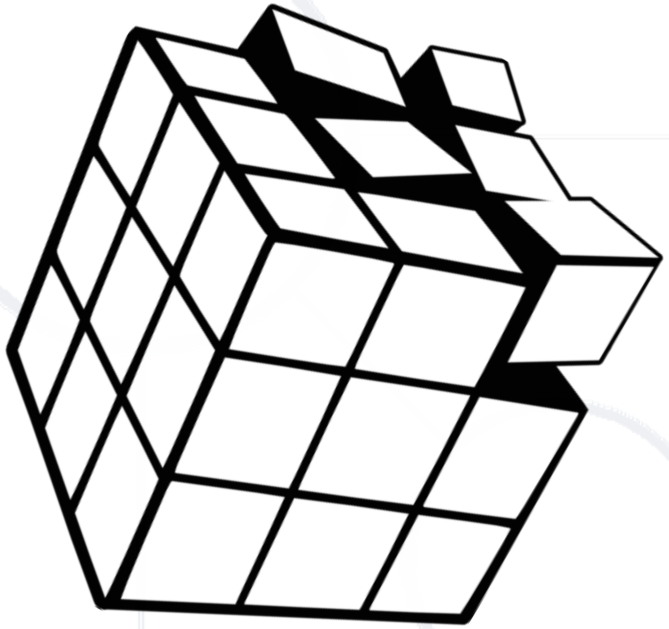
Multidimensional Arrays



SoftUni Team

Technical Trainers

Software University

SoftUni

Software University

https://softuni.bg/

Table of Contents

Software University

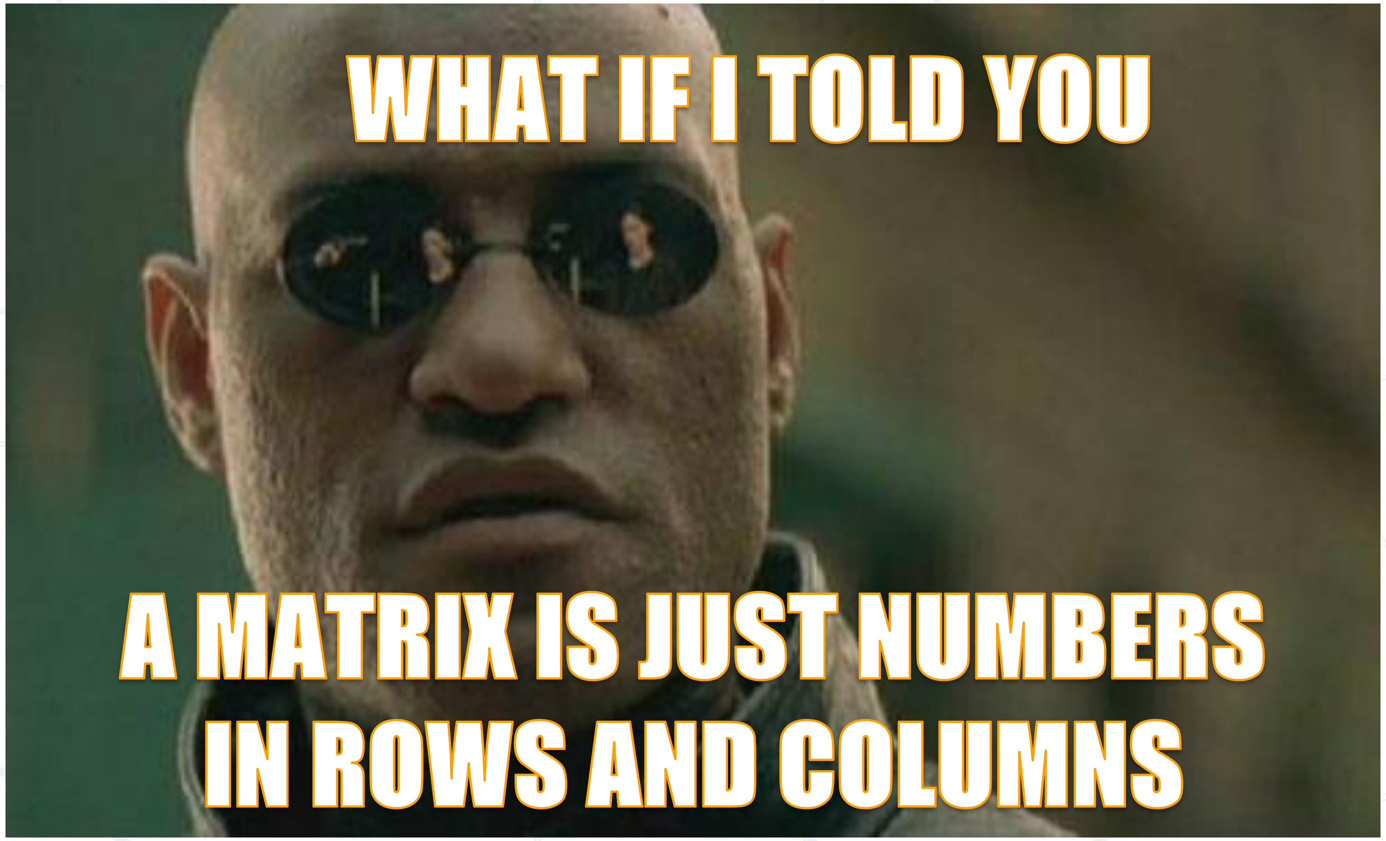
Have a Question?

