

1. Write the boolean value each expression evaluates to:

1.1 `true || false`

1.2 `-1 > -8`

1.3 `false && !true`

1.4 `"happy" === "joyful"`

1.5 `(4 * 5) === (40 / 2)`

1.6 `false || false`

1.7 `12 <= (2 * 8 - 4)`

1.8 `1 === -1`

1.9 `false || (4 > 2 * 1 / 3)`

1.10 `(true && true) && false`

1.11 `(!false || true) !== true`

1.12 `!((22 + 2 / 6) === 50)`

1.13 `(!true && false) === false`

1.14 `!(5 > 2 || false) !== false`

1.15 `"COMP" === "COMP"`

1.16 `(false === false) || false`

2. With the following code snippet, what output will appear on the screen when the following values are used for x and y?

```
1 let x: number;
2 let y: number;
3 //see question for actual values of x and y
4
5 let z: number = 0;
6 x = x + 1;
7 if (x < y) {
8     z = x ** y / 2;
9 } else {
10    z = y % x;
11 }
12 z = z + 1;
13 print(z);
```

2.1 When x = 3, y = 4?

2.2 When x = 3, y = 6?

2.3 When x = 7, y = 2?

2.4 When x = -5, y = 1?

2.5 On what line can the exponentiation operator be found?

2.6 What is the arithmetic operator on line 10 called?

3. With the following code snippet, for questions 3.1 - 3.4 what output will appear on the screen when the following values are used for a and b?

```
1 let a: number;
2 let b: number;
3 //see question for actual values of a and b
4
5 let z = 0;
6 if (a + b > 15) {
7     z = 2;
8 } else {
9     if (a < b) {
10        z = a - b;
11    } else {
12        z = a ** 2 + b;
13    }
14 }
15 print(z);
```

3.1 When a = 3, b = 7?

3.2 When a = 8, b = 9?

3.3 When a = 5, b = 3?

3.4 When a = 3, b = -1?

3.5 What is variable z's type?

3.6 On line 1, is a being declared, initialized, or both?

3.7 On line 5, is z being declared, initialized, or both?

4. In the space below, write code that meets the specifications. Declare two variables, named `foo` and `bar`, both of type `number`. If `foo` is an even number and is greater than `bar`, print the string "fizzbang". If `foo` is not even, but is greater than `bar`, print "boom". If `bar` is bigger than `foo`, print "pow". Hint: how can we use the remainder operator (%) to figure out if a number is even?

5. For each of the following code snippets, write how many statements will be printed and what the final value of `i` is. If there are an infinite number of statements printed, write `infinity`. Assume `print` has already been imported.

```
1 let i = 0;
2 while (i < 10) {
3   print(i);
4   i = i + 1;
5 }
```

5.1

```
1 let i = 0;
2 while (i > 10) {
3   print(i);
4   i = i + 1;
5 }
```

5.2

```
1 let i = 0;
2 while (i < 10) {
3   print(i);
4   i = i - 1;
5 }
```

5.3

```
1 let i = 5;
2 while (i >= 0) {
3   print(i);
4   i = i - 1;
5 }
```

5.4

6. For each part below, there are instructions on how many times a loop should iterate. For each code listing, fill in the blank(s) so that the loop will repeat that many times.

6.1 The loop should iterate 6 times.

```
1 let i = _____;
2
3 while ( _____ ) {
4     print("Yeehaw!");
5     i = i + 1;
6 }
```

6.2 The loop should iterate 4 times.

```
1 let i = 10;
2
3 while ( _____ ) {
4     print("Yeehaw!");
5     i = i - 1;
6 }
```

6.3 The loop should iterate 3 times.

```
1 let i = 1;
2
3 while ( i < 8 ) {
4     print("Yeehaw!");
5     i = _____;
6 }
```

1. Write the boolean value each expression evaluates to:

1.1 true || false

1.2 -1 > -8

1.3 false && !true

1.4 "happy" === "joyful"

1.5 (4 * 5) === (40 / 2)

