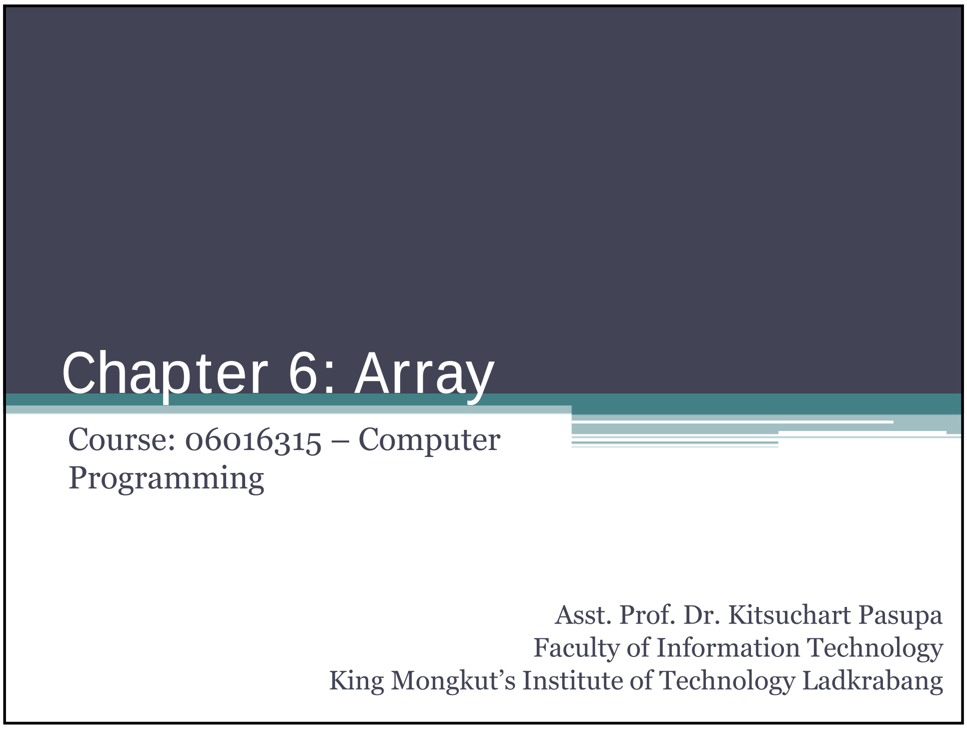
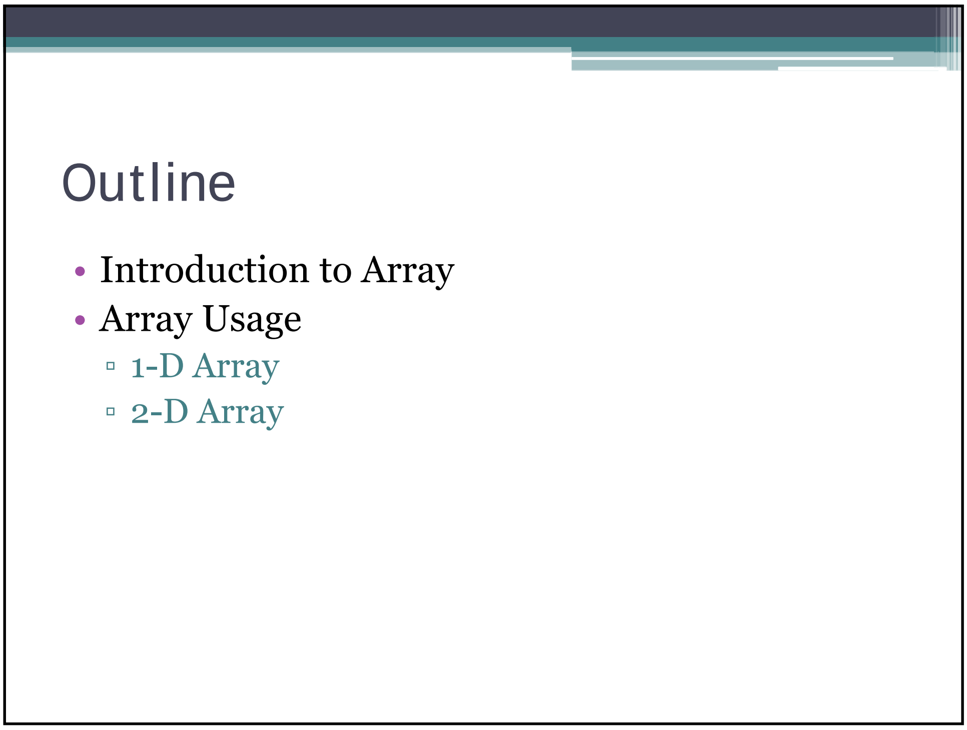


