

## Activity No. 5.1

### Multidimensional Arrays

**Course Code:** CPE007

**Program:** Computer Engineering

**Course Title:** Computer Programming Logic and Design

**Date Performed:** Sept 30, 2025

**Section:** CPE11S1

**Date Submitted:** Sept 30, 2025

**Name(s):** Lopez, Andrei Dion C.

**Instructor:** Engr. Jimlord M. Quejado

#### 6. Output

##### Code 1:

- Screenshot of Code(Readable):

```

1 #include <iostream>
2 #include <iomanip>
3 using namespace std;
4
5 int main() {
6     const int size = 10;
7     int multiTable[size][size];
8
9     for (int i = 1; i <= size; ++i) {
10        for (int j = 1; j <= size; ++j) {
11            multiTable[i - 1][j - 1] = i * j;
12        }
13    }
14
15    cout << "      ";
16
17    for (int j = 1; j <= size; ++j) {
18        cout << setw(5) << j;
19    }
20
21    cout << "\n";
22    cout << "      " << string(size * 5 + 1, '-') << "\n";
23
24    for (int i = 1; i <= size; ++i) {
25        cout << setw(2) << i << " | ";
26        for (int j = 1; j <= size; ++j) {
27            cout << setw(5) << multiTable[i - 1][j - 1];
28        }
29        cout << "\n";
30    }
31
32    return 0;
33 }
```

- Output of Code(label and compile ALL possible outputs):

	1	2	3	4	5	6	7	8	9	10
1	1	2	3	4	5	6	7	8	9	10
2	2	4	6	8	10	12	14	16	18	20
3	3	6	9	12	15	18	21	24	27	30
4	4	8	12	16	20	24	28	32	36	40
5	5	10	15	20	25	30	35	40	45	50
6	6	12	18	24	30	36	42	48	54	60
7	7	14	21	28	35	42	49	56	63	70
8	8	16	24	32	40	48	56	64	72	80
9	9	18	27	36	45	54	63	72	81	90
10	10	20	30	40	50	60	70	80	90	100

- To show a multiplication table I first initialized the 2 preprocessor commands “`<iostream>`” and “`<iomanip>`”, The use for the iomanip library is for the keyword “`setw`”, which will allow for a neat alignment of the table. Afterwards I proceeded to declare a constant integer called “`size`” and set its value to 10, then I declared the array and called the array “`multiTable`” and set its dimensions to the value of size. Then I proceeded to start a for loop, this for loop fills the array with every multiplication result starting from 10, afterwards I print out a blank space that will leave an empty space for the border of the multiplication table, then I used another for loop to print out the numbers 1 to 10 and aligned them with the keyword “`setw`” afterwards I put a border under it which outputs another blank and a string of “`-`” to the amount of the value “`size * 5 + 1`”, then afterwards I used another for loop which will first print out the vertical border and another for loop inside to print out the values of the multidimensional array.