1.6 false || false

1.7 12 <= (2 * 8 - 4)

1.8 1 === -1

1.9 false || (4 > 2 * 1 / 3)

1.10 (true && true) && false

1.11 (!false || true) !== true

1.12 !((22 + 2 / 6) === 50)

1.13 (!true && false) === false

1.14 !(5 > 2 || false) !== false

1.15 "COMP" === "COMP"

1.16 (false === false) || false

2. With the following code snippet, what output will appear on the screen when the following values are used for x and y?

```
1 let x: number;
2 let y: number;
3 //see question for actual values of x and y
4
5 let z: number = 0;
6 x = x + 1;
7 if (x < y) {
8     z = x ** y / 2;
9 } else {
10    z = y % x;
11 }
12 z = z + 1;
13 print(z);
```

- 2.1 When x = 3, y = 4? **1**
- 2.2 When x = 3, y = 6? **2049**
- 2.3 When x = 7, y = 2? **3**
- 2.4 When x = -5, y = 1? **-1**
- 2.5 On what line can the exponentiation operator be found? **line 8**
- 2.6 What is the arithmetic operator on line 10 called? **remainder operator**

3. With the following code snippet, for questions 3.1 - 3.4 what output will appear on the screen when the following values are used for a and b?

```
1 let a: number;
2 let b: number;
3 //see question for actual values of a and b
4
5 let z = 0;
6 if (a + b > 15) {
7     z = 2;
8 } else {
9     if (a < b) {
10        z = a - b;
11    } else {
12        z = a ** 2 + b;
13    }
14 }
15 print(z);
```

- 3.1 When a = 3, b = 7? **-4**
- 3.2 When a = 8, b = 9? **2**
- 3.3 When a = 5, b = 3? **28**
- 3.4 When a = 3, b = -1? **8**
- 3.5 What is variable z's type? **number**
- 3.6 On line 1, is a being declared, initialized, or both? **declared**
- 3.7 On line 5, is z being declared, initialized, or both? **both**

4. In the space below, write code that meets the specifications. Declare two variables, named `foo` and `bar`, both of type `number`. If `foo` is an even number and is greater than `bar`, print the string "fizzbang". If `foo` is not even, but is greater than `bar`, print "boom". If `bar` is bigger than `foo`, print "pow". Hint: how can we use the remainder operator (%) to figure out if a number is even?

```
let foo: number;
let bar: number;

// there are many possible solutions
if (foo > bar) {
  if (foo % 2 === 0) {
    print("fizzbang");
  } else {
    print("boom");
  }
} else {
  print("pow");
}
```

5. For each of the following code snippets, write how many statements will be printed and what the final value of `i` is. If there are an infinite number of statements printed, write `infinity`. Assume `print` has already been imported.

```
1 let i = 0;
2 while (i < 10) {
3   print(i);
4   i = i + 1;
5 }
```

5.1

10; i = 10

```
1 let i = 0;
2 while (i > 10) {
3   print(i);
4   i = i + 1;
5 }
```

5.2

0; i = 0

```
1 let i = 0;
2 while (i < 10) {
3   print(i);
4   i = i - 1;
5 }
```

5.3

infinity

```
1 let i = 5;
2 while (i >= 0) {
3   print(i);
4   i = i - 1;
5 }
```

5.4

6; i = -1

6. For each part below, there are instructions on how many times a loop should iterate. For each code listing, fill in the blank(s) so that the loop will repeat that many times.

6.1 The loop should iterate 6 times.

```
1 let i = _____;
2
3 while ( _____ ) {
4     print("Yeehaw!");
5     i = i + 1;
6 }
```

i=0, i < 6

6.2 The loop should iterate 4 times.

```
1 let i = 10;
2
3 while ( _____ ) {
4     print("Yeehaw!");
5     i = i - 1;
6 }
```

i > 6

6.3 The loop should iterate 3 times.

```
1 let i = 1;
2
3 while ( i < 8 ) {
4     print("Yeehaw!");
5     i = _____;
6 }
```

i + 3