Chapter 6: Array

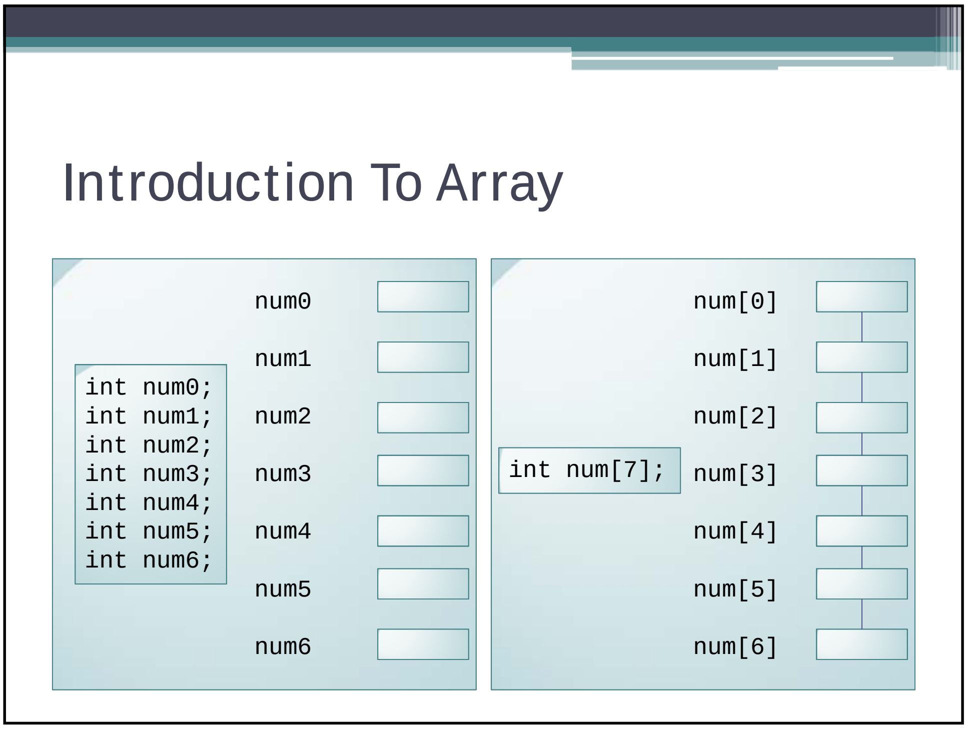
Course: 06016315 – Computer Programming

Asst. Prof. Dr. Kitsuchart Pasupa
Faculty of Information Technology
King Mongkut's Institute of Technology Ladkrabang



Outline

- Introduction to Array
- Array Usage
  - 1-D Array
  - 2-D Array



Introduction To Array

```
int num0;
int num1;
int num2;
int num3;
int num4;
int num5;
int num6;
```

num0

num1

num2

num3

num4

num5

num6

```
int num[7];
```

num[0]

num[1]

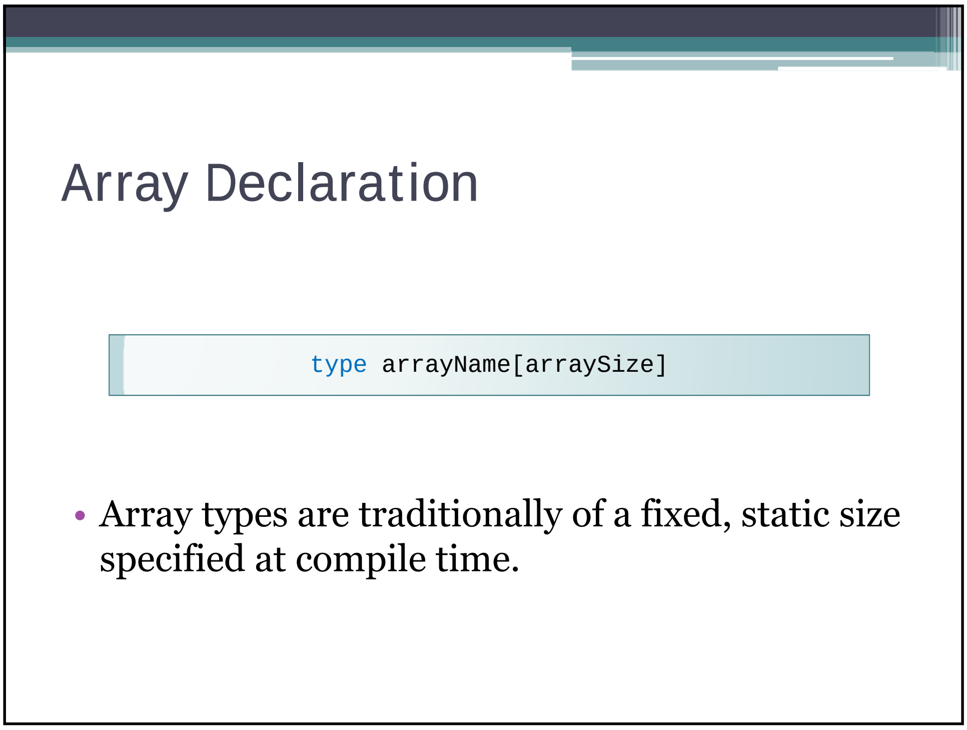
num[2]

num[3]

num[4]

num[5]

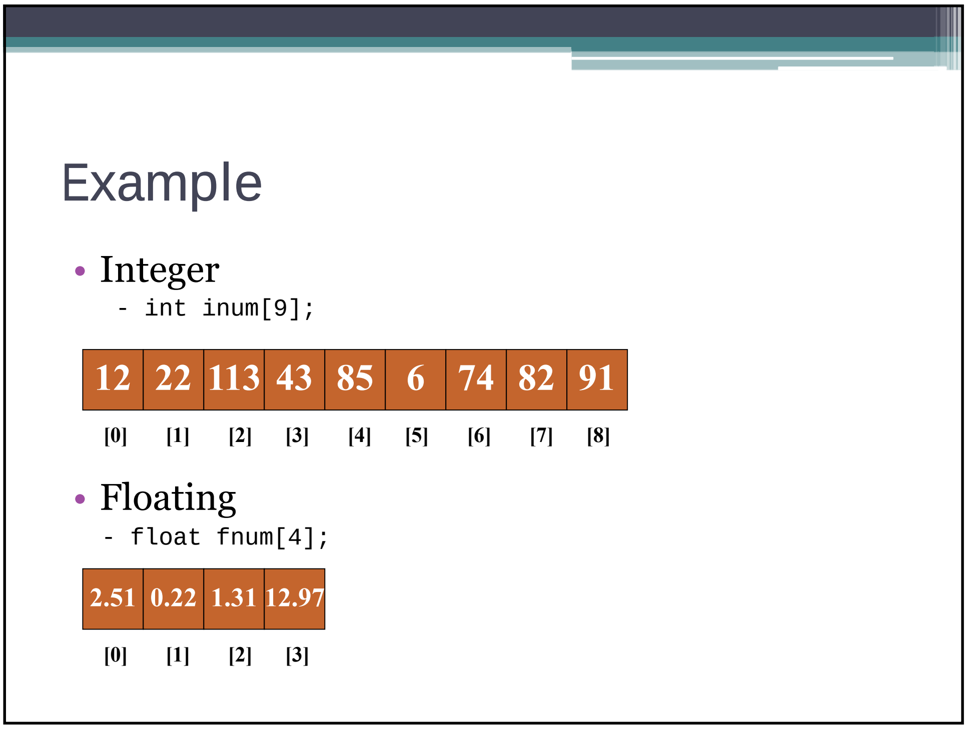
num[6]



Array Declaration

```
type arrayName[arraySize]
```

- Array types are traditionally of a fixed, static size specified at compile time.



