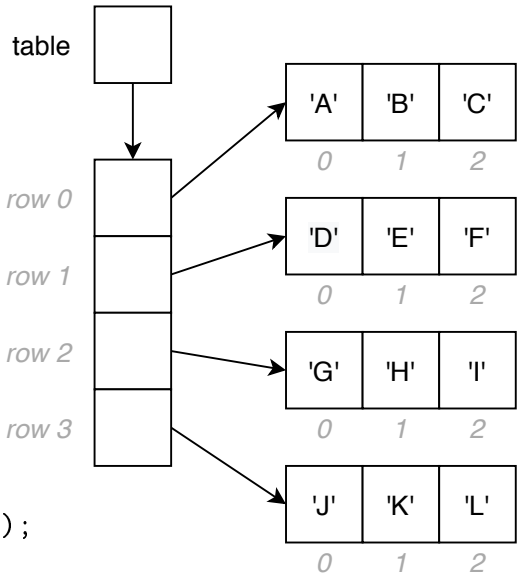
Model 1 Rectangular Array

```java
import java.util.Arrays;  // for toString

public class Rectangular {
    public static void main(String[] args) {
        char[][] table = {
            {'A', 'B', 'C'},
            {'D', 'E', 'F'},
            {'G', 'H', 'I'},
            {'J', 'K', 'L'},
        };
        System.out.println(table.length);
        System.out.println(table[0].length);
        System.out.println(table[1][2]);
        System.out.println(Arrays.toString(table));
    }
}
```



Questions (25 min) Start time:

1. Run the program. What is the output of the following expressions?

a) `table.length` 4 b) `table[0].length` 3 c) `table[1][2]` F

d) `Arrays.toString(table)` `[[C@4dc63996, [C@d716361, [C@6ff3c5b5, [C@3764951d]`

2. Based on the model, what is the data type of the following expressions? Please answer using Java syntax, for example: `String[]`

a) `table` `char[][]` b) `table[0]` `char[]` c) `table[1][2]` `char`

3. Explain the output of `Arrays.toString(table)`. What exactly is stored in the array referenced by the variable `table`?

The output shows four memory addresses. In other words, it's an array of references to other arrays (of characters).

4. Consider the expression `table[1][2]`.

a) Which index (1 or 2) represents the row number? the first one

b) Which index (1 or 2) represents the column number? the second one

5. What is the result of the following expressions?

a) `table[2][1]` H

b) `table[0][0]` A

c) `table[3][2]` L

d) `table[2][3]` out of bounds

Presenter: Implement your team's answers for the following questions in Model1.java, at the end of the main method. Make sure the code compiles and runs correctly.

6. Write three statements to print the first row of the table, one letter at a time. The output should be `A B C ` with a space after each letter.

```
System.out.print(table[0][0] + " ");
System.out.print(table[0][1] + " ");
System.out.print(table[0][2] + " ");
```

7. Summarize the main difference in these three lines of code.

The second index changes from 0 to 1 to 2.

8. Rewrite your previous code using a `for` loop. Name the loop variable `col` (instead of `i`). Your code should work for any number of columns (not just 3).

```
for (int col = 0; col < table[0].length; col++) {
    System.out.print(table[0][col] + " ");
}
```

9. Copy your answer to the previous question and paste it below two times. Modify the code so that it prints row 2 and row 3. Add `println` statements so that each row ends with a newline.

```
for (int col = 0; col < table[1].length; col++) {
    System.out.print(table[1][col] + " ");
}
System.out.println();

for (int col = 0; col < table[2].length; col++) {
    System.out.print(table[2][col] + " ");
}
System.out.println();
```

10. Summarize the main differences in these two `for` loops.

The first index changes from 1 to 2, both in the loop condition and in the loop body.

11. Rewrite your previous code using nested `for` loops. Name the outer loop variable `row` and the inner loop variable `col`. Your code should work for any number of rows and columns.

```
for (int row = 0; row < table.length; row++) {
    for (int col = 0; col < table[row].length; col++) {
        System.out.print(table[row][col] + " ");
    }
    System.out.println();
}
```

12. How would you modify your code in the previous question to print only the top half of the table? (You may assume there is an even number of rows.)

Simply change the for loop header to be:

```
for (int row = 0; row < table.length / 2; row++)
```