**👩🏽‍💻 C++ 101 – Session 6 Notes**

**1. Multi-Dimensional Arrays in C++**

A **multi-dimensional array** is an array of arrays. The most common example is a **2D array**, but you can go further and create 3D, 4D, and more (though complexity increases).

**🔹 1.1 Declaring a 2D Array**

string letters[2][4];

Creates:

* 2 rows
* 4 columns in each row

**🔹 1.2 Initializing a 2D Array**

string letters[2][4] = {

{ "A", "B", "C", "D" },

{ "E", "F", "G", "H" }

};

**🔹 1.3 Accessing Elements**

cout << letters[0][2]; // Outputs: C

➡️ Index [0][2] accesses row 0, column 2.

**🔹 1.4 Modifying Elements**

letters[0][0] = "Z";

cout << letters[0][0]; // Outputs: Z

**🔹 1.5 Looping Through a 2D Array**

for (int i = 0; i < 2; i++) {

for (int j = 0; j < 4; j++) {

cout << letters[i][j] << "\n";

}

}

**2. 3D Arrays in C++**

A **3D array** is like a cube: it has rows, columns, and depth.

**🔹 Declaration + Initialization:**

string letters[2][2][2] = {

{

{ "A", "B" },

{ "C", "D" }

},

{

{ "E", "F" },

{ "G", "H" }

}

};

**🔹 Accessing an Element:**

cout << letters[1][0][1]; // Outputs: F

➡️ This means:

* Outer block 1 (second group)
* Inner row 0 (first row)
* Column 1 (second item)

**✅ 3. Random Number Generation in C++**

Random numbers are useful in games, simulations, and more.

**🔹 Basic Usage:**

#include <cstdlib>

cout << rand(); // Random number between 0 and a very large value

**🔹 Limit the Range:**

int randomNum = rand() % 101; // 0 to 100

**🔹 Different Results Each Time:**

#include <cstdlib>

#include <ctime>

srand(time(0)); // Seed the generator

int randomNum = rand() % 101; // Random number from 0 to 100

cout << randomNum;

🧠 **Why seed it?** Without srand(time(0)), you’ll get the same "random" number every time.

**🛠️ Task**

**✅ Assignment:**

Write a C++ program that:

1. Declares and initializes a **3D array** (minimum 2×2×2).
2. Uses **nested for loops** to print every element in the array.

📌 Bonus:

* Include meaningful or random values in the array.
* Add comments to explain your logic.