sli.do

#cpp-advanced

Multidimensional Arrays

Definition and Usage



Multidimensional Arrays

Matrices and Higher Dimensions

What is Multidimensional Array?

Software University

- Array is a systematic arrangement of similar elements
- Multidimensional arrays have more than one dimension
  - They are just normal arrays which are indexed differently
- Most-common usage: making a matrix/table

ROWS	COLS				
	[0][0]	[0][1]	[0][2]	[0][3]	[0][4]
	[1][0]	[1][1]	[1][2]	[1][3]	[1][4]
	[2][0]	[2][1]	[2][2]	[2][3]	[2][4]

Col Index

Row Index

Multidimensional Arrays

- C++ can make arrays act "as if" they have many dimensions
  - "as if" – they are just normal arrays which are indexed differently
  - Compiler enforces dimension syntax in code
- Imagine each element is actually an array
  - 2D (matrix): array of arrays (each element is a "normal" array)
  - 3D array: array of 2D arrays (each element is a matrix)

Accessing Multidimensional Arrays

- Accessing:

Index of row

Index of column

```
int element = matrix[1][0];
```

1st element of the 2nd row

- Accessing elements is done with one indexer per dimension
- Multidimensional arrays represent a **rows with values**
- The rows represent the first dimension and the columns - the second (**the one inside the first**)

Declaring Multidimensional Arrays

- Declaring: add a **size** for each additional dimension

```
int matrix[2][3];
```

```
int matrix[][3];
```

