command once you have installed node
to get typeScript to work.
npm install typescript -g
```

2. Create a git repository for your answers to this problem sheet. Push the repository to GitHub. Make a commit and push it to GitHub after each exercise.

Solution:

```
mkdir answers-TypeScript
------------------------------------------------------------
cd answers-TypeScript
------------------------------------------------------------
echo > README.md
------------------------------------------------------------
git init
Initialized empty Git repository in /Users/martin/answers-TypeScript/.git/
------------------------------------------------------------
git add .
------------------------------------------------------------
git commit -m "Empty first commit."
[master (root-commit) 6583cf6] Empty first commit.
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 README.md
------------------------------------------------------------

git remote add origin https://github.com/mkenirons/WHATEVER.git
You will need to add the path to your GitHub Repository here
------------------------------------------------------------
git push -u origin master
Add you file to GitHub (remote repository)
------------------------------------------------------------
```

3. Copy the following code and compile it using tsc

```
function addition(value: string) {
    console.log("Value is: " + value);
}

let firstVal: number = 42;
let secondVal: number = 1;
let sumOfVals: string = (firstVal + secondVal).toLocaleString();

addition(sumOfVals);
```

Run the resulting JavaScript file using node.

Solution:

```
tsc duck.ts
------------------------------------------------------------

node duck.js
```

4. Create a typescript file which demonstrates the use of types. Basic types are described on the following web page:

`http://www.typescriptlang.org/docs/handbook/basic-types.html`

For each type described on this page, implement an example. You can do this by defining a set of variables, use let instead of var for declaration, and set the type for each of these variables. For each of the variables declared, assign a value to it, and output to the screen.

```
1   //boolean
2   let flag: boolean = true;
3   console.log("Value assigned to flag is: "+flag);
```

Solution:

```
//boolean
let flag: boolean = true;
console.log("Value assigned to flag is: "+flag);

//number
```

```
let num1: number = 6;
console.log("Value assigned to num1 is: "+num1);

//string
let color: string = "blue";
console.log("Value assigned to color is: "+color);

//array
let list: number[] = [1, 2, 3];
for(let i = 0; i < list.length;i++){
    console.log("Number "+(i+1)+" is: "+list[i]+".");
}

//tuple
let x: [string, number];
x = ["hello", 10];
console.log("Tuple example: "+x[0].substr(1));

//enum
enum Color {Red = 1, Green, Blue}
let colorName: string = Color[2];
let c: Color = Color.Green;
console.log("Enum: Value of colorName is: "+colorName);
console.log("Enum: Value of c is: "+c);
console.log("Enum: Name of c is: " + Color[c]);

//any
let notSure: any = 4;
notSure = "maybe a string instead";
console.log("Value is a string: "+notSure+".");
notSure = false;
console.log("Now value is a boolean: "+notSure+".");

//any array
let listany: any[] = [1, true, "free"];
for(let i = 0; i < listany.length;i++){
    console.log("Before: Entry "+(i+1)+" is: "+listany[i]+".");
}

listany[1] = 100;
for(let i = 0; i < listany.length;i++){
    console.log("After: Entry "+(i+1)+" is: "+listany[i]+".");
```

```
}
```

5. For this exercise, refer to the following web page:

   `http://www.typescriptlang.org/docs/handbook/functions.html`

   For this exercise, you are required to implement parameter types and return types for a function. Complete the following 3 exercises to demonstrate knowledge of this:

   (a) Create a function which accepts a string parameter, and returns a count of the number of characters in that string. For example, if the string provided as an input is "test 1" then the count returned is 6.

   (b) Create a function which accepts a string parameter, and returns a count of the number of characters in that string, excluding spaces. For example, if the string provided as an input is "test 1" then the count returned is 5.

   (c) Combine both function created in 1 and 2, into one function which accepts an optional parameter, so character count on input string can include or exclude spaces.

Solution:

```
function str_len(value: string): number{
    let num: number = value.length;
    return num;
}

function str_len_nospaces(value: string): number{
    let num: number = value.replace(/\s+/, "").length;
    return num;
}

console.log("String length with spaces and all is: "+str_len("test 1"));
console.log("String length with spaces not included in the count: "
+str_len_nospaces("test 1"));



function str_len_both(value: string, spaces?: boolean): number{
//note ? for optional parameter,
//so will default to false due to
//code in the function, but I could
//change from optional and provide
//a default to the parameter by writing
```

```
//the signature as
//function str_len_both(value: string, spaces: boolean=false): number{
    let num: number;
    if(spaces){
        num = value.replace(/\s+/, "").length;
    }else{
        num = value.length;
    }
    return num;
}

console.log("Function combined: String length with spaces and all is:
"+str_len_both("test 1",false));
console.log("Function combined: String length with spaces not included
in the count: "+str_len_both("test 1", true));
console.log("Function combined: String length with spaces and all is, not
setting spaces parameter so will default to false: "+str_len_both("test 1"));
```