JOBSHEET 10

Array 2

Name
Sherly Lutfi Azkiah Sulistyawati

NIM
2341720241

Class
1I

Department
Information Technology

Study Program
D4 Informatics Engineering

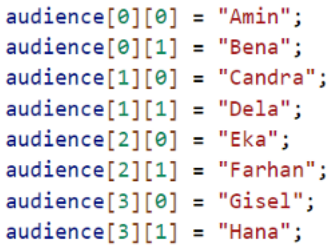
Labs Activity

Question! (Experiment 1)

1. Do array elements have to be filled in sequentially starting from the 0th index? Please explain!

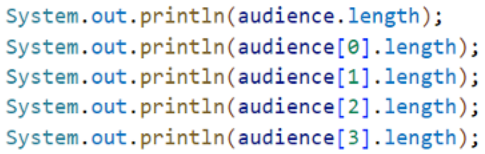
2. Why is there a null in the list of audience names?

3. Complete the audience list in step 4 so that it looks like the following program code



```
audience[0][0] = "Amin";
audience[0][1] = "Bena";
audience[1][0] = "Candra";
audience[1][1] = "Dela";
audience[2][0] = "Eka";
audience[2][1] = "Farhan";
audience[3][0] = "Gisel";
audience[3][1] = "Hana";
```

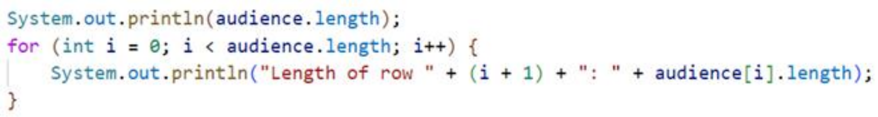
4. Add the following program code:



```
System.out.println(audience.length);
System.out.println(audience[0].length);
System.out.println(audience[1].length);
System.out.println(audience[2].length);
System.out.println(audience[3].length);
```

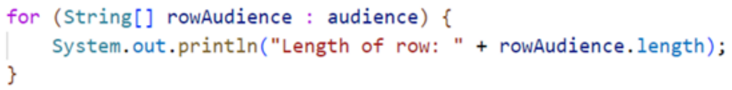
Explain the function of audience.length and audience[0].length! Do audience[0].length, audience[1].length, audience[2].length, and audience[3].length have the same value? Why?

5. Modify the program code in step 4 to display the length of each row in the array using a for loop. Compile, run, then commit.



```
System.out.println(audience.length);
for (int i = 0; i < audience.length; i++) {
    System.out.println("Length of row " + (i + 1) + ": " + audience[i].length);
}
```

6. Modify the program code in step 5 to display the length of each row in the array using a foreach loop. Compile, run, then commit.



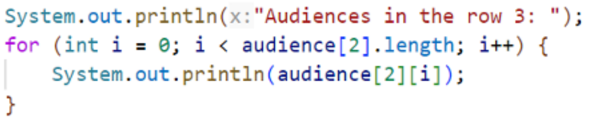
```
for (String[] rowAudience : audience) {
    System.out.println("Length of row: " + rowAudience.length);
}
```

7. In your opinion, what are the advantages and disadvantages of foreach loop compared to for loop?

8. What is the max row index for the audience array?

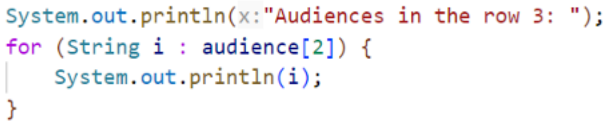
9. What is the max column index for the audience array?

10. Add program code to display the audience's name on the 3rd line using a for loop. Compile, run, then commit.



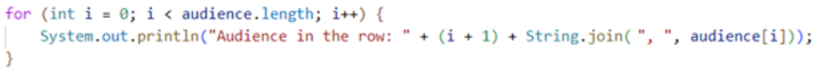
```
System.out.println(x:"Audiences in the row 3: ");
for (int i = 0; i < audience[2].length; i++) {
    System.out.println(audience[2][i]);
}
```

11. Modify the code in question number 10 to repeat using a foreach loop. Compile, run, then commit.



```
System.out.println(x:"Audiences in the row 3: ");
for (String i : audience[2]) {
    System.out.println(i);
}
```

12. Modify the program code in question number 11 again to display the audience's name for each line. Compile and run the program then observe the results, then commit.



```
for (int i = 0; i < audience.length; i++) {
    System.out.println("Audience in the row: " + (i + 1) + String.join(", ", audience[i]));
}
```

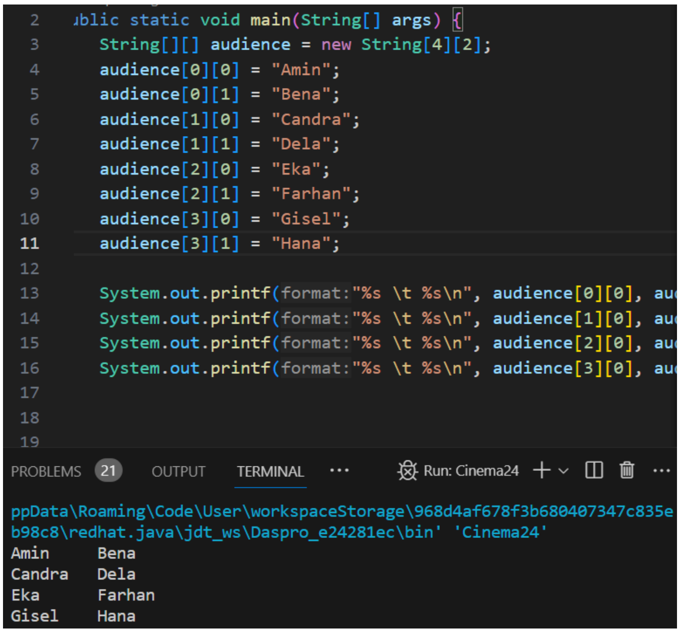
13. What is the function of String.join()?

