

Interviewee: Lim Zhi Cheng

Github Repo: [Click here to the repo \(all the code can be found in this repo\)](#)

## Question 1:

Use **Java**, do this coding test & print out the results in terminal:

- Variable "arrayA" is a string array of car brands: ["Honda", "Toyota", "Proton"]
- Add "Nissan" to it. Print it out.
- Add "Mazda" as the first element within the string array. Print it out.
- Now make a new variable "arrayB" which is a string array of animals. Combine both arrays into variable "arrayA". Print it out.
- Variable "mapC" & "mapD" are both separate Map[String, String] objects of employee IDs and employee names, where:
  - mapC → ["123": "Justin", "456": "Owen", "789": "Hugo"]
  - mapD → ["123": "George", "555": "Jack", "888": "Julian"]
- Combine them both into the variable "mapC" and print it out.
- Print out the reason why the result no longer has the value "Justin"?
- Return values in "mapC" where key contains "5"
- Remove map entries where the value contains the "o" character in "mapC". (ignore case)
- I have a string variable "str" of "111222888222555". Reverse this string and assign it back to variable "str". Print it out.
- Then, replace the last occurrence of "222" in variable "str" with "aaa" and assign it back to variable "str". Print it out.

Code:

```
import java.util.*;

public class Main {
    public static void main(String[] args) {
        // Step 1: Initialize arrayA
        List<String> arrayA = new ArrayList<>(Arrays.asList("Honda",
"Toyota", "Proton"));

        // Step 2: Add "Nissan" to arrayA and print it
        arrayA.add("Nissan");
        System.out.println("After adding 'Nissan': " + arrayA);

        // Step 3: Add "Mazda" as the first element and print it
        arrayA.add(0, "Mazda");
        System.out.println("After adding 'Mazda' as the first element: ")
```

```

+ arrayA);

// Step 4: Create arrayB and combine with arrayA
List<String> arrayB = Arrays.asList("Cat", "Dog", "Elephant");
arrayA.addAll(arrayB);
System.out.println("Combined arrayA: " + arrayA);

// Step 5: Initialize mapC and mapD
Map<String, String> mapC = new HashMap<>();
mapC.put("123", "Justin");
mapC.put("456", "Owen");
mapC.put("789", "Hugo");

Map<String, String> mapD = new HashMap<>();
mapD.put("123", "George");
mapD.put("555", "Jack");
mapD.put("888", "Julian");

// Step 6: Combine mapD into mapC and print it
mapC.putAll(mapD);
System.out.println("Combined mapC: " + mapC);

// Step 7: Explain why "Justin" is no longer in mapC
System.out.println("Reason: The key '123' in mapC was
overwritten by the value from mapD.");

// Step 8: Return values in mapC where the key contains "5"
System.out.println("Values where key contains '5':");
mapC.forEach((key, value) -> {
    if (key.contains("5")) {
        System.out.println("Key: " + key + ", Value: " + value);
    }
});

// Step 9: Remove entries where the value contains "o" (ignore
case)
mapC.entrySet().removeIf(entry ->
entry.getValue().toLowerCase().contains("o"));

// Step 10: Reverse the string and assign it back to str
String str = "111222888222555";
String reversedStr = new
StringBuilder(str).reverse().toString();
System.out.println("Reversed string: " + reversedStr);

// Step 11: Replace the last occurrence of "222" with "aaa"