First dimension can omit size if it is a function parameter

Using Multidimensional Arrays

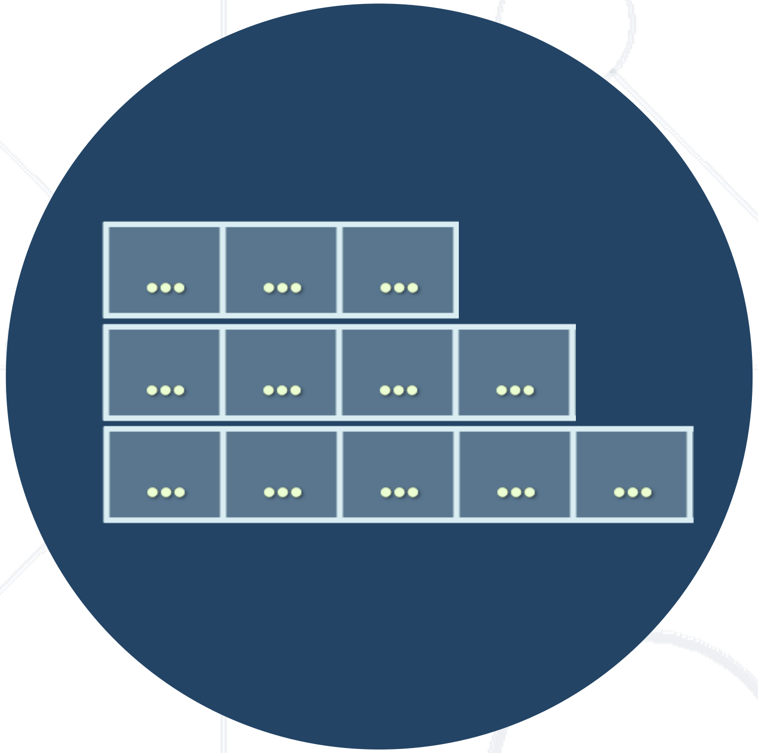
- In this example, each **n**-dimention is an array with **(n - 1)** dimensions

```
int matrix[][3] = {
  { 11, 12, 13 },
  { 21, 22, 23 }
};
```

If no initializer { } brackets, values are undefined

If more elements than initialized, others are defaults

```
int cube[2][3][4] = {
  { {111, 112, 113, 114}, {121, 122, 123, 124}, {131, 132, 133, 134} },
  { {211, 212, 213, 214}, {221, 222, 223, 224}, {231, 232, 233, 234} }
};
```



Multidimensional Arrays

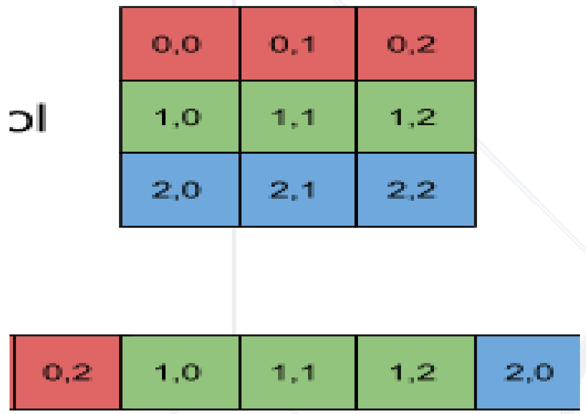
LIVE DEMO

Quick Quiz TIME:

What will the following code do?

a) cause a compile-time error
b) cause a runtime error due to index being out of bounds
c) set `matrix[2][0]` to `0`
d) summon demons
e) you know nothing

```
const int rows = 4;
const int cols = 3;
int matrix[rows][cols] = {
    {11, 12, 13},
    {21, 22, 23},
    {31, 32, 33},
    {41, 42, 43}
};

matrix[1][3] = 0;
```

