**PROJECT Customer Data Simulation: Using Multi - Dimensional Arrays**

**Objective** To type a simple Java program, execute ( run ) the program for some particular values, observe the output and then modify the program.

***PROJECT DESCRIPTION***

Write a program that uses a two - dimensional Java language array, which will simulate the process of counting the number of customers that are serviced through the checkout lanes of a retail store. The simulation will be used to analyze front - end store employee efficiency.  Assuming that the store has seven cashier lanes, your simulation will use random numbers to record customer counts as the fictitious customers pass through their respective cashier lanes.  The two - dimensional array will store the customer count data for each hour, over an eight - hour span.

The initial Java array that you will create will have 6 rows, for each of the cashier lanes, and 8 columns, for each hour in the eight - hour data study.  The 6 times 8 or 48 array entries will be populated by random number generation.  The first lane in the two - dimensional array will be a special 15 items or less checkout lane and thus will take on a larger random number value than the other checkout lanes.

After you populate the array elements you will perform a statistical analysis on the collected data.  The goal of the simulation is to determine if more bagging clerks are required to be utilized for increased productivity and faster checkout times for the store's customers.  The determining factors for adding additional checkout clerks will be when the average number of customers served per hour per lane will be greater than some value, such as 10 .

Run your simulation multiple times ( at least five times ) to ascertain the rationale for adding more clerks.

Type, compile and run the basic Java program that is shown in **Figure 1** , which follows.  
 Then compile and run your program, observe the output then modify the program.

***Information About This Project***

Efficiency testing for the up - front checkout counter lanes in a retail store could mean counting the number of customers that pass through each lane for every hour in the study time frame. This efficiency event can be simulated by a computer program that can generate customer counts with random numbers.

Here is an example of data that can be collected.

**[ Data Simulation ]**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | hour 1 | hour 2 | hour 3 | hour 4 | hour 5 | hour 6 | hour 7 | hour 8 |  |
| **lane 1** | **12** | **16** | **2** | **6** | **10** | **7** | **17** | **15** | ***express checkout*** |
| lane 2 | 6 | 7 | 4 | 1 | 2 | 8 | 8 | 10 |  |
| lane 3 | 9 | 5 | 7 | 8 | 6 | 2 | 5 | 7 |  |
| lane 4 | 6 | 1 | 4 | 6 | 9 | 7 | 8 | 3 |  |
| lane 5 | 10 | 6 | 6 | 7 | 7 | 7 | 1 | 10 |  |
| lane 6 | 1 | 5 | 7 | 9 | 10 | 6 | 3 | 9 |  |

We can perform row analysis on the about table data and find the average number of customers per lane. We can also perform column analysis on the about table data and find the average number of customers per hour.

**PROJECT Customer Data Simulation: Using Multi - Dimensional Arrays**

***Steps to Complete This Project***

**STEP 1**  **Open NetBeans**

Open NetBeans and create a Java project with the following details.

For Project Name include **Lab9**

For the Main Class include **lab9.RetailStore**

In your **Code** window, shown below, copy in the program code shown in **Figure 1** below, in the appropriate places, except substitute your own name in place of Sammy Student.

**Figure 1 Source Code for the Multi - Dimensional Array Program**

|  |
| --- |
| **import java.util.Scanner;**  **import java.util.Random;**  **import javax.swing.JOptionPane;**  **// Sammy Student**  **public class RetailStore**  **{**  **Scanner scan = new Scanner(System.*in*);**    **public static void main(String args[])**  **{**  **int rows = 6;**  **int columns = 8;**  **int[ ][ ] table = new int[rows][columns];**    **Random randomGen = new Random();**  **StringBuilder s = new StringBuilder();**  **int randomInt1 = 0;**  **int randomInt2 = 0;**    **// generate column titles**  **for (int j = 0; j < columns; j++)**  **{**  **s.append("\thr " + (j + 1));**  **}**  **s.append("\n\n");**  **// populate data for each cashier lane**  **for (int i = 0; i < rows; i++)**  **{**  **s.append("lane " + (i + 1));**  **s.append("\t");**  **for (int j = 0; j < columns; j++)**  **{**  **// express checkout lane**  **randomInt1 = 1 + randomGen.nextInt(20);**  **// standard checkout lane**  **randomInt2 = 1 + randomGen.nextInt(10);** |