14. Commit and push to GitHub

Answer!

1. No, array elements do not have to be filled in sequentially starting from the 0th index. In Java, as in many programming languages, arrays are zero-indexed, meaning the first element is at index 0. You can assign values to array elements in any order.

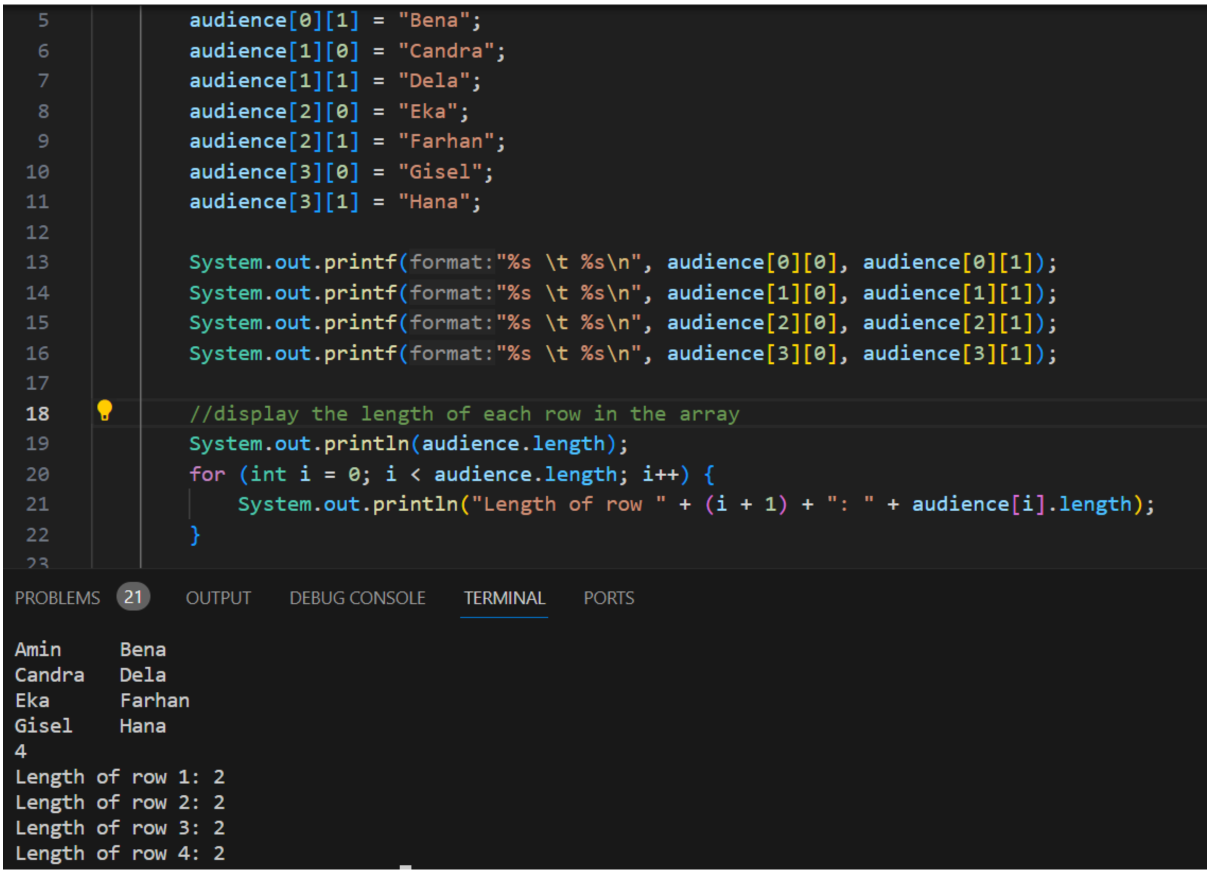
2. Because not all the elements of the array have been initialized with values.

3. 

```
ıblic static void main(String[] args) {
    String[][] audience = new String[4][2];
    audience[0][0] = "Amin";
    audience[0][1] = "Bena";
    audience[1][0] = "Candra";
    audience[1][1] = "Dela";
    audience[2][0] = "Eka";
    audience[2][1] = "Farhan";
    audience[3][0] = "Gisel";
    audience[3][1] = "Hana";

    System.out.printf(format:"%s \t %s\n", audience[0][0], au
    System.out.printf(format:"%s \t %s\n", audience[1][0], au
    System.out.printf(format:"%s \t %s\n", audience[2][0], au
    System.out.printf(format:"%s \t %s\n", audience[3][0], au
```

```
ppData\Roaming\Code\User\workspaceStorage\968d4af678f3b680407347c835e
b98c8\redhat.java\jdt_ws\Daspro_e24281ec\bin' 'Cinema24'
Amin     Bena
Candra   Dela
Eka      Farhan
Gisel    Hana
```