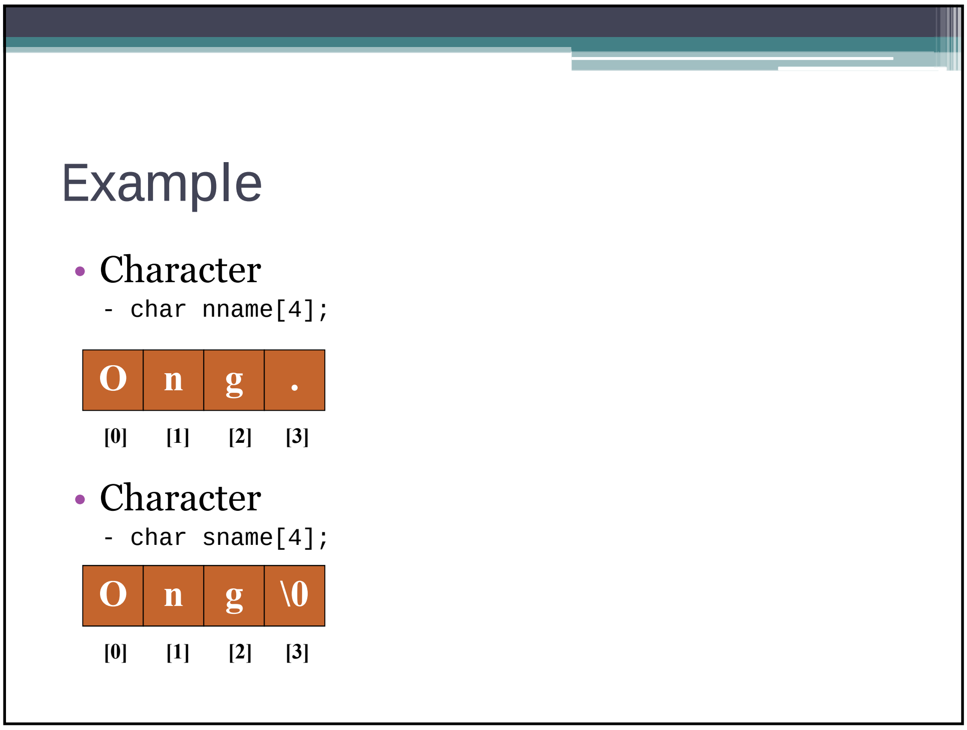
Example

- Integer
  - int inum[9];

12	22	113	43	85	6	74	82	91
[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]

- Floating
  - float fnum[4];

2.51	0.22	1.31	12.97
[0]	[1]	[2]	[3]



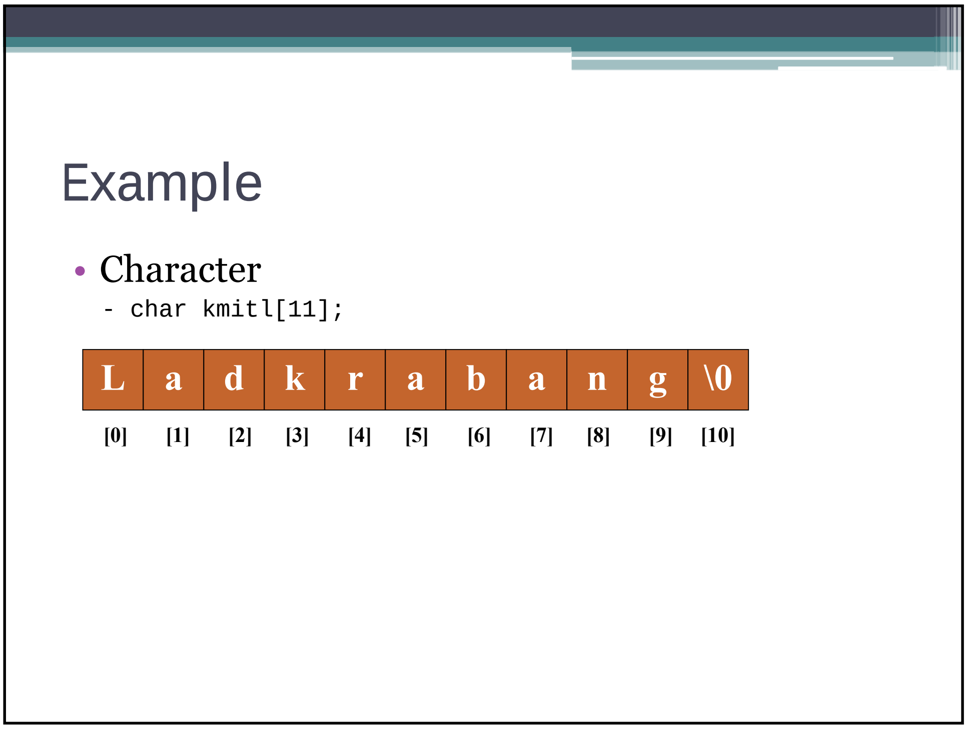
Example

- Character
  - char nname[4];

O	n	g	.
[0]	[1]	[2]	[3]

- Character
  - char sname[4];

O	n	g	\0
[0]	[1]	[2]	[3]

