

Multi-dimensional arrays

Multi-dimensional arrays

Example. To declare a 5 x 10 array of integers named `table`, put

```
int[][] table = new int[5][10];
```

Multi-dimensional arrays

Example. Nested **for** loops to print all integers in the array `table`:

```
for (int i = 0; i < 5; i++) {  
    for (int j = 0; j < 10; j++)  
        System.out.print(table[i][j] + " ");  
    System.out.println();  
}
```

Multi-dimensional arrays

Example. Nested **for** loops to print the sum of all numbers in the array `table`:

```
int sum = 0;
for (int i = 0; i < 5; i++)
    for (int j = 0; j < 10; j++)
        sum += table[i][j];
System.out.println("Sum: " + sum);
```

Multi-dimensional arrays

Example. Nested **for** loops to print the maximum number in the array `table`:

```
int maxSoFar = MININT;
for (int i = 0; i < 5; i++)
    for (int j = 0; j < 10; j++)
        if (table[i][j] > maxSoFar)
            maxSoFar = table[i][j];
System.out.println("Max: " + maxSoFar);
```

Multi-dimensional arrays

Exercise. Write nested **for** loops to print the sum of numbers in each row of the array `table`:

```
int sum;
for (int i = 0; i < 5; i++) {
    sum = 0;
    for (int j = 0; j < 10; j++)
        sum += table[i][j];
    System.out.println("Row " + i + ": " + sum);
}
```

Multi-dimensional arrays

Exercise. Write nested **for** loops to construct a multiplication table in an **int** array named `multTable`:

```
int multTable[][] = new int[13][13];
```

```
for (int i = 1; i <= 12; i++)  
    for (int j = 1; j <= 12; j++)  
        multTable[i][j] = i*j;
```

Square arrays

Exercise. Consider an $n \times n$ **int** array M .
Write nested **for** loops to *double each entry* in M :

```
for (int i = 0; i < n; i++)  
    for (int j = 0; j < n; j++)  
        M[i][j] *= 2;
```

Square arrays

Exercise. Consider an $n \times n$ **int** array M . Write nested **for** loops to determine if *all entries* in M are zero:

```
boolean isAllZero = true;
outerLoop;
for (int i = 0; i < n; i++)
    for (int j = 0; j < n; j++)
        if (M[i][j] != 0) {
            isAllZero = false;
            break outerLoop;
        }
```

Square arrays

Exercise. Consider an $n \times n$ **int** array M . Write nested **for** loops to determine if *all diagonal entries* in M are zero:

```
boolean hasZeroDiagonal = true;
for (int i = 0; i < n; i++)
    if (M[i][i] != 0) {
        hasZeroDiagonal = false;
        break;
    }
```

Square arrays

Exercise. Consider an $n \times n$ **int** array M . Write nested **for** loops to determine if M is *symmetric* along the diagonal:

```
boolean isSymmetric = true;
outerLoop;
for (int i = 0; i < n; i++)
    for (int j = 0; j < n; j++)
        if (M[i][j] != M[j][i]) {
            isSymmetric = false;
            break outerLoop;
        }
```

Square arrays

Exercise. Consider an $n \times n$ **int** array M . Write nested **for** loops to determine if M is *symmetric* along the diagonal:

```
boolean isSymmetric = true;
outerLoop;
for (int i = 0; i < n; i++)
    for (int j = 0; j < i; j++)
        if (M[i][j] != M[j][i]) {
            isSymmetric = false;
            break outerLoop;
        }
```