WESTERN SYDNEY
UNIVERSITY

Array Initialisation & Assignment

ELEC1006: ENGINEERING COMPUTING

Array Initialisation

WESTERN SYDNEY UNIVERSITY

- Arrays can be initialised with an initialisation list:

```
const int SIZE = 5;
int tests[SIZE] = {79,82,91,77,84};
```

- The values are stored in the array in the order in which they appear in the list.
- The initialisation list cannot exceed the array size.

Example 1

WESTERN SYDNEY UNIVERSITY

```
7     const int MONTHS = 12;
8     int days[MONTHS] = { 31, 28, 31, 30,
9                          31, 30, 31, 31,
10                         30, 31, 30, 31};
11
12    for (int count = 0; count < MONTHS; count++)
13    {
14       cout << "Month " << (count + 1) << " has ";
15       cout << days[count] << " days.\n";
16    }
```

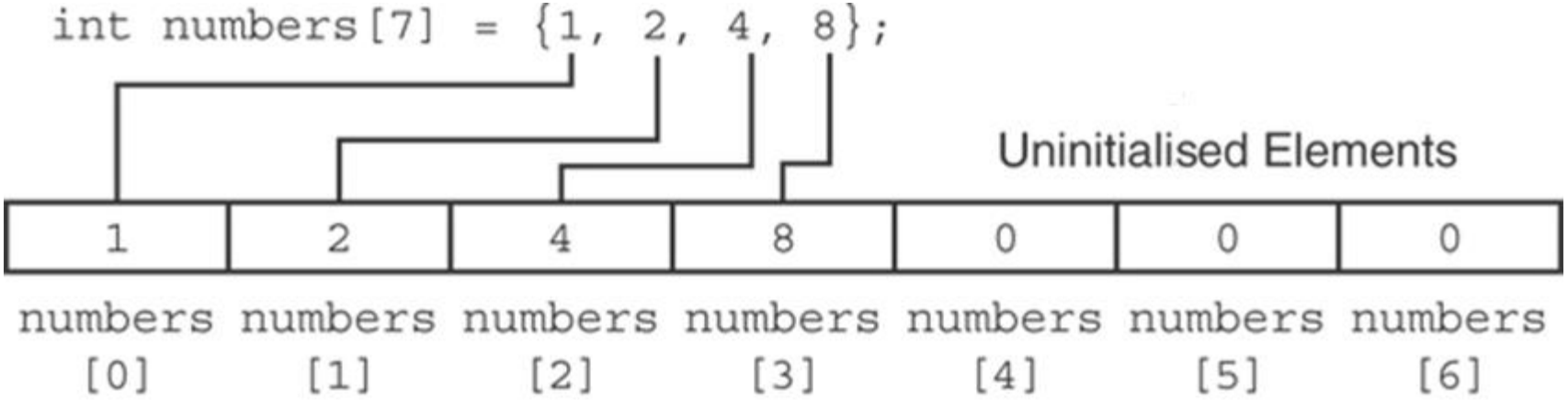
Program Output

```
Month 1 has 31 days.
Month 2 has 28 days.
Month 3 has 31 days.
Month 4 has 30 days.
Month 5 has 31 days.
Month 6 has 30 days.
Month 7 has 31 days.
Month 8 has 31 days.
Month 9 has 30 days.
Month 10 has 31 days.
Month 11 has 30 days.
Month 12 has 31 days.
```

Partial Array Initialisation

WESTERN SYDNEY UNIVERSITY

- If an array is initialised with fewer initial values than the size declarator, the remaining elements will be set to `0`:



Implicit Array Sizing

WESTERN SYDNEY UNIVERSITY

- Can determine array size by the size of the initialisation list:

```
int quizzes[]={12,17,15,11};
```

- Must use either array size declarator or initialisation list for array definition.

Initialising an Array with a String

WESTERN SYDNEY UNIVERSITY

- Character array can be initialised by enclosing string in " ":

```
const int SIZE = 6;
char fName[SIZE] = "Henry";
```

- Must leave room for \0 at end of array
- If initialising character-by-character, must add in \0 explicitly:

```
char fName[SIZE] =
{ 'H', 'e', 'n', 'r', 'y', '\0'};
```

Array Assignment

To copy one array to another,

- Don’t try to assign one array to the other:

```
newTests = tests;  // Won't work
```

- Instead, assign element-by-element:

```
for (i = 0; i < ARRAY_SIZE; i++)
   newTests[i] = tests[i];
```

Printing String Contents of an Array

WESTERN SYDNEY UNIVERSITY

- You can display the contents of a *character* array by sending its name to cout:

```
char fName[] = "Henry";
cout << fName << endl;
```

But, this ONLY works with character arrays!

Printing Numeric Contents of an Array

WESTERN SYDNEY UNIVERSITY

- For other types of arrays, you must print element-by-element:

```
for (i = 0; i < ARRAY_SIZE; i++)
   cout << tests[i] << endl;
```

More info

WESTERN SYDNEY UNIVERSITY

[1] cplusplus.com: Arrays
https://www.cplusplus.com/doc/tutorial/arrays/

[2] leanrcpp.com: 6.1 – Arrays (Part I)
https://www.learncpp.com/cpp-tutorial/61-arrays-part-i/

[3] learncpp.com: 6.2 – Arrays (Part II)
https://www.learncpp.com/cpp-tutorial/62-arrays-part-ii/

[4] learncpp.com: 6.3 – Arrays and loops
https://www.learncpp.com/cpp-tutorial/63-arrays-and-loops/