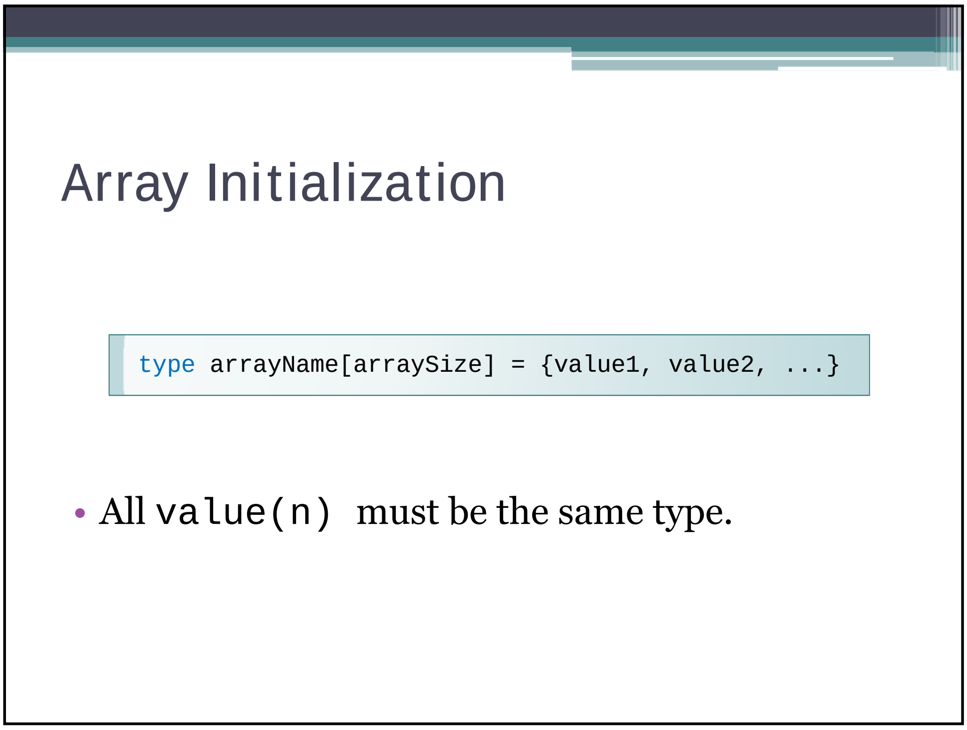


Example

- Character

  - char kmitl[11];

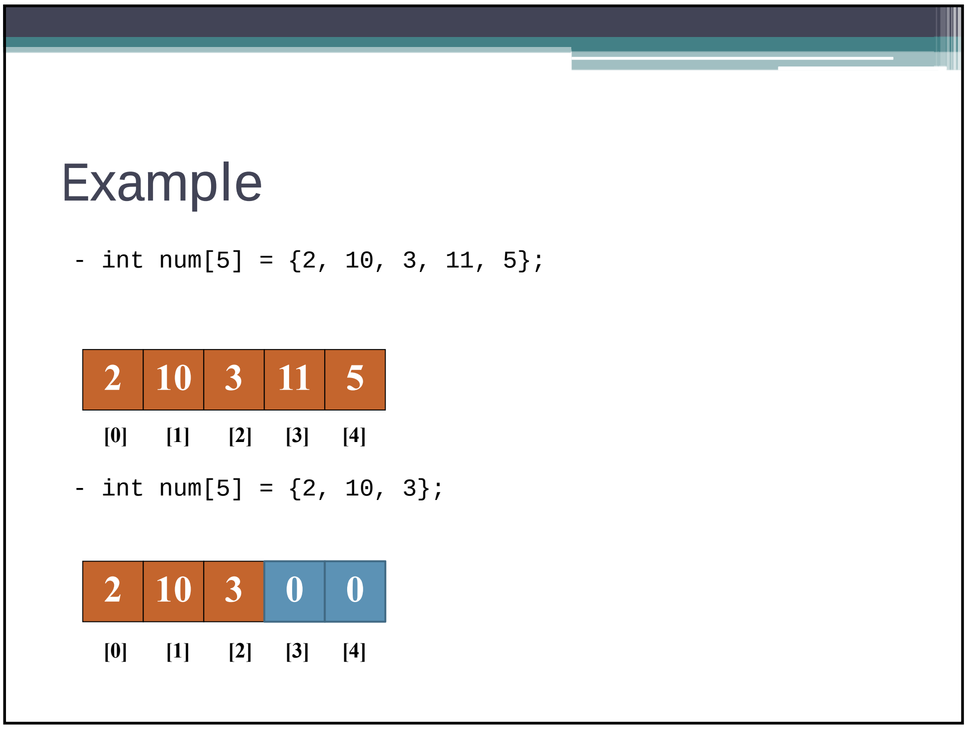
L	a	d	k	r	a	b	a	n	g	\0
[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]



Array Initialization

```
type arrayName[arraySize] = {value1, value2, ...}
```

- All `value(n)` must be the same type.



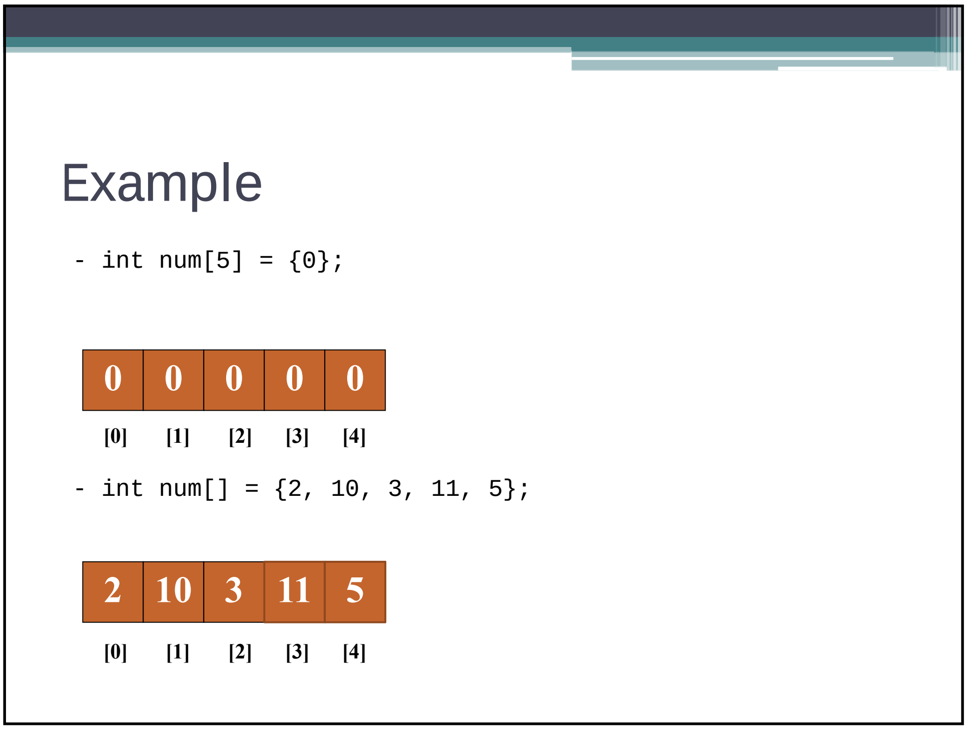
Example

- `int num[5] = {2, 10, 3, 11, 5};`

2	10	3	11	5
[0]	[1]	[2]	[3]	[4]

