

CS 204

Advanced Programming

Matrix – Vector of Vectors

Instructor: Duygu Karaoğlu Altop
FENS G001-C, duygu.altop@sabanciuniv.edu

Thanks to Albert Levi and Berrin Yanıkoğlu for the slides

The Matrix

- To represent two dimensional arrays
 - We define a matrix as a vector of vectors

Number
of rows

Number of
columns

```
vector<vector<int> > mat(rows,vector<int>(cols));  
vector<vector<int> > mat(3, vector<int>(5));
```

	0	1	2	3	4
0					
1					
2				100	

```
mat[2][3] = 100;
```

- First index is for row, second is for column

Possible Matrix Definitions

- Possible matrix declarations
 - 4 different declarations

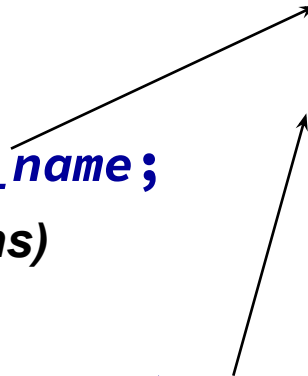
vector<vector<type> > matrix_variable_name;
empty matrix (zero rows, zero columns)

vector<vector<type> > matrix_variable_name(rows);
matrix with *rows* rows; each row is an empty vector<type>

**vector<vector<type> > matrix_variable_name(rows,
vector<type>(cols));**
matrix with *rows*cols* elements
(initialized via default constructor; if *type* is `int`, initialized to zero)

**vector<vector<type> > matrix_variable_name(rows,
vector<int>(cols,
init_value));**
matrix with *rows*cols* elements: all initialized to *init_value*

Use
`push_back`
to fill up



To get the size of rows and columns

matrix_variable_name.size()

e.g. ***mymatrix.size()***

- Number of rows in matrix

matrix_variable_name [0].size()

mymatrix[0].size()

- Number of columns in matrix
- Instead of 0, any valid row index can be used, if each row has equal number of elements; otherwise, the structure is not a matrix