```

```

        final String TARGET_SUBSTRING = "222";          // The substring to
find and replace
        final String REPLACEMENT_SUBSTRING = "aaa"; // The substring to
replace with
        final int SUBSTRING_LENGTH = TARGET_SUBSTRING.length(); //
Length of the target substring

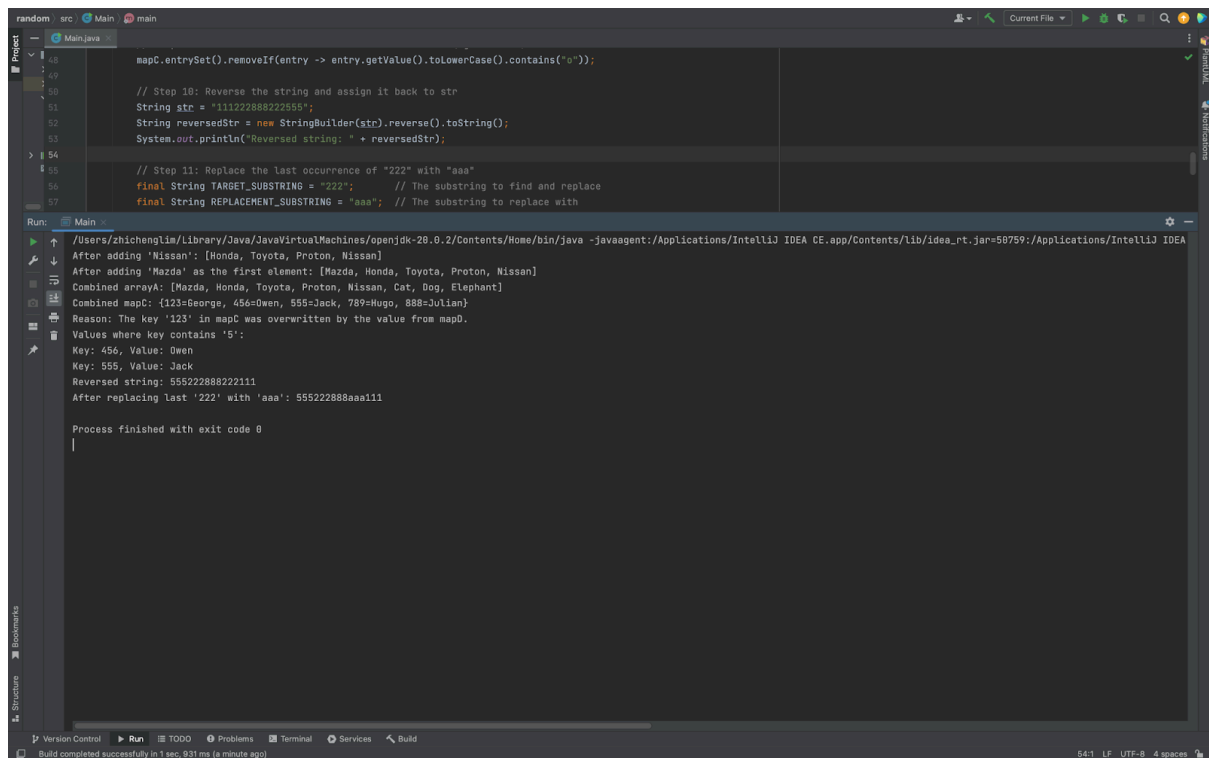
        int lastIndex = reversedStr.lastIndexOf(TARGET_SUBSTRING);

        // Check if "222" exists in the string
        if (lastIndex != -1) {
            str = reversedStr.substring(0, lastIndex)
                + REPLACEMENT_SUBSTRING
                + reversedStr.substring(lastIndex +
SUBSTRING_LENGTH);
        }

        System.out.println("After replacing last '" + TARGET_SUBSTRING +
"" with '" + REPLACEMENT_SUBSTRING + "': " + str);
    }
}

```

**Results in terminal:**



The screenshot shows an IDE with a Java file named 'Main.java'. The code includes a map, string reversal, and string replacement. The 'Run' window shows the following output:

```

/Users/zhichenglin/Library/Java/JavaVirtualMachines/openjdk-28.0.2/Contents/Home/bin/java -javaagent:/Applications/IntelliJ IDEA CE.app/Contents/lib/idea_rt.jar=50759:/Applications/IntelliJ IDEA
After adding 'Nissan': [Honda, Toyota, Proton, Nissan]
After adding 'Mazda' as the first element: [Mazda, Honda, Toyota, Proton, Nissan]
Combined arrayA: [Mazda, Honda, Toyota, Proton, Nissan, Cat, Dog, Elephant]
Combined mapC: {123=George, 456=Owen, 555=Jack, 789=Hugo, 888=Julian}
Reason: The key '123' in mapC was overwritten by the value from mapD.
Values where key contains '5':
Key: 456, Value: Owen
Key: 555, Value: Jack
Reversed string: 55522888222111
After replacing last '222' with 'aaa': 55522888aaa111

Process finished with exit code 0

```

## Question 2:

Using **Javascript**. Create an array of 10 random integer numbers between 1 - 1000 and use `console.log()` to display each of the numbers in the array and the sum of the array.

**Code:**

```
const randomNumbers = Array.from({ length: 10 }, () =>
Math.floor(Math.random() * 1000) + 1);

randomNumbers.forEach((num, index) => console.log(`Number ${index + 1}:
${num}`));

let sum = 0;
for (let i = 0; i < randomNumbers.length; i++) {
    sum += randomNumbers[i];
}

console.log("Sum of the array:", sum);
```

## Results in console:

```
> const randomNumbers = Array.from({ length: 10 }, () => Math.floor(Math.random() * 1000) + 1);
randomNumbers.forEach((num, index) => console.log('Number ' + (index + 1) + ': ' + num));

let sum = 0;
for (let i = 0; i < randomNumbers.length; i++) {
  sum += randomNumbers[i];
}

console.log("Sum of the array:", sum);
Number 1: 191
Number 2: 43
Number 3: 380
Number 4: 796
Number 5: 580
Number 6: 752
Number 7: 73
Number 8: 5
Number 9: 947
Number 10: 701
Sum of the array: 4300
undefined
>
```