- `int num[5] = {2, 10, 3};`

2	10	3	0	0
[0]	[1]	[2]	[3]	[4]



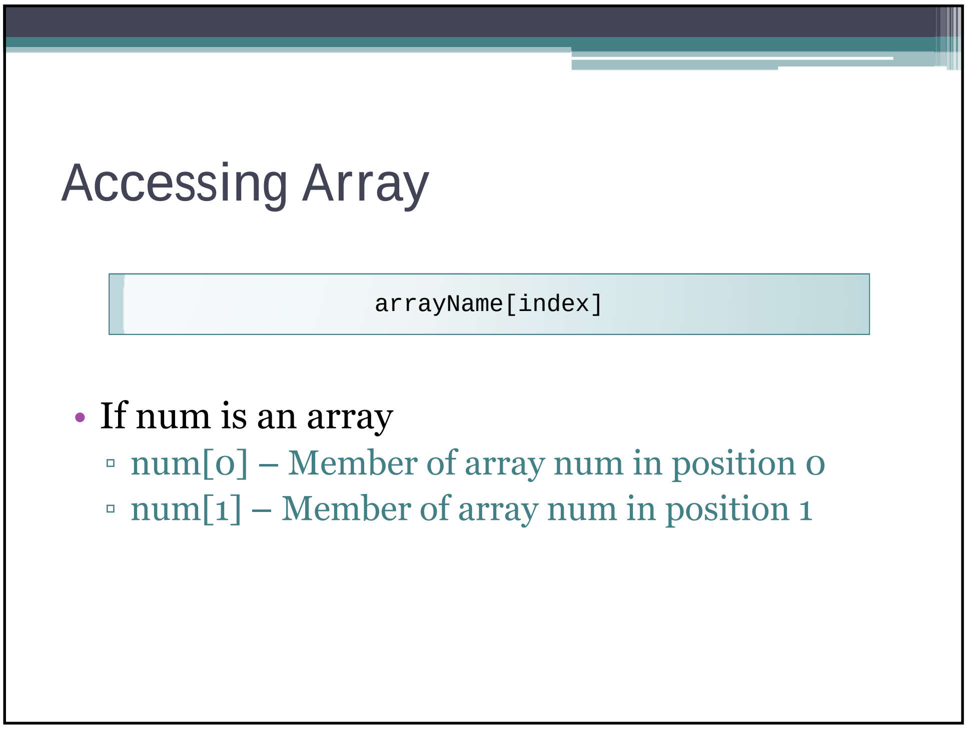
Example

- `int num[5] = {0};`

0	0	0	0	0
[0]	[1]	[2]	[3]	[4]

- `int num[] = {2, 10, 3, 11, 5};`

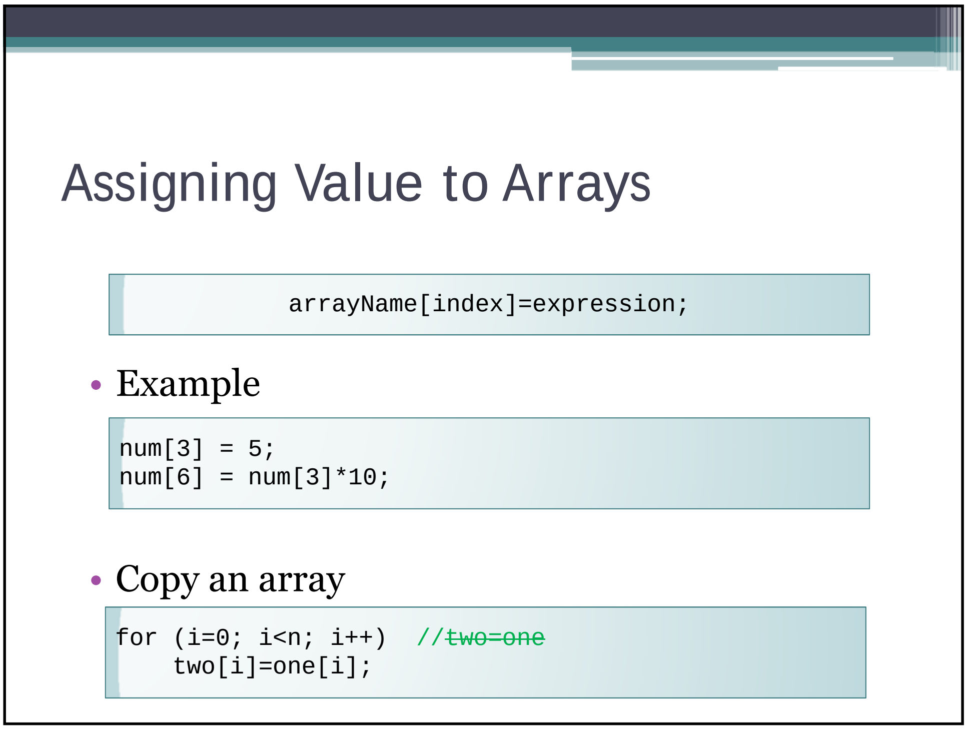
2	10	3	11	5
[0]	[1]	[2]	[3]	[4]



Accessing Array

```
arrayName[index]
```

- If num is an array
  - num[0] – Member of array num in position 0
  - num[1] – Member of array num in position 1