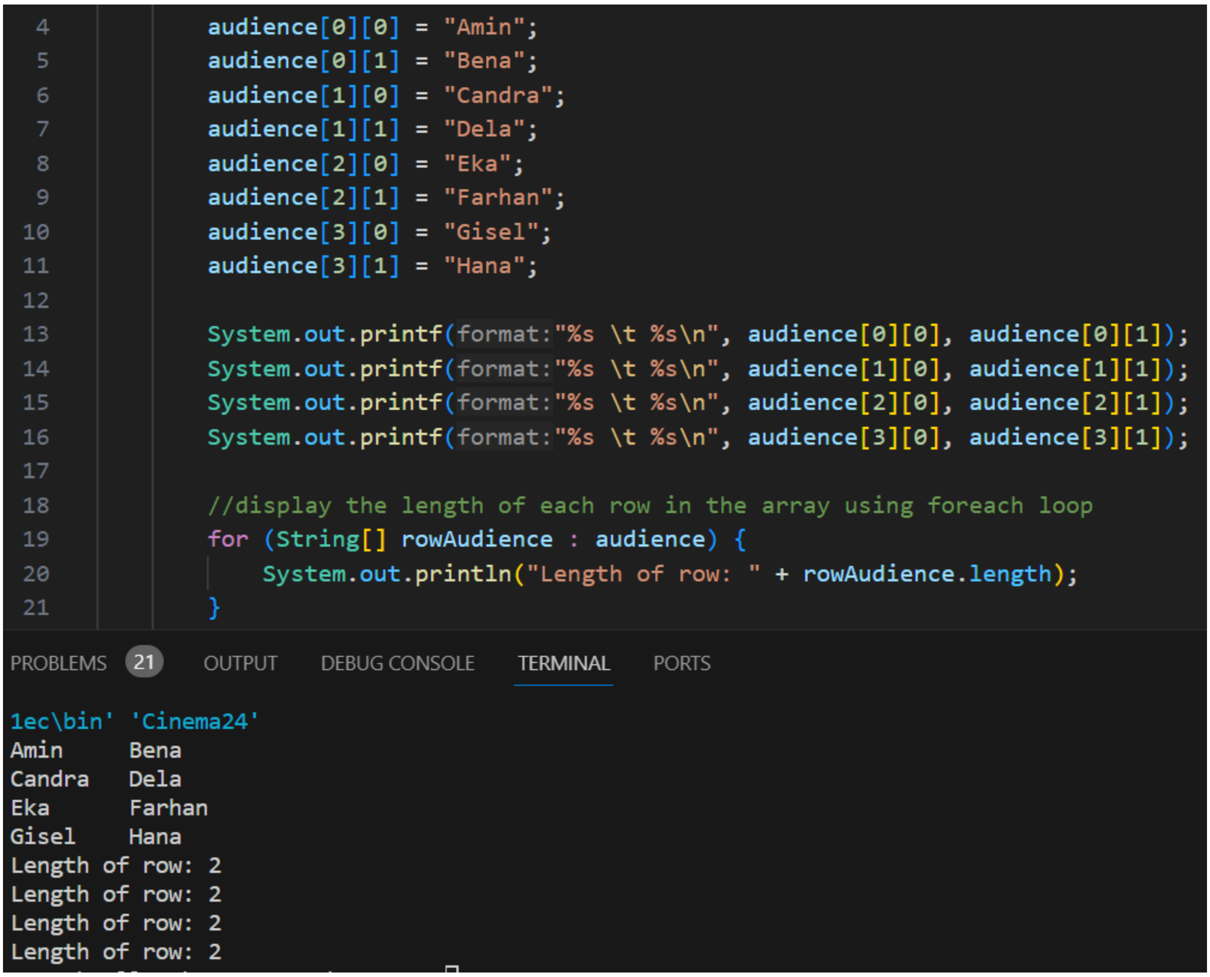
4. audience.length gives the number of rows in the audience array, and audience[i].length gives the number of columns in the i-th row of the array. audience.length will output the number of rows in the audience array. In this case, it will be 4 because we've created a 4x2 array. Whereas, audience[0].length, audience[1].length, audience[2].length, and audience[3].length all have the same value (2) because we've defined the array to have 2 columns for each row.



```java
        audience[0][1] = "Bena";
        audience[1][0] = "Candra";
        audience[1][1] = "Dela";
        audience[2][0] = "Eka";
        audience[2][1] = "Farhan";
        audience[3][0] = "Gisel";
        audience[3][1] = "Hana";

        System.out.printf(format:"%s \t %s\n", audience[0][0], audience[0][1]);
        System.out.printf(format:"%s \t %s\n", audience[1][0], audience[1][1]);
        System.out.printf(format:"%s \t %s\n", audience[2][0], audience[2][1]);
        System.out.printf(format:"%s \t %s\n", audience[3][0], audience[3][1]);

        //display the length of each row in the array
        System.out.println(audience.length);
        for (int i = 0; i < audience.length; i++) {
            System.out.println("Length of row " + (i + 1) + ": " + audience[i].length);
        }
```

```
Amin     Bena
Candra   Dela
Eka      Farhan
Gisel    Hana
4
Length of row 1: 2
Length of row 2: 2
Length of row 3: 2
Length of row 4: 2
```

5.

