Section 2: Variables: what you need

1. Which is wrong?

   A. `let student_grades;`
   B. `var $2age;`
   C. `const numberOfStars = 5;`
   D. `let 2_name = "Hero"`

Ans: D - Wrong because variable names don't start with numbers

2. What is wrong?

   A. let x = 4

      x = 7

   B. const x;

      x = 7

   C. var x = 5;

      var x = 9

   D. var x, y, z;

Ans: B - this is wrong because const variables always need to be initialized

3. var and let can be declared and must be initialized

   A. True
   B. False

Ans: B - only const variables must be initialized

4. what's the value for y?

```
let x = 8

let y = x++
```

A. 8
B. 9

Ans A: right because the value of x will be assigned to y before incrementing x

5. What would this give?

```
const x = 5
const z = --x
console.log(z)
```

A. 4
B. Error
C. 5
D. 6

Ans: B - it'll give a type error “assignment to constant variable” since --x means x = x - 1 and x is const

Section 4: Functions: what you need

1. All these are uses of a function except?

A. Makes our code readable and organized
B. Helps us in code reuse
C. We can use it to bring down the walls of America
D. With functions, we can easily maintain code

Ans: C - this is very wrong, you know that 😂. Just for no reason it's wrong 😂

2. What does this produce?

```
function go(){

    return false
    console.log(true)
}
console.log(go())
```

A. True
B. false true
C. false

Ans: C - the code after the return statement doesn't work

3. What does this log?

```
function go(){

    console.log("Desmond is")
    return "a superhero"
    console.log("He's also a good tutor")
}
console.log(go())
```

A. Desmond is

He's also a good tutor

B. Desmond is

a superhero

He's also a good tutor

C. Desmond is

a superhero

Ans: C - the last console log after the return statement doesn't work

4. How do you call ...z?

```
const max = (...z) => Math.max(...z)
console.log(max(11,22,3,4))
```

A. Rest params
B. Rest Parameters
C. None of the Parameters

Ans: B -

5. Produces a random number from what to what?

```
const random = () => Math.floor(Math.random() * 6) + 5
console.log(random())
```

A. from 5 to 11
B. from 5 to 10
C. from 5 to 6
D. from 1 to 5

Ans: B

Section 5: Objects: what you need

1. How do I access the city of this person object?

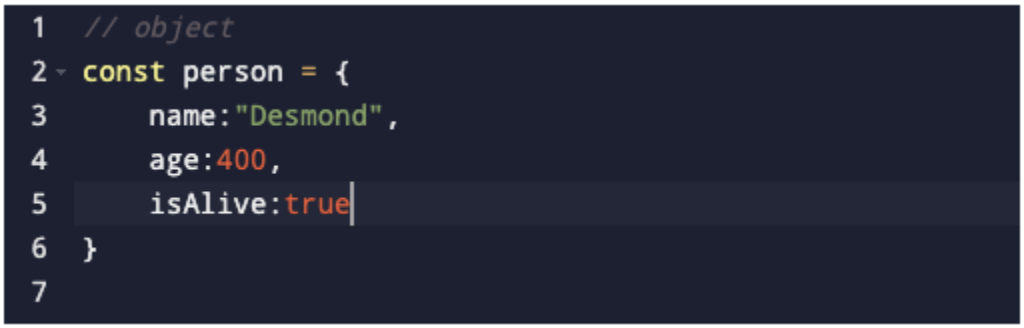
```
// object
const person = {
    name:"Desmond",
    location:{
        country:{
            city:"My city",
            code:334512
        }
    }
}

```

A. person.location.city
B. person.city
C. person.location.country.city
D. person["location"]["city"]

Ans: C - We're getting the city inside the country inside the location and still inside the person object

2. How do I get the values of this object in array form?



```
// object
const person = {
    name:"Desmond",
    age:400,
    isAlive:true
}

```

A. object.keys(person)
B. Object.values(object)

C. person.values()

D. Object.values(person)

Ans: D

3. What does this log on screen?

```
1  // object
2  const person = {
3      name:"Desmond",
4      age:400,
5      isAlive:true
6  }
7
8  for(key in person){
9      console.log(key)
10 }
11
```

A. `name, age and isAlive`

B. `Desmond, 400 and true`

C. `name, 400 and isAlive`

D. Error

Ans: A

Section 6: Arrays: what you need

1. What would this produce?

```
const colors = ["red","blue","orange"]
console.log(colors[3])
```

A. orange
B. undefined
C. blue
D. red

Ans: B - Undefined because the position doesn't exist on the array, Array ends at position 2(orange)

2. const colors = ["red","green","blue]
how do we pick out green using array Destructuring?

A. `const [,greenColor] = colors`
B. `const [green] = colors`
C. `const green from colors`

Ans: A - We don't want the first one so we put a comma, then pick the second one

3. let numbers = [4,5,3,2,8,9,0]
how do you pick out all numbers less than or equal to 3? ([3,2,0])

A. `const lessThanOrEqualThree = numbers.filter(n => n < 3)`
B. `const lessThanOrEqualThree = numbers.filter(n => n <= 3)`
C. `const lessThanOrEqualThree = numbers.find(n => n <= 3)`
D. `const lessThanOrEqualThree = numbers.map(n => n <= 3)`

Ans: B - Filter returns an array of elements that pass certain conditions

4. const names = ["Desmond","Oben","Superhero"]
How do we know if this variable is an array?

A. `typeof names`
B. `names.isArray()`
C. `Array.isArray(names)`
D. `typeof names[]`

Ans: C

5. const prices = [45,90,20]
How do we add a $5.4 charge on every price to produce a new array like [50.4, 95.4, 25.4]

A. const newArray = prices.map(p => p + 5.4)
console.log(newArray)
B. const newArray = prices.filter(p => p + 5.4)
console.log(newArray)
C. const newArray = prices.find(p => p + 5.4)
console.log(newArray)
D. const newArray = [50.4, 95.4, 25.4]

Ans: A - The map produces a new array which can be modified