## Code 2:

- Screenshot of Code(Readable):

```

1 #include <iostream>
2 using namespace std;
3
4 const int SIZE = 3;
5
6 void clearScreen () {
7     system ("cls"); // For Windows
8 }
9
10 void printBoard (char board [SIZE][SIZE]) {
11     cout << "\n";
12     for (int i = 0; i < SIZE; i++) {
13         for (int j = 0; j < SIZE; j++) {
14             cout << " " << board [i][j];
15             if (j < SIZE - 1) cout << " | ";
16         }
17         cout << "\n";
18         if (i < SIZE - 1) cout << "----+----+----\n";
19     }
20     cout << "\n";
21 }
22
23 bool checkWin (char board [SIZE][SIZE], char player) {
24
25     for (int i = 0; i < SIZE; i++) {
26         if (board [i][0] == player && board [i][1] == player && board [i][2] == player)
27             return true;
28     }
29
30     for (int j = 0; j < SIZE; j++) {
31         if (board [0][j] == player && board [1][j] == player && board [2][j] == player)
32             return true;
33     }
34 }
```

```

34
35     if (board[0][0] == player && board[1][1] == player && board[2][2] == player )
36         return true ;
37     if (board[0][2] == player && board[1][1] == player && board[2][0] == player )
38         return true ;
39
40     return false ;
41 }
42
43 int main () {
44     char board [SIZE][SIZE];
45
46
47     for (int i = 0; i < SIZE; i++) {
48         for (int j = 0; j < SIZE; j++) {
49             board [i][j] = ' ';
50         }
51     }
52
53     int moves = 0;
54     char player = 'X';
55     int row, col;
56
57     printBoard (board);
58
59     while (moves < 9) {
60         cout << "Player " << player << ", enter row and column (0-2): " ;
61         cin >> row >> col;
62
63         if (row < 0 || row >= SIZE || col < 0 || col >= SIZE) {
64             cout << "Out of range. Try again.\n" ;
65             continue ;
66         }
67
68         if (board [row][col] != ' ') {
69             cout << "Cell occupied. Try again.\n" ;
70             continue ;
71         }
72
73         clearScreen ();
74
75         board [row][col] = player ;
76         moves++;
77         printBoard (board);
78
79         if (checkWin( board , player )) {
80             cout << "Player " << player << " wins!\n" ;
81             return 0;
82         }
83
84         player = (player == 'X') ? 'O' : 'X';
85     }
86
87     cout << "It's a draw!\n" ;
88     return 0;
89 }
```

- Output of Code(label and compile ALL\_possible outputs):  
Before User Input:

```
C:\Users\Shinji\Documents\lan.exe

|   |
+---+
|   |
+---+
|   |

Player X, enter row and column (0-2):
```

After User Input of 0 1:

```
C:\Users\Shinji\Documents\lan.exe

| X |
+---+
|   |
+---+
|   |

Player O, enter row and column (0-2):
```

Diagonal Win:

```
X | O |
+---+
O | X |
+---+
O | X | X

Player X wins!

-----
Process exited after 26.15 seconds with return value 0
Press any key to continue . . .
```

Horizontal Win:

```
X | X | X
+---+
O | O |
+---+
|   |

Player X wins!

-----
Process exited after 17.66 seconds with return value 0
Press any key to continue . . .
```

Vertical Win:

```
o | x |
-----+
| x |
-----+
| x | o

Player X wins!

-----
Process exited after 23.73 seconds with return value 0
Press any key to continue . . .
```

- In this tic tac toe program I wanted to use functions so to start I initialized a function that would clear out the screen after a user input, Then I used another function that would handle the printing of the board layout, and lastly I used another function to check if the board has 3 matching symbols diagonally, horizontally and vertically it will then output true if it has matching symbols. Then moving to the main function of the program I first initialized the char “**board**” as an array where its dimensions is the value of size, and then used a for-loop to initialize the empty spaces of the board. Then I initialized a variable for the amount of moves and set its value to 0, a character where its value is the character “X”, and 2 int variables for the rows and columns, after which I then printed out the format of the board using its function name. Using a while loop to check if the amount of moves is less than 9, I used if statements to check if the values inputted by the user is within the array range or if there is a character occupying the values given by the user and output an error message, if it does not meet any of the if statements conditions it would then proceed to clear out the screen and print out the new board with the cell of the given value occupied by the player, It would then check if there is a 3 matching cells diagonally, horizontally, or vertically, if not it will change the character of “**player**” to the next player of the game and loop back to asking for new values for row and column. It will repeat this until the program detects a winner or if all the moves are exhausted which will end the program and output a draw.

## 7. Supplementary Activity

## 8. Conclusion

This activity has allowed me to expand my skills in c++ by using new keywords and libraries and has given me a proper introduction to multidimensional arrays, by making a multiplication table and a simple tic tac toe game I have further expanded my knowledge in the area of C++. The process of using an multidimensional array to store the values of multiple results of a multiplication operation and using logical operators to check if there are matching symbols in a row, column or both to end a program has been a great learning experience for me. Personally I feel that I have done decently in this activity even with the slight inconvenience I had learning this in f2f.