6. 

```
        audience[0][0] = "Amin";
        audience[0][1] = "Bena";
        audience[1][0] = "Candra";
        audience[1][1] = "Dela";
        audience[2][0] = "Eka";
        audience[2][1] = "Farhan";
        audience[3][0] = "Gisel";
        audience[3][1] = "Hana";

        System.out.printf(format:"%s \t %s\n", audience[0][0], audience[0][1]);
        System.out.printf(format:"%s \t %s\n", audience[1][0], audience[1][1]);
        System.out.printf(format:"%s \t %s\n", audience[2][0], audience[2][1]);
        System.out.printf(format:"%s \t %s\n", audience[3][0], audience[3][1]);

        //display the length of each row in the array using foreach loop
        for (String[] rowAudience : audience) {
            System.out.println("Length of row: " + rowAudience.length);
        }
```

```
1ec\bin' 'Cinema24'
Amin     Bena
Candra   Dela
Eka      Farhan
Gisel    Hana
Length of row: 2
Length of row: 2
Length of row: 2
Length of row: 2
```

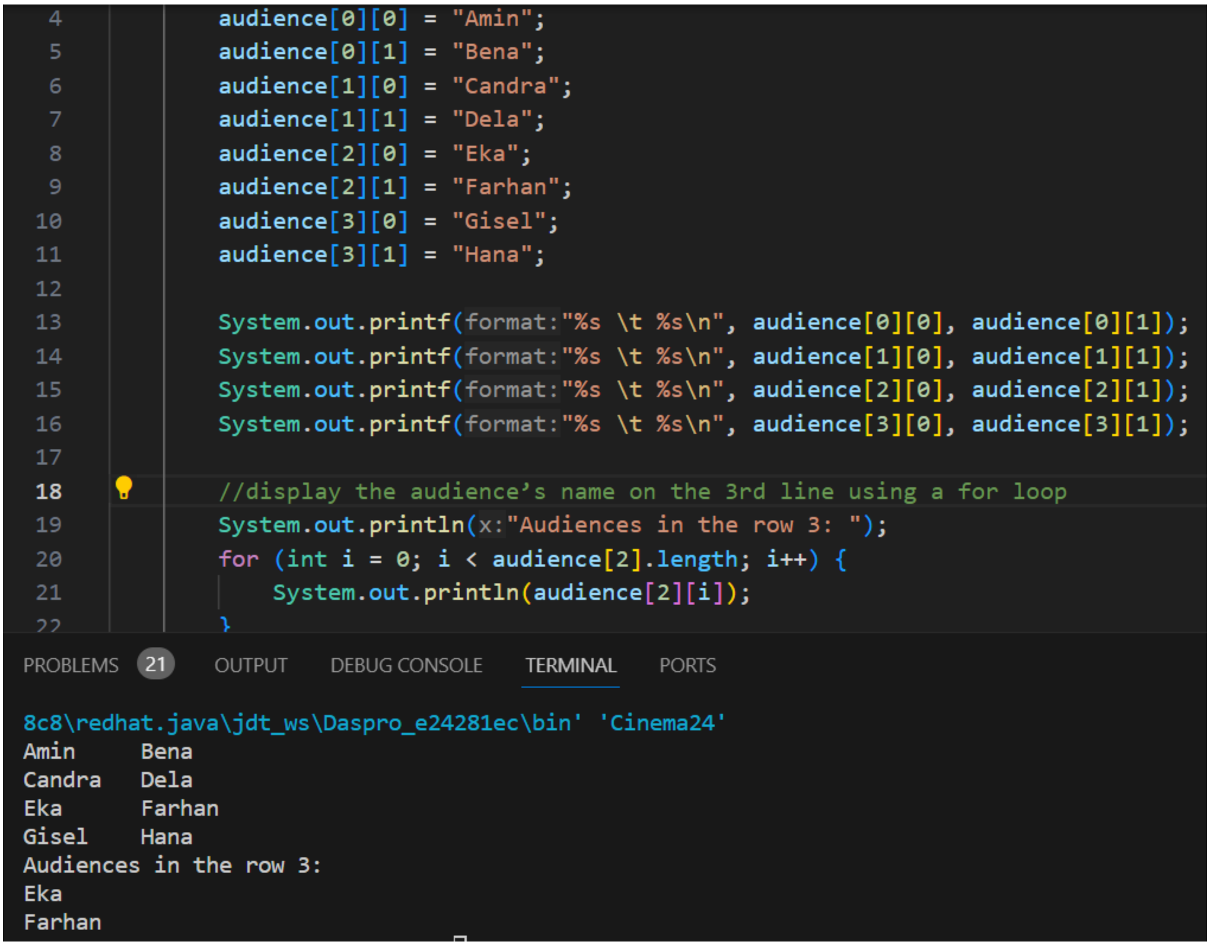
7. Advantages: 1. The foreach loop is more concise and can enhance the readability of the code, 2. No need to manage indices explicitly, reducing the chance of off-by-one errors or other index-related mistakes, 3. Less likely to encounter ArrayIndexOutOfBoundsException errors because not directly accessing elements using indices.

   Disadvantages: 1. The foreach loop doesn't provide direct access to the index of the current element, which can be a limitation in certain scenarios where we need the index, 2. we cannot modify the elements of the collection or array being iterated over while using a foreach loop. This can be a limitation if we need to update elements during iteration.

8. The maximum row index for the audience array is 3, because it has 4 rows (dimension 4x2), so the valid row indices are 0, 1, 2, and 3.