Assigning Value to Arrays

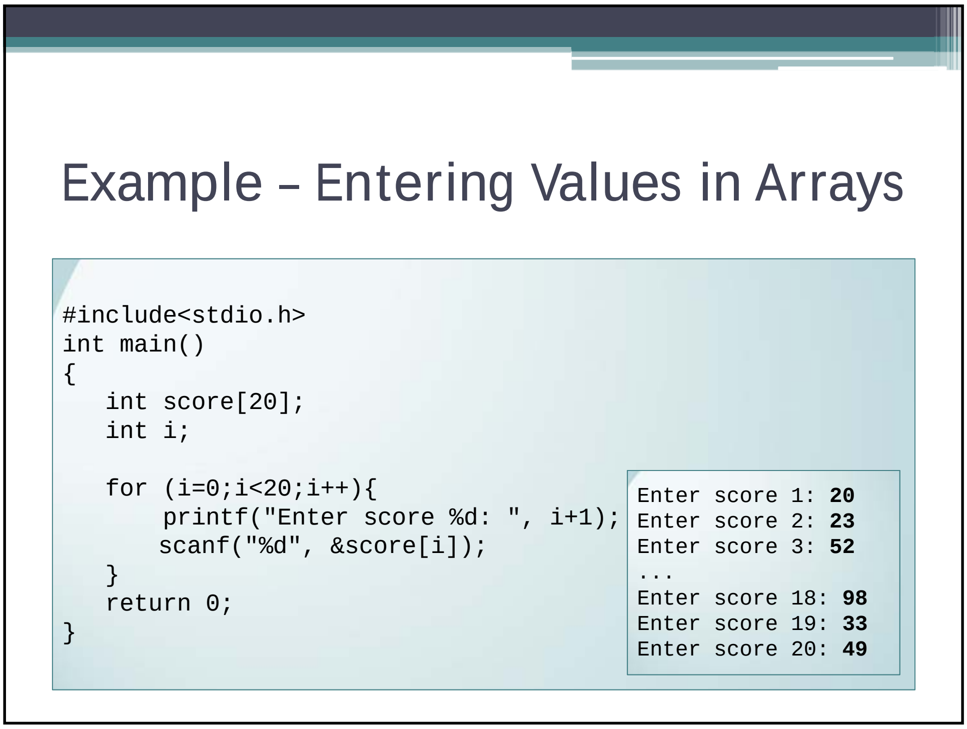
```
arrayName[index]=expression;
```

- Example

```
num[3] = 5;
num[6] = num[3]*10;
```

- Copy an array

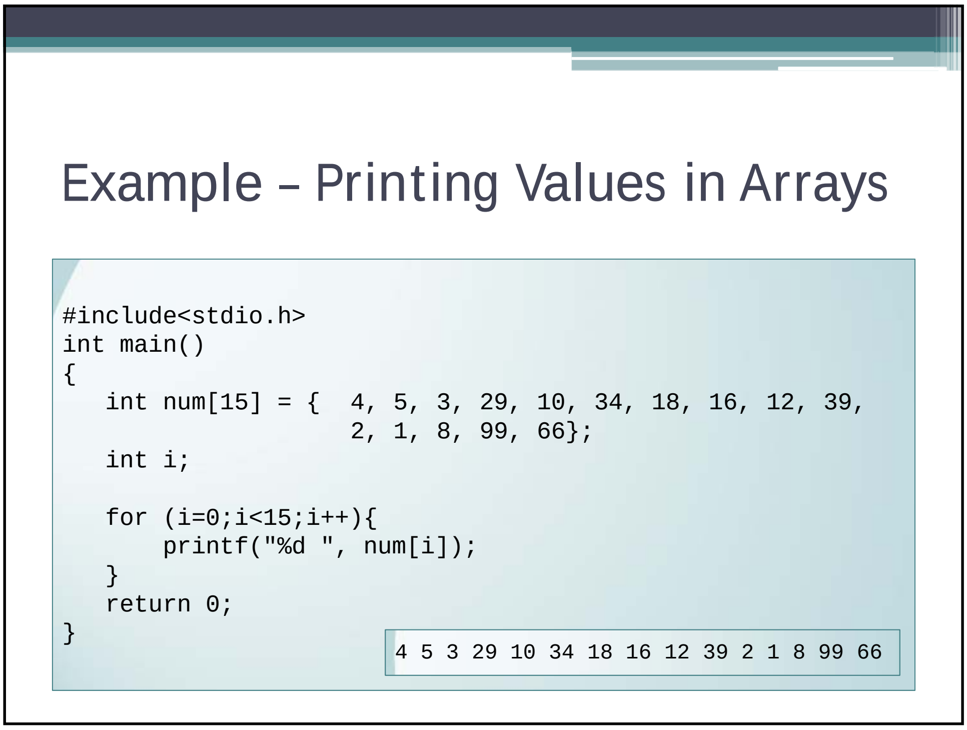
```
for (i=0; i<n; i++)  //~~two=one~~
    two[i]=one[i];
```



Example – Entering Values in Arrays

```
#include<stdio.h>
int main()
{
   int score[20];
   int i;

   for (i=0;i<20;i++){
       printf("Enter score %d: ", i+1);
       scanf("%d", &score[i]);
   }
   return 0;
}
```

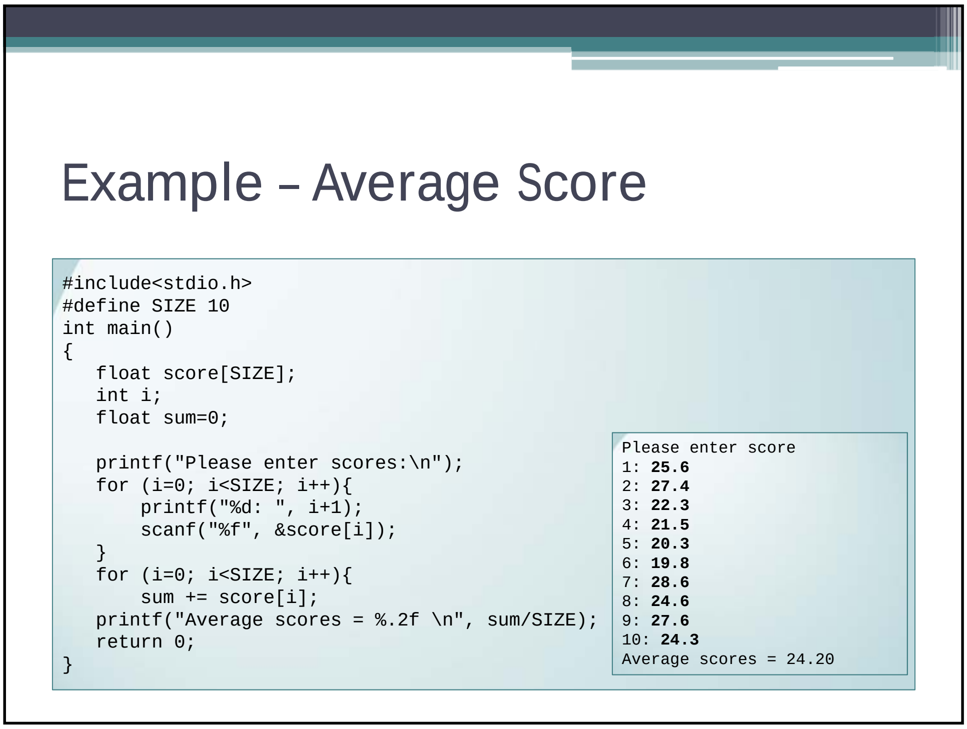
```
Enter score 1: 20
Enter score 2: 23
Enter score 3: 52
...
Enter score 18: 98
Enter score 19: 33
Enter score 20: 49
```



Example – Printing Values in Arrays

```
#include<stdio.h>
int main()
{
   int num[15] = {  4, 5, 3, 29, 10, 34, 18, 16, 12, 39,
                    2, 1, 8, 99, 66};
   int i;

   for (i=0;i<15;i++){
       printf("%d ", num[i]);
   }
   return 0;
}
```

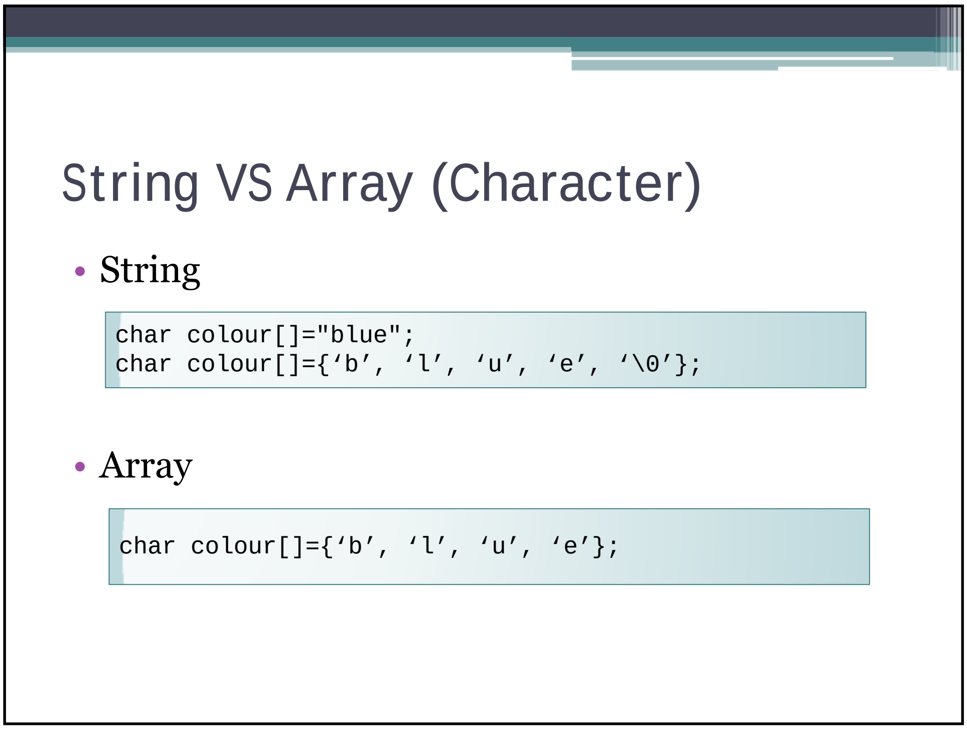
```
4 5 3 29 10 34 18 16 12 39 2 1 8 99 66
```



Example – Average Score

```
#include<stdio.h>
#define SIZE 10
int main()
{
   float score[SIZE];
   int i;
   float sum=0;

   printf("Please enter scores:\n");
   for (i=0; i<SIZE; i++){
       printf("%d: ", i+1);
       scanf("%f", &score[i]);
   }
   for (i=0; i<SIZE; i++){
       sum += score[i];
   printf("Average scores = %.2f \n", sum/SIZE);
   return 0;
}
```

```
Please enter score
1: 25.6
2: 27.4
3: 22.3
4: 21.5
5: 20.3
6: 19.8
7: 28.6
8: 24.6
9: 27.6
10: 24.3
Average scores = 24.20
```



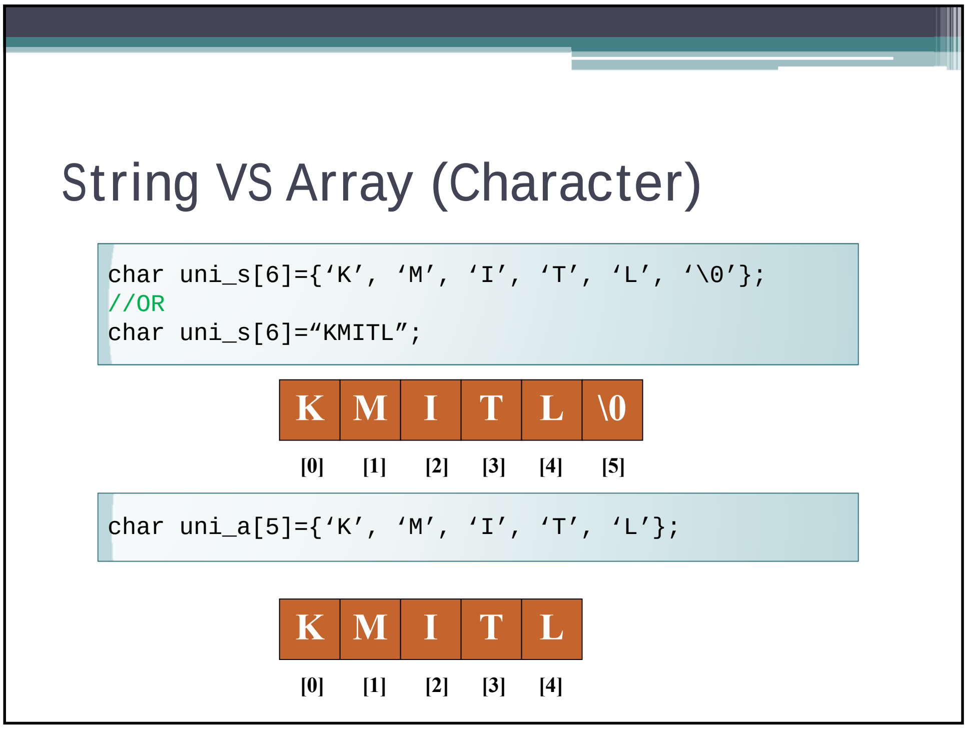
String VS Array (Character)

