COMP110 - Worksheet 00 Name: _____ PID: _____

1. (a) **F** The three primitive data types in TypeScript are `number`, `boolean`, and `character`.
   (b) **T** All functions must have a `return` type.
   (c) **T** Parameters are local variables within the function body's block meaning their names are only accessible inside the function.
   (d) **F** When the `+` operator is used with one or more numbers, the process is called concatenation.
   (e) **T** In order to use a function, there must be a call to that function elsewhere in the code.
   (f) **T** A function call is **always** complete as soon as any `return` statement is reached.
   (g) **F** Arguments are specified within the parentheses of a function definition.
   (h) **F** Variable names can contain spaces.

2. Fill in each column with the value of the variable after each line of code has run. If the variable has not been declared, write "?". If the variable has been declared but not initialized, write "*".

	x	y	z
let x: number = 15;	**15**	**?**	**?**
let y: number;	**15**	*	**?**
x = x + 8;	**23**	*	**?**
let z: number = 9;	**23**	*	**9**
y = x + z;	**23**	**32**	**9**

3. Given the incomplete functions to the left and the function bodies to the right, write the letter of the function for which each body could be used (with no errors). A body can work with 0, 1, or more function definitions.

```
let a = (x: number): number => {
    // ???
};
let b = (x: boolean): boolean => {
    // ???
};
let c = (x: boolean): number => {
    // ???
};
let d = (x: string): boolean => {
    // ???
};
let e = (x: number): string => {
    // ???
};
let f = (x: number): boolean => {
    // ???
};
let g = (x: string): string => {
    // ???
};
```

(a) **A, B, G** `return x;`

(b) **B, D, F** `return true;`

(c) **A** `return x * 2;`

(d) **F** `return x < 10;`

(e) **A, G** `return 7 + x;`

(f) **B** `return x && true;`

(g) **A, C** `return 5;`

(h) **E, G** `return x + "2";`

4. Write a function called `repeatBeat` that takes in a string `str` and a number `n` and returns the string repeated n times with a space between each repeat. You may assume `n >= 0`. There are some sample inputs and outputs below.
Note: To receive full credit, there can be no space at the end of the output.

`repeatBeat("whoop", 3);` `// returns "whoop whoop whoop"`	`repeatBeat("swoosh", 0);` `// returns ""`

```
//Write your code here

let repeatBeat = (s: string, n: number): string => {
     if (n === 0) {
          return "";
     }
     let output: string = s;
     while (n > 1) {
          output = output + " " + s;
          n = n - 1;
     }
     return output;
};
```

5. Write a function named `modulo` that takes in 2 number parameters, named `x` and `y`, and returns the result `x % y`. You may assume that both `x` and `y` are positive.
Note: In order to receive any credit, you may not use the modulo operator, and you must use a while loop.

```
let modulo =  (x: number, y: number): number => {
     while (x >= y) {
          x = x - y;
     }
     return x;
};
```

6. In the space provided below, write a function named `compare` that takes 2 number parameters, named `num1` and `num2`, that returns -1 if `num1` is less than `num2`, 0 if `num1` is equal to `num2`, and 1 if `num1` is greater than `num2`.

```
let compare = (num1: number, num2: number): number => {
      if (num1 < num2) {
            return -1;
      } else if (num1 === num2) {
            return 0;
      } else {
            return 1;
      }
};
```

7. Given the two functions on the left, answer the questions in the box on the right.

```
let isEven = (x: number): boolean => {
    return x % 2 === 0;
};

let getName = (y: number): string => {
    if (isEven(y)) {
        return "Michael";
    } else {
        if (y === 4) {
            return "Dwight";
        } else {
            if (y === 3) {
                return "Jim";
            }
            return "Stanley";
        }
    }
};
```

(a) What will be returned by following function call: `getName(3);`

Jim

(b) What will be returned by the following function call: `getName(4);`

Michael

(c) If there is any unreachable code (i.e. code that can never be reached in any circumstance), circle it. If there is not, write "NA" below.

(highlighted)

(d) Make a function call that will result in the value `"Stanley"` being returned.

`getName(7);`