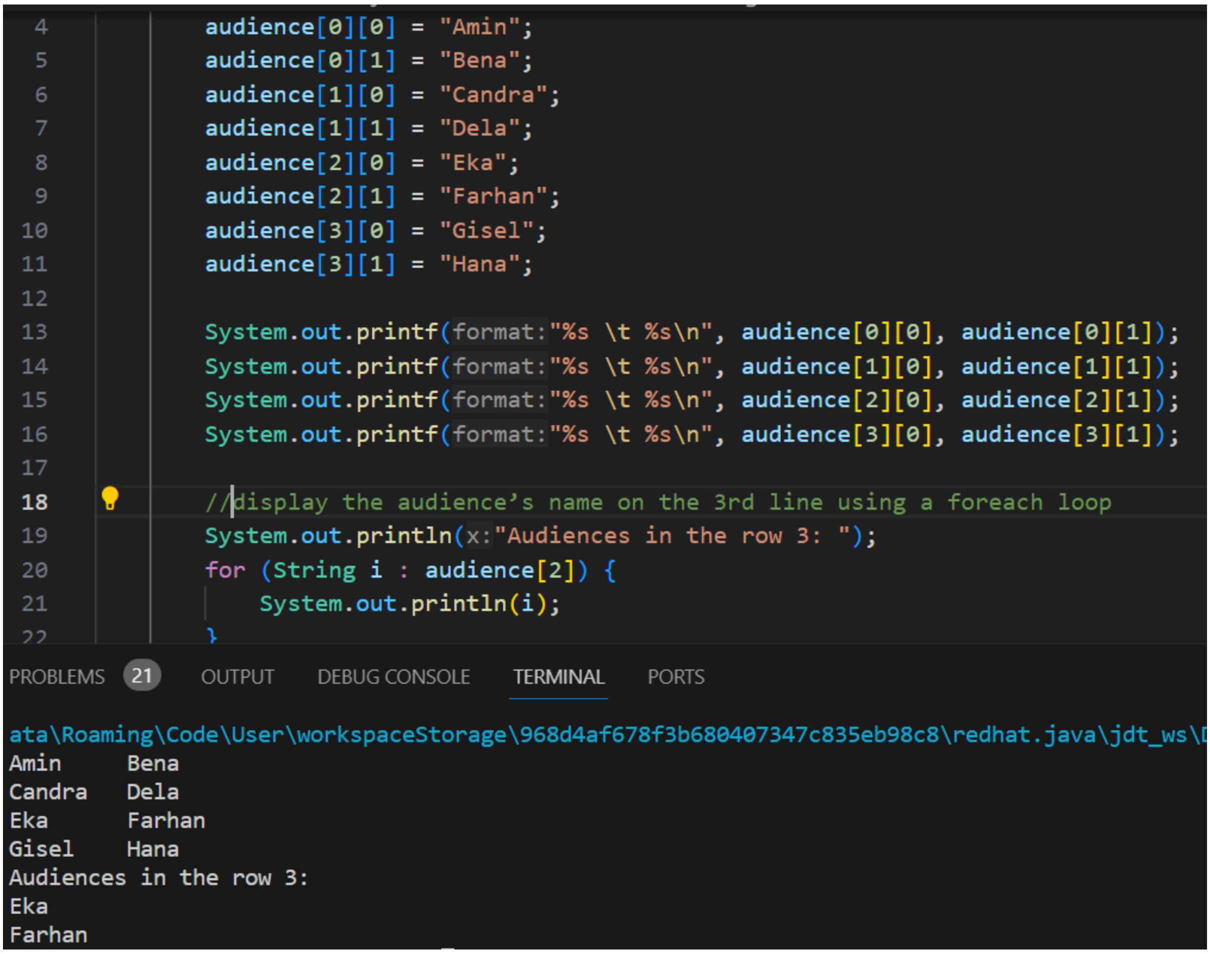
9. The maximum column index for the audience array is 1, because the each row has 2 columns (dimension 4x2), so the valid column indices are 0 and 1.

10.



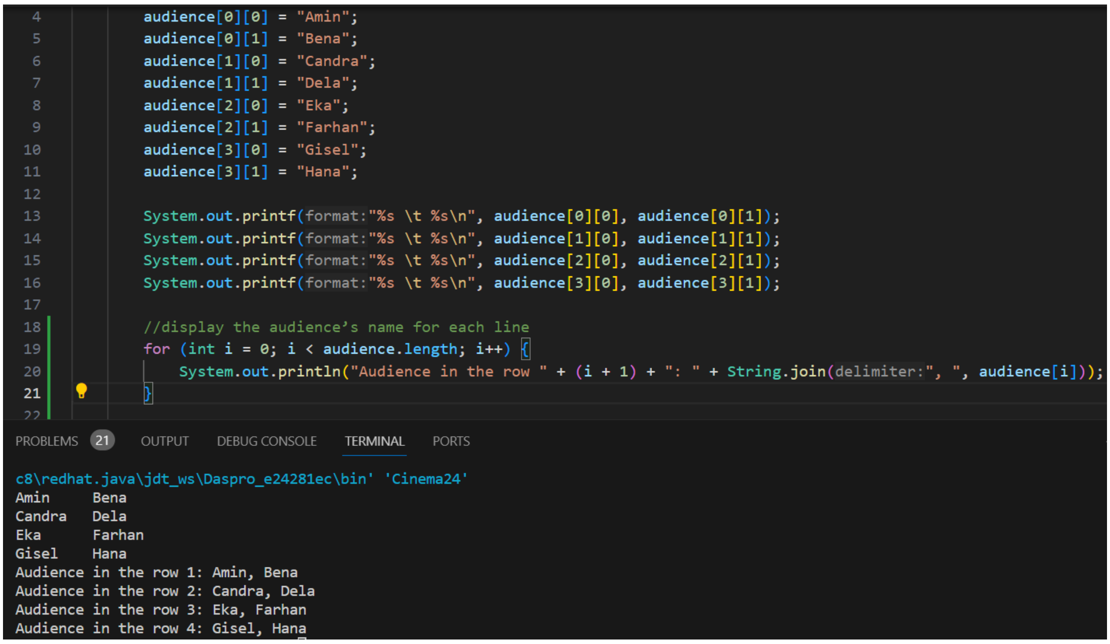
```
    audience[0][0] = "Amin";
    audience[0][1] = "Bena";
    audience[1][0] = "Candra";
    audience[1][1] = "Dela";
    audience[2][0] = "Eka";
    audience[2][1] = "Farhan";
    audience[3][0] = "Gisel";
    audience[3][1] = "Hana";

    System.out.printf(format:"%s \t %s\n", audience[0][0], audience[0][1]);
    System.out.printf(format:"%s \t %s\n", audience[1][0], audience[1][1]);
    System.out.printf(format:"%s \t %s\n", audience[2][0], audience[2][1]);
    System.out.printf(format:"%s \t %s\n", audience[3][0], audience[3][1]);

    //display the audience’s name on the 3rd line using a for loop
    System.out.println(x:"Audiences in the row 3: ");
    for (int i = 0; i < audience[2].length; i++) {
        System.out.println(audience[2][i]);
    }
```

```
8c8\redhat.java\jdt_ws\Daspro_e24281ec\bin' 'Cinema24'
Amin     Bena
Candra   Dela
Eka      Farhan
Gisel    Hana
Audiences in the row 3:
Eka
Farhan
```