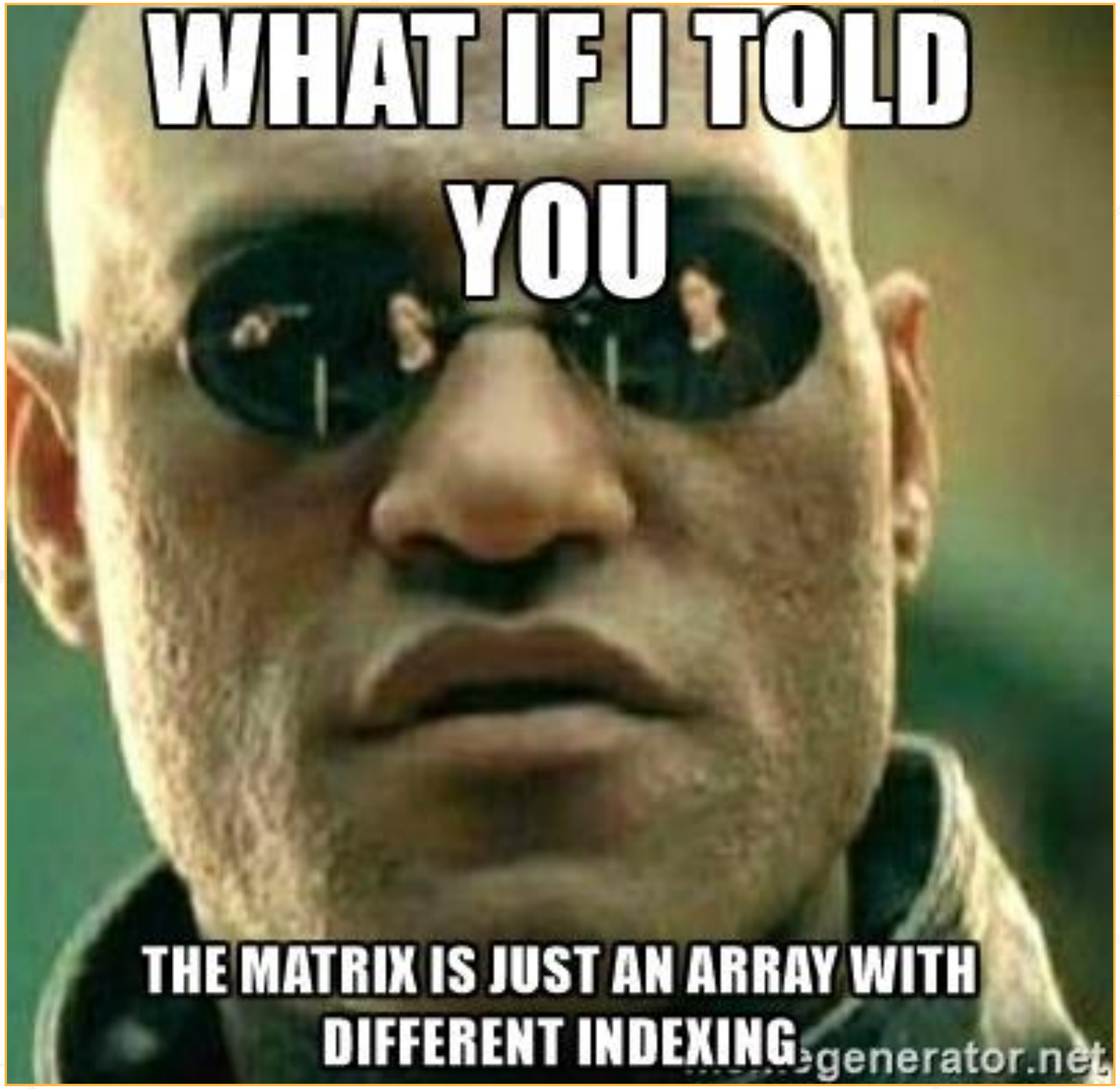


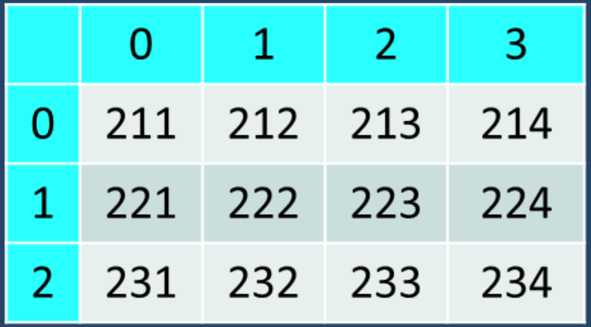
C++ PITFALL: "OUT OF BOUNDS INSIDE" MULTIDIMENSIONAL ARRAYS

C++ (C actually) stores multidimensional arrays as 1D, by joining up together 1st dimension elements, e.g. for 2D arrays – joining up rows into a 1D array.