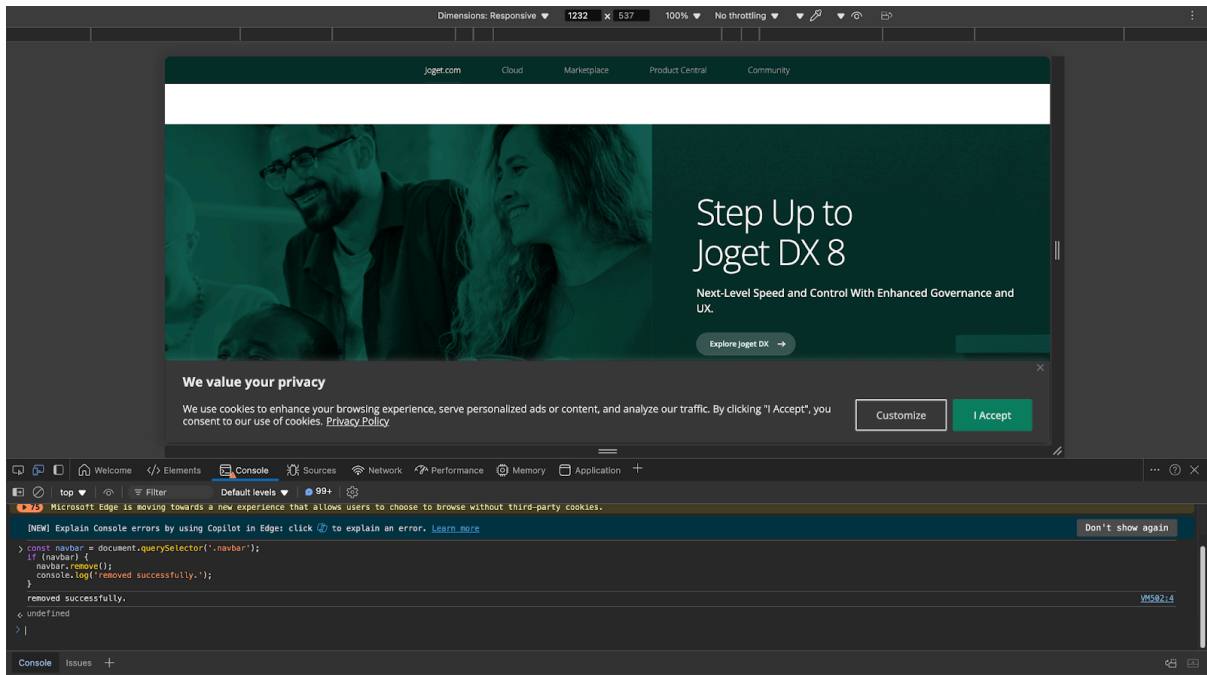
## Question 3:

Go to <https://www.joget.com/>, use **Javascript** to make the nav bar disappear. Send us the code.

## Code:

```
const navbar = document.querySelector('.navbar');
if (navbar) {
  navbar.remove();
  console.log('removed successfully.');
}
```

## Screenshot:



#### Question 4:

Using **HTML** and **CSS**, create a simple webpage layout with a header, footer, and main content section. The header should have a background color of blue and contain a centered Joget logo and a navigation menu with links to Home, About, and Contact pages. The main content section should have a white background and contain a heading and a paragraph of text. The footer should have a gray background and contain copyright information and a link to a privacy policy page.

Link to the Web page: [click here](#)

Github Repo (code can be found here): [click here](#)

**HTML screenshot:**

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Simple Webpage</title>
7   <link rel="stylesheet" href="style.css">
8 </head>
9 <body>
10  <header>
11    <div class="logo">
12      
13    </div>
14    <nav class="nav-menu">
15      <ul>
16        <li><a href="#">Home</a></li>
17        <li><a href="#">About</a></li>
18        <li><a href="#">Contact</a></li>
19      </ul>
20    </nav>
21  </header>
22
23  <main>
24    <h1>Welcome to Our Website</h1>
25    <p>This is a simple webpage layout with a header, footer, and main content section.</p>
26  </main>
27
28  <footer>
29    <p>©copy; 2024 Joget Inc. | <a href="#">Privacy Policy</a></p>
30  </footer>
31 </body>
32 </html>
33
```

**CSS screenshot:**

```
1
2  √ * {
3    margin: 0;
4    padding: 0;
5    box-sizing: border-box;
6  }
7
8  /* Flexbox layout for body */
9  √ body {
10    display: flex;
11    flex-direction: column;
12    min-height: 100vh;
13  }
14
15  /* Header styles */
16  √ header {
17    background-color: blue;
18    color: white;
19    padding: 10px 20px;
20    display: flex;
21    justify-content: space-between;
22    align-items: center;
23  }
24
25  √ header .logo img {
26    height: 50px;
27  }
28
29  √ header .nav-menu ul {
30    list-style: none;
31    display: flex;
32    gap: 15px;
33  }
34
35  √ header .nav-menu ul li a {
36    color: white;
37    text-decoration: none;
38    font-weight: bold;
39  }
40
41  /* Main content */
42  √ main {
43    background-color: white;
44    padding: 20px;
45    text-align: center;
46  }
47
```

```
48  /* Footer styles */
49  √ footer {
50    background-color: gray;
51    color: white;
52    text-align: center;
53    padding: 10px 0;
54    margin-top: auto;
55  }
56
57  √ footer a {
58    color: white;
59    text-decoration: underline;
60  }
61 |
```



Web Page Screenshot:

