

1. (a) _____ The three primitive data types in TypeScript are **number**, **boolean**, and **character**.
- (b) _____ All functions must have a **return** type.
- (c) _____ Parameters are local variables within the function body's block meaning their names are only accessible inside the function.
- (d) _____ When the **+** operator is used with one or more numbers, the process is called concatenation.
- (e) _____ In order to use a function, there must be a call to that function elsewhere in the code.
- (f) _____ A function call is **always** complete as soon as any **return** statement is reached.
- (g) _____ Arguments are specified within the parentheses of a function definition.
- (h) _____ 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;			
let y: number;			
x = x + 8;			
let z: number = 9;			
y = x + z;			

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.

<pre> 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 => { // ??? }; </pre>	<p>(a) _____ return x;</p> <p>(b) _____ return true;</p> <p>(c) _____ return x * 2;</p> <p>(d) _____ return x < 10;</p> <p>(e) _____ return 7 + x;</p> <p>(f) _____ return x && true;</p> <p>(g) _____ return 5;</p> <p>(h) _____ return x + "2";</p>
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

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.

<pre>repeatBeat("whoop", 3); // returns "whoop whoop whoop"</pre>	<pre>repeatBeat("swoosh", 0); // returns ""</pre>
<pre>//Write your code here</pre>	

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.

```
//Write your code here
```

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`.

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)`;
- (b) What will be returned by the following function call: `getName(4)`;
- (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.
- (d) Make a function call that will result in the value "Stanley" being returned.