This is called "row-major order"

E.g. for a `matrix[rows][cols]` accessing `[r][c]` just means `[r * cols + c]` in the actual array





	0	1	2	3
0	211	212	213	214
1	221	222	223	224
2	231	232	233	234

Reading and Printing Matrices

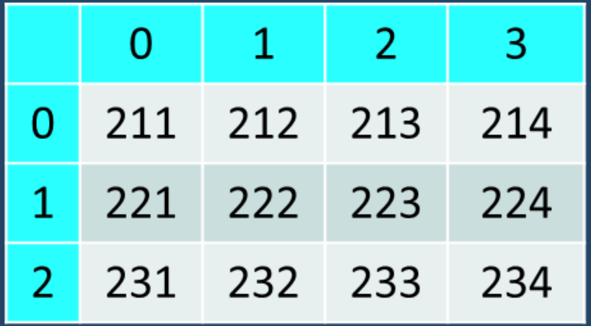
Matrices and Higher Dimensions

Reading a Matrix in C++

```
int main() {
    int a[5][5];
    int row, col;
    cin >> row >> col;

    for (int i = 0; i < row; i++) {
        for (int j = 0; j < col; j++) {
            cin >> a[i][j];
        }
    }

    return 0;
}
```

Printing a Matrix in C++

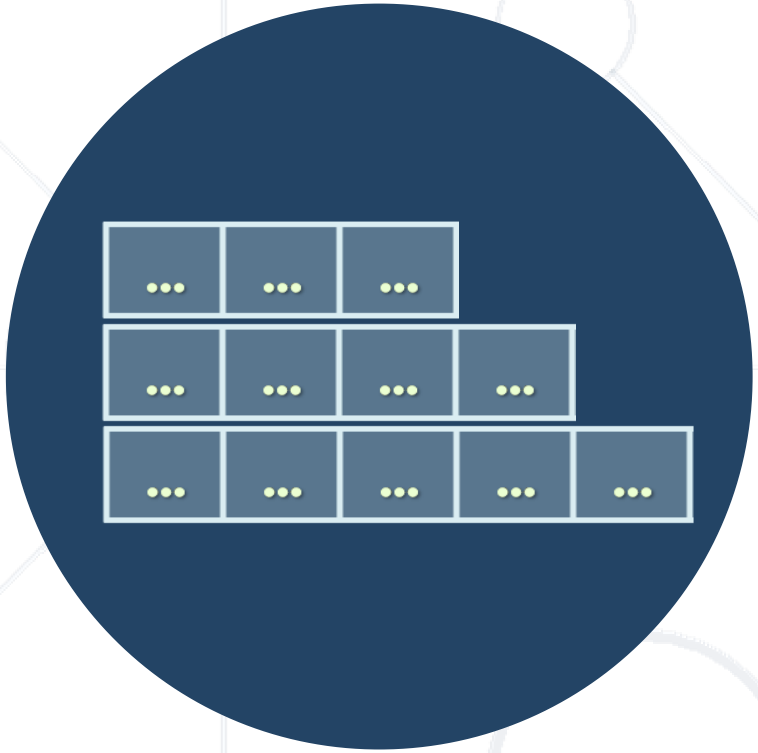
```cpp
int main() {
    int a[5][5];
    int row, col;
    cin >> row >> col;

    for (int i = 0; i < row; i++) {
        for (int j = 0; j < col; j++) {
            cin >> a[i][j];
        }
    }
    for (int i = 0; i < row; i++) {
        for (int j = 0; j < col; j++) {
            cout << a[i][j] << " ";
        }
        cout << endl;
    }
    return 0;
}
```



