**QUE 1 : What are arrays in C++? Explain the difference between single-dimensional and**

**multi- dimensional arrays.**

**Arrays in C++** are used to store multiple values of the same type in a single variable.  
They are like containers that hold a fixed number of items.

**Types of Arrays**

**1. Single-Dimensional Array**

* A list of elements in one row.
* Example:  
  int numbers[5] = {1, 2, 3, 4, 5};

**2. Multi-Dimensional Array**

* Arrays with two or more dimensions (like a table or grid).

Example (2D array):  
int matrix[2][3] = {

{1, 2, 3},

{4, 5, 6}

};

**Key Difference**

|  |  |
| --- | --- |
| **Single-Dimensional** | **Multi-Dimensional** |
| One row of elements | Rows and columns (or more) |
| arr[3] | arr[2][3], arr[2][3][4] |

**QUE 2 : Explain string handling in C++ with examples.**

**String Handling in C++**

In C++, strings can be handled in **two main ways**:

**1. C-style Strings**

* An array of characters ending with a null character ‘\0'.

Example:  
#include <iostream>

using namespace std;

int main() {

char name[] = "Hello";

cout << name;

return 0;

}

**2. C++ string Class (from <string> library)**

* Easier and safer to use.
* Comes with many built-in functions.

Example:  
  
#include <iostream>

#include <string>

using namespace std;

int main() {

string greeting = "Hello";

cout << greeting;

return 0;

}

**Common String Operations**

|  |  |
| --- | --- |
| **Operation** | **C++ String Example** |
| Length | greeting.length(); |
| Concatenation | greeting + " World"; |
| Input | getline(cin, greeting); |
| Access character | greeting[1]; |
| Compare strings | str1 == str2; |

**QUE 4 : How are arrays initialized in C++? Provide examples of both 1D and 2D arrays.**

**How Are Arrays Initialized in C++?**

Arrays in C++ can be initialized when declared, and they store multiple values of the same data type.

**1. One-Dimensional Array (1D Array)**

**Declaration and Initialization:**

int numbers[5] = {10, 20, 30, 40, 50};

This creates a 1D array of 5 integers.

**Example Program:**

#include <iostream>

using namespace std;

int main() {

int numbers[5] = {10, 20, 30, 40, 50};

cout << "1D Array Elements: ";

for (int i = 0; i < 5; i++) {

cout << numbers[i] << " ";

}

return 0;

}

**2. Two-Dimensional Array (2D Array)**

**Declaration and Initialization:**

int matrix[2][3] = {

{1, 2, 3},

{4, 5, 6}

};

This creates a 2D array with 2 rows and 3 columns.

**Example Program:**

#include <iostream>

using namespace std;

int main() {

int matrix[2][3] = {

{1, 2, 3},

{4, 5, 6}

};

cout << "2D Array Elements:" << endl;

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

for (int j = 0; j < 3; j++) {

cout << matrix[i][j] << " ";

}

cout << endl;

}

return 0;

}

**QUE 4 : Explain string operations and functions in C++.**

**String Operations and Functions in C++**

C++ provides a built-in **string class** (from the <string> header) that makes it easy to perform various operations on strings.

🔹 **Basic String Operations**

|  |  |  |
| --- | --- | --- |
| **Operation** | **Example** | **Description** |
| Declare string | string str = "Hello"; | Create and store text |
| Input string | getline(cin, str); | Read a line with spaces |
| Concatenation | str1 + str2 | Combine two strings |
| Access character | str[0] | Get a character at a position |
| Compare strings | str1 == str2 | Check if two strings are equal |

🔹 **Common String Functions**

|  |  |  |
| --- | --- | --- |
| **Function** | **Description** | **Example** |
| length() / size() | Returns number of characters in string | str.length() |
| empty() | Checks if string is empty | str.empty() |
| append("text") | Adds text to end of string | str.append(" World"); |
| insert(pos, "text") | Inserts text at given position | str.insert(0, "Hi "); |
| replace(pos, len, "text") | Replaces part of string | str.replace(0, 5, "Hi") |
| erase(pos, len) | Removes characters | str.erase(0, 3) |
| substr(pos, len) | Returns part of the string (substring) | str.substr(0, 4) |
| find("text") | Finds index of first match | str.find("World") |

**Example Program**

#include <iostream>

#include <string>

using namespace std;

int main() {

string str1 = "Hello";

string str2 = "World";

// Concatenate

string result = str1 + " " + str2;

cout << "Concatenated: " << result << endl;

// Length

cout << "Length: " << result.length() << endl;

// Substring

cout << "Substring: " << result.substr(0, 5) << endl;

// Replace

result.replace(0, 5, "Hi");

cout << "After Replace: " << result << endl;

// Find

int pos = result.find("World");

cout << "Found 'World' at position: " << pos << endl;

return 0;

}