**PROJECT Customer Data Simulation: Using Multi - Dimensional Arrays**

**Figure 1 Source Code for the Multi - Dimensional Array Program ( continued )**

|  |
| --- |
| **if(i == 0)**  **table[i][j] = randomInt1;**  **else**  **table[i][j] = randomInt2;**  **s.append(table[i][j]);**  **s.append(" ");**  **s.append("\t");**  **}**  **s.append("\n");**  **}**  **System.*out*.println("data simulation: \n\n" + s);**  **System.*out*.println("");**    **// perform data analysis**  **// row analysis**  **String str =   JOptionPane.*showInputDialog*(null, "enter a cashier lane number : ");**  **// subtract 1 to compensate for a zero indexed array**  **int laneNum = Integer.*parseInt*(str) - 1;**  **double average = 0.0, sum = 0.0;**    **for (int j = 0; j < columns ; j++)**  **{**  **sum += table[laneNum][j];**  **}**  **average = sum / columns;**  **System.*out*.println( "" );**    **String outputMsg = "";**  **outputMsg += "\n for cashier lane " + (laneNum + 1);**  **outputMsg += "\n the data analysis is: ";**  **outputMsg += "\n customer count -> " + Math.*round*(sum);**  **outputMsg += "\n average -> " + Math.*round*(average);**    **JOptionPane.*showMessageDialog*(null, outputMsg,**  **"Data Row Analysis", JOptionPane.*PLAIN\_MESSAGE*);**    **// column analysis**  **str = JOptionPane.*showInputDialog*(null, "enter an hour number : ");**  **// subtract 1 to compensate for a zero indexed array**  **int hourNum = Integer.*parseInt*(str) - 1;** |

**PROJECT Customer Data Simulation: Using Multi - Dimensional Arrays**

**Figure 1 Source Code for the Multi - Dimensional Array Program ( continued )**

|  |
| --- |
| **// reset the accumulating variable**  **sum = 0;**  **for (int i = 0; i < rows ; i++)**  **{**  **sum += table[i][hourNum];**  **System.*out*.println( table[i][hourNum] );**  **}**  **average = sum / rows;**  **System.*out*.println( "" );**    **outputMsg = "";**  **outputMsg += "\n for hour number " + (hourNum + 1);**  **outputMsg += "\n the data analysis is: ";**  **outputMsg += "\n customer count -> " + Math.*round*(sum);**  **outputMsg += "\n average -> " + Math.*round*(average);**    **JOptionPane.*showMessageDialog*(null, outputMsg,**  **"Data Column Analysis", JOptionPane.*PLAIN\_MESSAGE*);**  **}**  **}** |

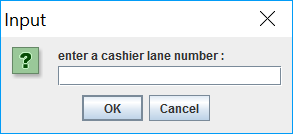
**STEP 2 Build, Compile and Run the Program**

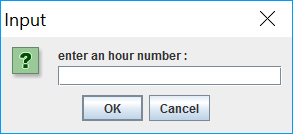
From the NetBeans Run menu select Run Project ( Lab9 ) to run your app.

**STEP 3 Test the Program**

Once you have successfully compiled your program, review the output that appears in the message boxes that follow in **Figure 2** . The output will of course be randomized and changed with each program execution.

With the program running enter valid values in the two input boxes that appear.





**PROJECT Customer Data Simulation: Using Multi - Dimensional Arrays**

Observe your program output which will be similar to that given below.

**Figure 2 Initial Test Run**

|  |
| --- |
| data simulation:  hr 1 hr 2 hr 3 hr 4 hr 5 hr 6 hr 7 hr 8  lane 1 16 17 3 13 12 20 13 9  lane 2 1 4 2 3 2 9 6 5  lane 3 4 5 7 9 6 7 8 4  lane 4 7 1 9 5 7 2 8 1  lane 5 3 9 10 6 10 5 4 1  lane 6 8 2 9 5 7 6 6 5 |

Verify the customer counts and averages that appear in the message boxes.

**STEP 4 Modify the Program**

Once your program runs successfully you can now modify the program to include an additional checkout lane. Keep the same number of columns in your multi - dimensional array but just add another row.

Test your modified program.

**PROJECT Customer Data Simulation: Using Multi - Dimensional Arrays**

Extra Credit: include an if() statement that will determine if the average of either your row or column analysis is more than 10 .

**STEP 5 Submit Your Project**

Once you have determined that your modified program is correctly displaying the required information, complete the submission process as follows:

Open MS Word and type a heading for a new document that includes your full name, course number, lab number and date.

Within the document paste snapshots of your modified program in action. Label the snapshots of your modified run with a reasonable description.

After your snapshot, paste in your finished source code as well copied in from your NetBeans editor.

Submit your MS Word document to Blackboard when complete.