11. 

```java
        audience[0][0] = "Amin";
        audience[0][1] = "Bena";
        audience[1][0] = "Candra";
        audience[1][1] = "Dela";
        audience[2][0] = "Eka";
        audience[2][1] = "Farhan";
        audience[3][0] = "Gisel";
        audience[3][1] = "Hana";

        System.out.printf(format:"%s \t %s\n", audience[0][0], audience[0][1]);
        System.out.printf(format:"%s \t %s\n", audience[1][0], audience[1][1]);
        System.out.printf(format:"%s \t %s\n", audience[2][0], audience[2][1]);
        System.out.printf(format:"%s \t %s\n", audience[3][0], audience[3][1]);

        //display the audience’s name on the 3rd line using a foreach loop
        System.out.println(x:"Audiences in the row 3: ");
        for (String i : audience[2]) {
            System.out.println(i);
        }
```

```
ata\Roaming\Code\User\workspaceStorage\968d4af678f3b680407347c835eb98c8\redhat.java\jdt_ws\D
Amin     Bena
Candra   Dela
Eka      Farhan
Gisel    Hana
Audiences in the row 3:
Eka
Farhan
```

12. 

```java
        audience[0][0] = "Amin";
        audience[0][1] = "Bena";
        audience[1][0] = "Candra";
        audience[1][1] = "Dela";
        audience[2][0] = "Eka";
        audience[2][1] = "Farhan";
        audience[3][0] = "Gisel";
        audience[3][1] = "Hana";

        System.out.printf(format:"%s \t %s\n", audience[0][0], audience[0][1]);
        System.out.printf(format:"%s \t %s\n", audience[1][0], audience[1][1]);
        System.out.printf(format:"%s \t %s\n", audience[2][0], audience[2][1]);
        System.out.printf(format:"%s \t %s\n", audience[3][0], audience[3][1]);

        //display the audience’s name for each line
        for (int i = 0; i < audience.length; i++) {
            System.out.println("Audience in the row " + (i + 1) + ": " + String.join(delimiter:", ", audience[i]));
        }
```

```
c8\redhat.java\jdt_ws\Daspro_e24281ec\bin' 'Cinema24'
Amin     Bena
Candra   Dela
Eka      Farhan
Gisel    Hana
Audience in the row 1: Amin, Bena
Audience in the row 2: Candra, Dela
Audience in the row 3: Eka, Farhan
Audience in the row 4: Gisel, Hana
```

13. The String.join() method in Java is used to concatenate multiple strings into a single string.

Question! (Experiment 2)

1. Should the array elements from the scanner be filled in sequentially starting from the 0th index? Please explain!
2. Modify the program code to provide the following menu options:

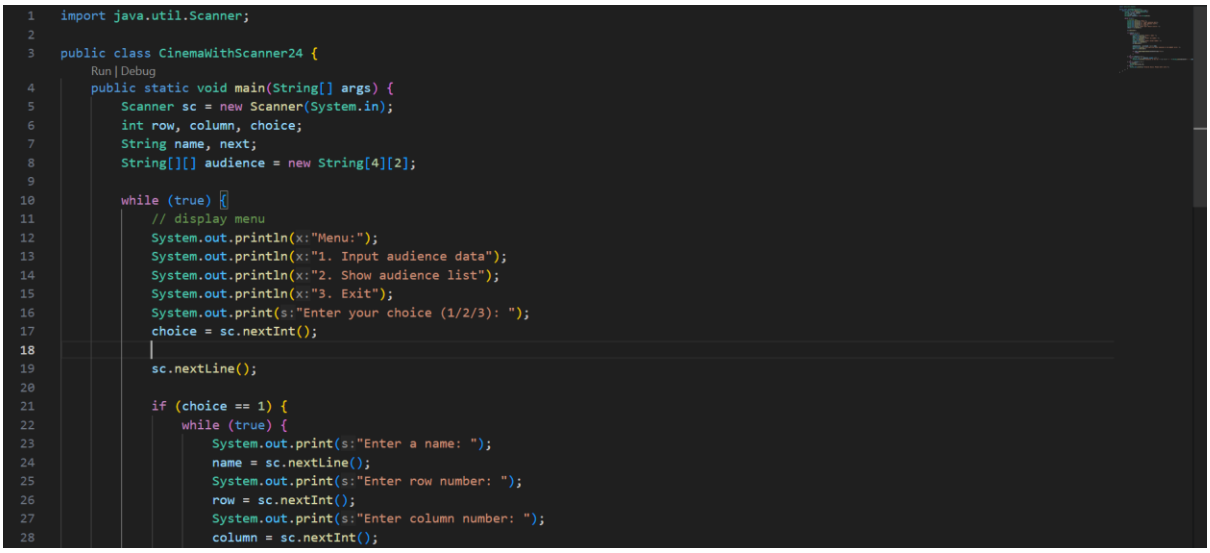
   - Menu 1: Input audience data

   - Menu 2: Show audience list

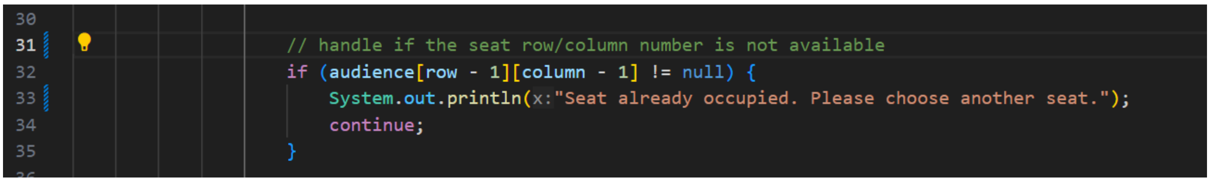
   - Menu 3: Exit
3. Modify the program code to handle if the seat row/column number is not available
4. In menu 1, modify the program code to give a warning if the selected seat is already occupied by other audiences, then display a command to enter rows and columns again
5. In menu 2, if the seat is empty, replace null with ***
6. Commit and push the program code to GitHub

Answer!

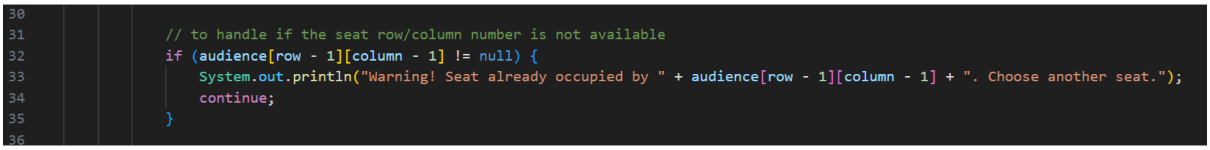
1. The decision of whether the array elements from the scanner should be filled sequentially starting from the 0th index depends on the requirements of the program and how we want to organize the data.

2. 

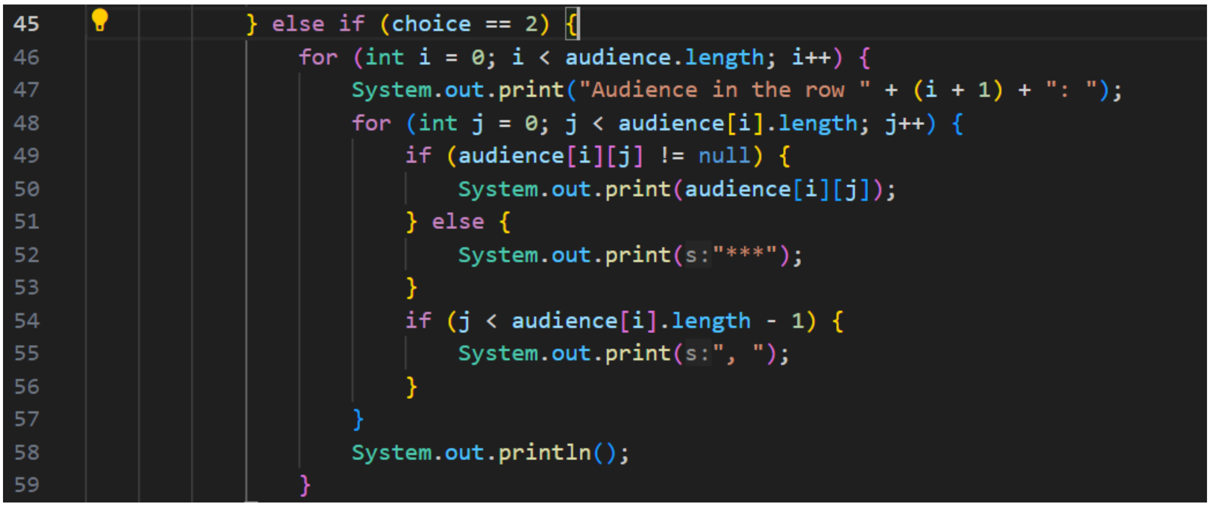
```java
import java.util.Scanner;

public class CinemaWithScanner24 {
    Run | Debug
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int row, column, choice;
        String name, next;
        String[][] audience = new String[4][2];

        while (true) {
            // display menu
            System.out.println(x:"Menu:");
            System.out.println(x:"1. Input audience data");
            System.out.println(x:"2. Show audience list");
            System.out.println(x:"3. Exit");
            System.out.print(s:"Enter your choice (1/2/3): ");
            choice = sc.nextInt();
            
            sc.nextLine();

            if (choice == 1) {
                while (true) {
                    System.out.print(s:"Enter a name: ");
                    name = sc.nextLine();
                    System.out.print(s:"Enter row number: ");
                    row = sc.nextInt();
                    System.out.print(s:"Enter column number: ");
                    column = sc.nextInt();
```

3. 

```java

                    // handle if the seat row/column number is not available
                    if (audience[row - 1][column - 1] != null) {
                        System.out.println(x:"Seat already occupied. Please choose another seat.");
                        continue;
                    }
```

4. 