	0	1	2	3
0	211	212	213	214
1	221	222	223	224
2	231	232	233	234

Reading and Printing Matrices

LIVE DEMO



Passing Arrays to Methods

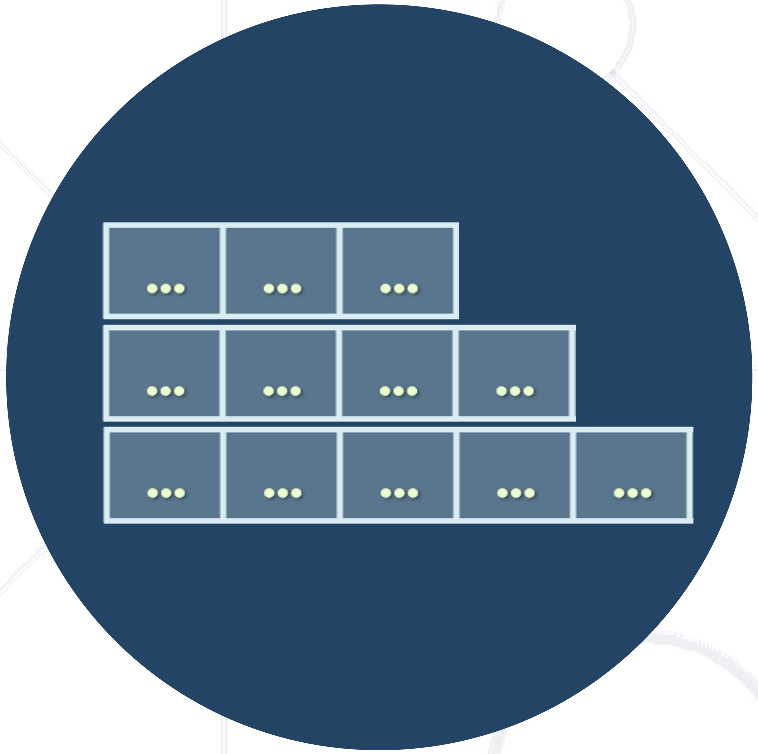
Passing Arrays to Methods

- Arrays can be passed to methods

```
void foo(int arr[3][5])
```

```
void foo(int arr [][5])
```

- The first dimension could be skipped
- NOTE: the array is not copied here (It decays to a pointer. This means it is passed by reference)



C-style Arrays as Function Parameters

LIVE DEMO

"Multidimensional" Containers

- We know `std::vector` can contain any type
  - *Any type with a default constructor*
  - `int`, `double`, `char`, `string`, even another `std::vector`, etc.
- Often containers (e. g. `vector`s) will contain other containers
- E. g. a vector of vectors (2D), a vector of vector of vectors (3D)
  - Element access is the same code as with multidimensional arrays
  - Note: no row-major order (not contiguous in memory)

"Multidimensional" Containers

LIVE DEMO

`std::array` & `std::vector`

- Multidimensional arrays could be created with
  - `std::array`
  - `std::vector`
- If we know the needed size in advance we use `std::array`
- Arrays' data is allocated on the stack
- *We have to be careful not to consume a big portion of the stack, otherwise a stack overflow exception will be thrown*

std::array Matrix

```
const int rows = 3;
const int cols = 5;
// create an empty matrix
std::array<std::array<int, cols>, rows> matrix;

//initialize a matrix
std::array<std::array<int, cols>, rows> matrix {
  { 0, 1, 2, 3, 4 },
  { 1, 2, 3, 4, 5 },
  { 2, 3, 4, 5, 6 }
};
```

C++ Arrays

LIVE DEMO

std::vector Matrix

- If we don't know the size we use a **std::vector**

```
//create an empty matrix
std::vector<std::vector<int>> matrix;

//initialize a matrix
std::vector<std::vector<int>, rows> matrix {
  { 0, 1, 2, 3, 4 },
  { 1, 2, 3 }
  { 2, 3, 4, 5, 6, 7, 8 }
};
```

