1. Which of the following best describes what a function in JavaScript is used for?

A function stores data.

A function creates new variables.

A function is a reusable piece of code that can accept input and performs a specific task.

A function allows for the use of mathematical operators.

1. What will this code print to the console?

const sleepTimer = (alarm) => {  
  console.log('My alarm is set for: ' + alarm);  
}  
  
sleepTimer('8:30AM');

My alarm is set for: undefined

My alarm is set for: sleepTimer

My alarm is set for: alarm

My alarm is set for: 8:30AM

1. Which of the following is the correct way to call the function below?

const multiplier = (number) => {  
  console.log(3 \* number);  
};

**multiplier 5**

**multiplier[5]**

**multiplier{5}**

**multiplier(5)**

1. Which of the following is a parameter in the block of code below?

let input = 8;  
const controlVal = input / 2 + 3;  
const multiplier = (number, phase) => {  
  const val = number \* controlVal + phase;  
  console.log(val);  
};

**val**

**controlVal**

**number**

**input**

1. This code prints: **I ran 3 miles at an average of undefined per mile**. Why does the text include **undefined**?

const workoutJournal = (miles, avgTime) => {  
  console.log('I ran ' + miles + ' miles at an average of ' + avgTime + ' per mile.');  
};  
  
workoutJournal('3');

The call to **workoutJournal** is missing a second argument.

**workoutJournal** does not print the value of both arguments.

The **miles** parameter is not declared.

**workoutJournal** is not defined.

1. What is wrong with the code snippet provided?

const greeting = => {  
  console.log('Hello Programmer!');  
};

The **greeting** function is missing a set of **()** between the **=** and **=>**.

The function expression cannot be declared with a **const** keyword.

The curly braces **{}** should be parentheses **()**.

The ordering of **=** and **=>** should be switched.

1. What will be printed to the console?

const eatFruit = (fruit = 'apple') => {  
  console.log(`This ${fruit} is delicious!`);  
};  
  
eatFruit();

This undefined is delicious!

This apple is delicious!

This fruit is delicious!

This is delicious!

1. Which correctly represents the most condensed form of the function? Recall that this syntax is also known as ‘concise body.’

const areaOfCircle = radius => Math.PI \* radius \* radius;

const areaOfCircle = radius => { Math.PI \* radius \* radius };

const areaOfCircle = radius => { return Math.PI \* radius \* radius };

const areaOfCircle = radius => return Math.PI \* radius \* radius;

1. Which of the following most accurately describes the **volumeOfCube()** function?

function volumeOfCube (side) {  
  return side \* side \* side;  
}  
  
volumeOfCube(5);  
// Output: 125.

It is an arrow function.

It is a function declaration.

It is a function expression.

It is an anonymous function.

1. What’s the purpose of a parameter?

To allow a function to accept data.

To specify actual values passed to a function.

To call a function.