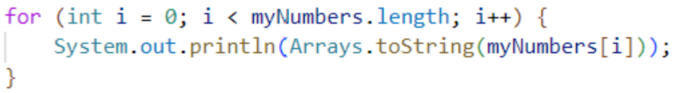
```
        // to handle if the seat row/column number is not available
        if (audience[row - 1][column - 1] != null) {
            System.out.println("Warning! Seat already occupied by " + audience[row - 1][column - 1] + ". Choose another seat.");
            continue;
        }
```

5. 

```
            } else if (choice == 2) {
                for (int i = 0; i < audience.length; i++) {
                    System.out.print("Audience in the row " + (i + 1) + ": ");
                    for (int j = 0; j < audience[i].length; j++) {
                        if (audience[i][j] != null) {
                            System.out.print(audience[i][j]);
                        } else {
                            System.out.print(s:"***");
                        }
                        if (j < audience[i].length - 1) {
                            System.out.print(s:", ");
                        }
                    }
                    System.out.println();
                }
```

Question! (Experiment 3)

1. Add the following program code:

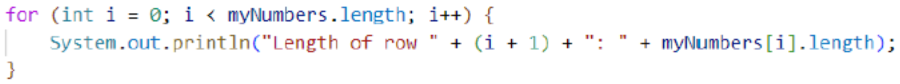


```
for (int i = 0; i < myNumbers.length; i++) {
    System.out.println(Arrays.toString(myNumbers[i]));
}
```

2. What is the function of Arrays.toString()?

3. What is the default value for elements in an array with the data type int?

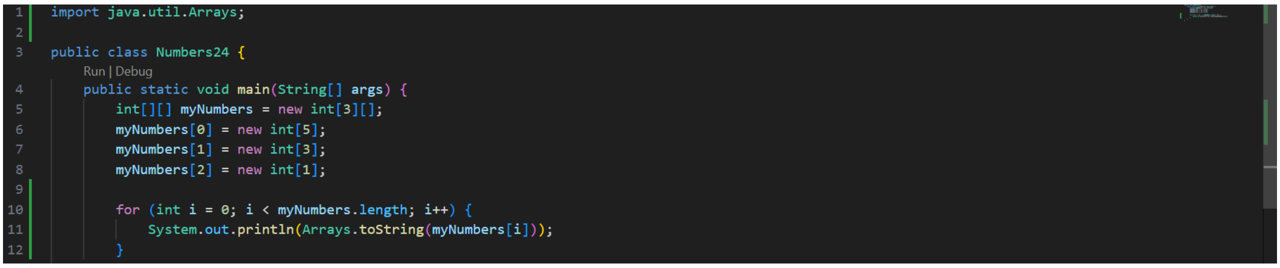
4. Add the following program code:



```
for (int i = 0; i < myNumbers.length; i++) {
    System.out.println("Length of row " + (i + 1) + ": " + myNumbers[i].length);
}
```

5. The myNumbers array has a different length for each row. How to make the length for each row the same? Can the array length be modified?

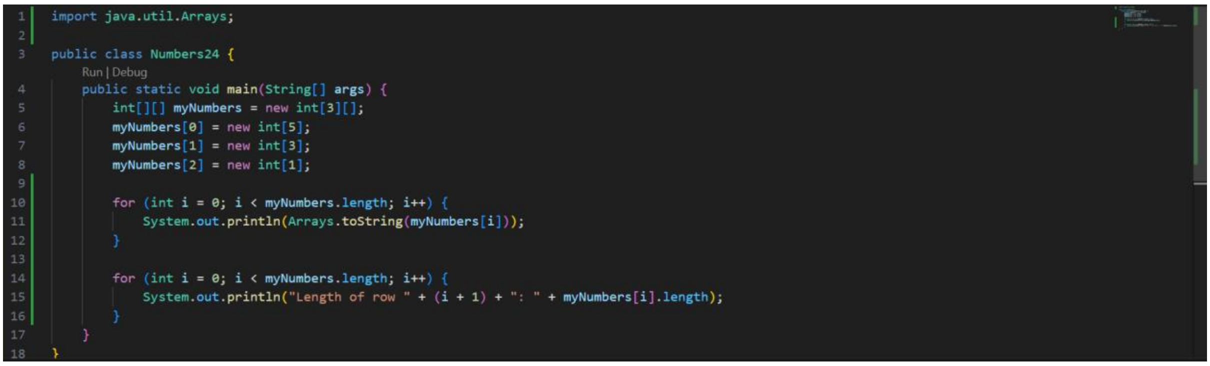
Answer!

1. 

```
import java.util.Arrays;

public class Numbers24 {
    Run | Debug
    public static void main(String[] args) {
        int[][] myNumbers = new int[3][];
        myNumbers[0] = new int[5];
        myNumbers[1] = new int[3];
        myNumbers[2] = new int[1];

        for (int i = 0; i < myNumbers.length; i++) {
            System.out.println(Arrays.toString(myNumbers[i]));
        }
```

2. Arrays.toString() method is used to return a string representation of the contents of an array.

3. The default value for elements in an array of the int data type is 0. When create an array of int and do not explicitly assign values to its elements, all elements in the array will be initialized to 0.

4. 

```java
import java.util.Arrays;

public class Numbers24 {
    public static void main(String[] args) {
        int[][] myNumbers = new int[3][];
        myNumbers[0] = new int[5];
        myNumbers[1] = new int[3];
        myNumbers[2] = new int[1];

        for (int i = 0; i < myNumbers.length; i++) {
            System.out.println(Arrays.toString(myNumbers[i]));
        }

        for (int i = 0; i < myNumbers.length; i++) {
            System.out.println("Length of row " + (i + 1) + ": " + myNumbers[i].length);
        }
    }
}
```

5. The length of an array is fixed at the time of its creation and cannot be modified during the program's execution. If we need to change the size of an array, we need to create a new array with the desired size and copy elements from the old array to the new one.