When we have vectors - the matrix can have any size

N-dimensional Vectors

LIVE DEMO

Working with 2D std::vector

- Working with 2D **`std::vector`** when dealing with **methods**
- A method can return a populated matrix

```
std::vector<std::vector<int>> readMatrix()
```

- A method can accept the 2D std::matrix as a normal function parameter

```
void foo(std::vector<std::vector<int>> matrix); // makes a copy
```

```
void foo(std::vector<std::vector<int>>& matrix); // passed by
reference
```

N-dimensional Vectors as Function Parameters

LIVE DEMO

Problem

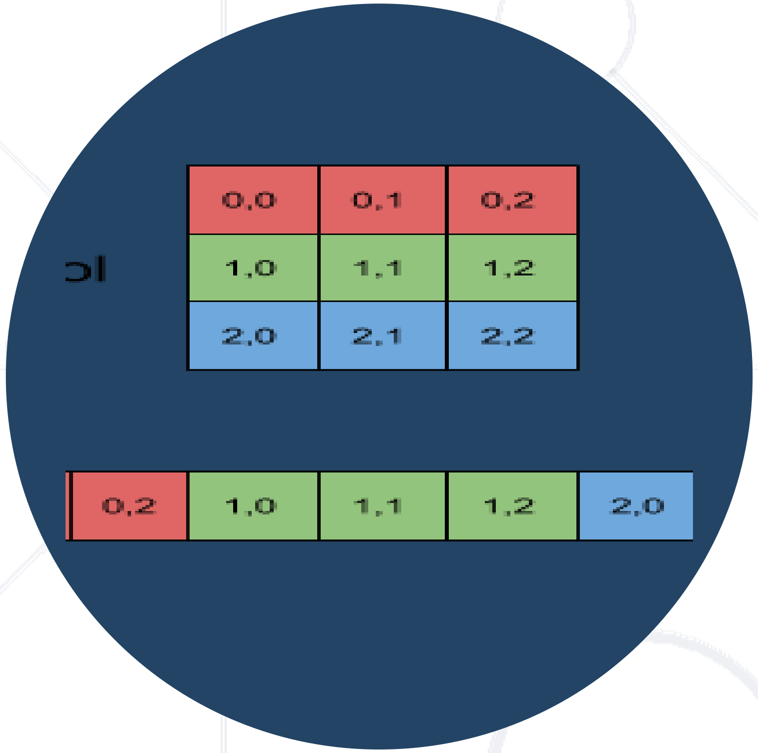
- Create a 2D array of RANDOM integers in the range [0, 100]
- Sums all the integers for every individual column.
- Print to the standard output the column of the 2D array, which has the biggest sum of elements

Practice in Class

LIVE DEMO

C++ - Row-Major Programming Language

- In row-major order:
  - The consecutive elements of a row reside next to each other, whereas the same holds true for consecutive elements of a column in column-major order
- C++ is Row-Major Based



Row-Major Order in Multidimensional Arrays

LIVE DEMO

Problem

- Create a 2D array of RANDOM integers in the range [0, 100]
- Sums all the integers for every individual column
- Print to the standard output the column of the 2D array, which has the biggest sum of elements
- **This time use your new knowledge that C++ is a row-based language**

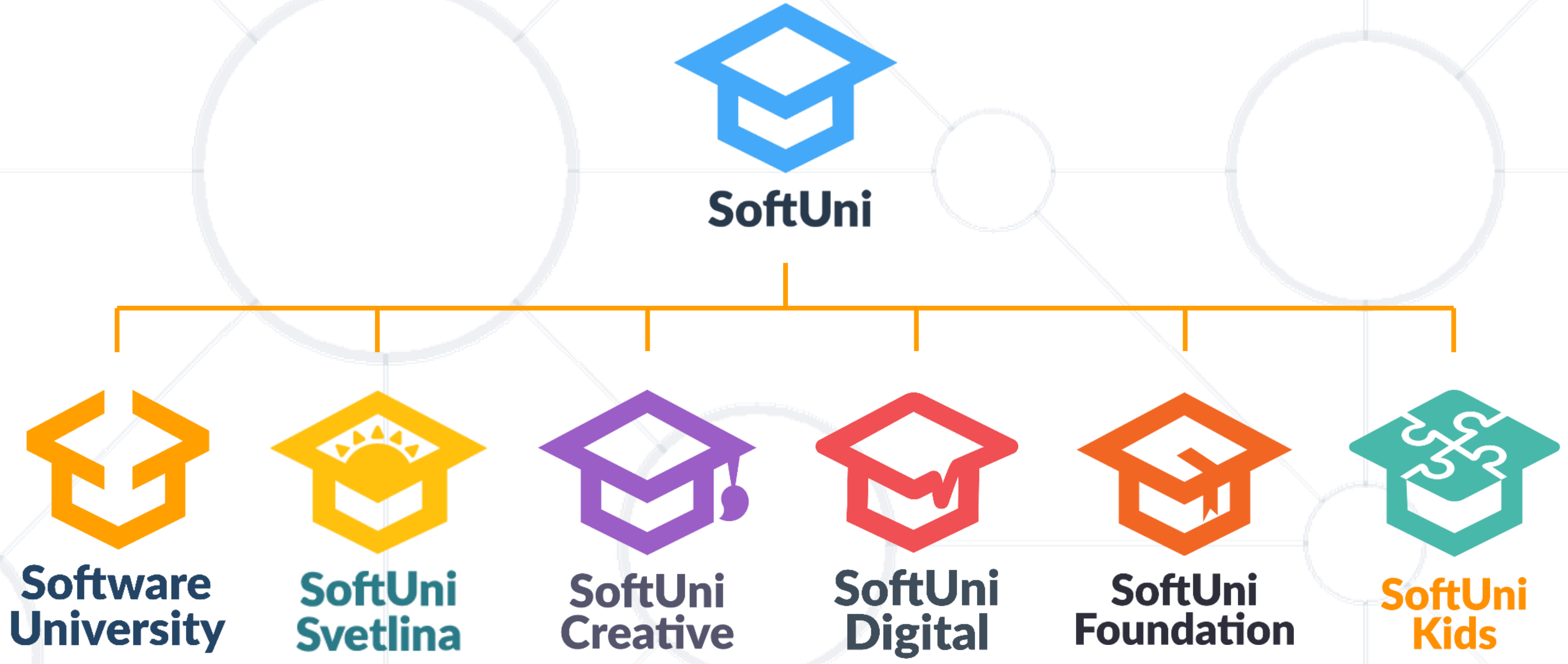
Practice in Class

LIVE DEMO

Summary

- Multidimensional arrays
  - Have **more than one** dimension
  - Two-dimensional arrays are like tables with **rows** and **columns**
  - Most-common usage: making a matrix or a table
- C++ is Row-Major Based

Questions?


SoftUni
Software University
SoftUni Svetlina
SoftUni Creative
SoftUni Digital
SoftUni Foundation
SoftUni Kids

SoftUni Diamond Partners

Software University

DRAFT KINGS

INDEAVR Serving the high achievers

Coca-Cola HBC Bulgaria

SCHWARZ

LIDL

Kaufland

SCHWARZ Produktion

pre zero

Postbank Решения за твоето утре

POKERSTARS | CAREERS

SUPER HOSTING .BG

SOFTWARE GROUP

SmartIT

PHAR VISION

createX

taulia®

Educational Partners

CODE IT UP

VIRTUAL RACING SCHOOL

VRS

License

Trainings @ Software University (SoftUni)

- Software University – High-Quality Education, Profession and Job for Software Developers
  - softuni.bg, about.softuni.bg
- Software University Foundation
  - softuni.foundation
- Software University @ Facebook
  - facebook.com/SoftwareUniversity
- Software University Forums
  - forum.softuni.bg