- String

```
char colour[]="blue";
char colour[]={‘b’, ‘l’, ‘u’, ‘e’, ‘\0’};
```

- Array

```
char colour[]={‘b’, ‘l’, ‘u’, ‘e’};
```



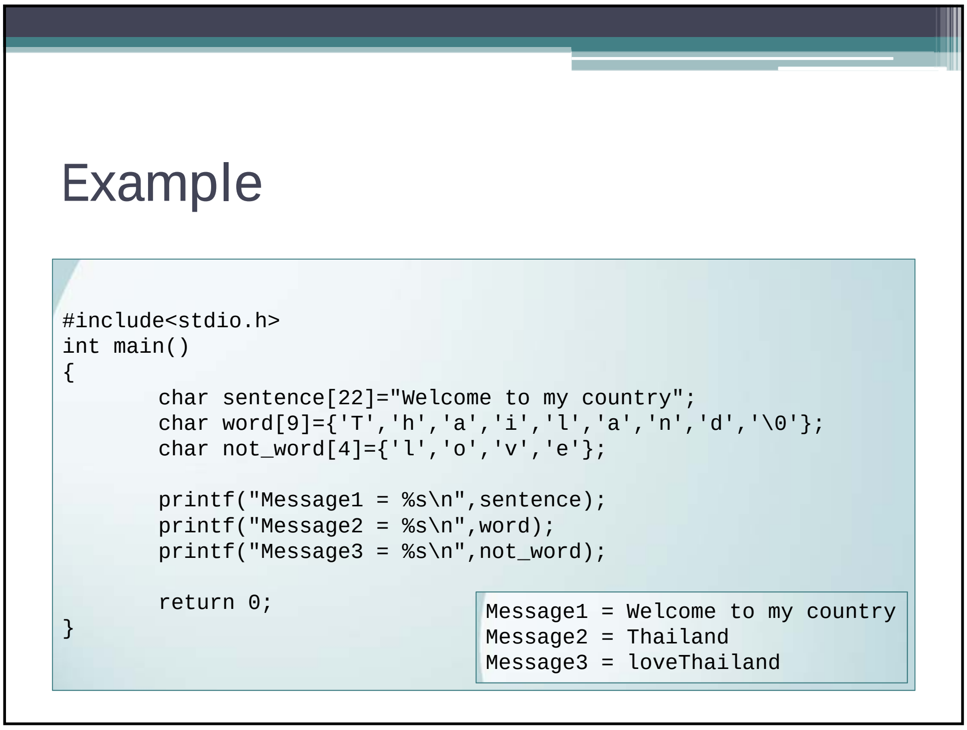
String VS Array (Character)

```
char uni_s[6]={‘K’, ‘M’, ‘I’, ‘T’, ‘L’, ‘\0’};
//OR
char uni_s[6]=“KMITL”;
```

K	M	I	T	L	\0
[0]	[1]	[2]	[3]	[4]	[5]

```
char uni_a[5]={‘K’, ‘M’, ‘I’, ‘T’, ‘L’};
```

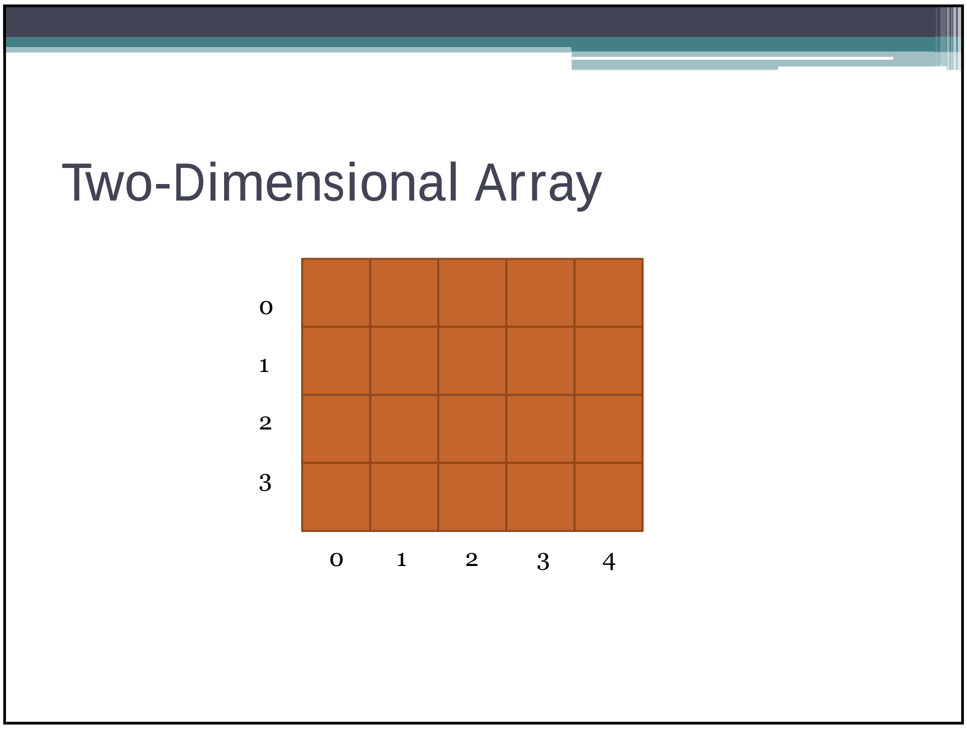
K	M	I	T	L
[0]	[1]	[2]	[3]	[4]



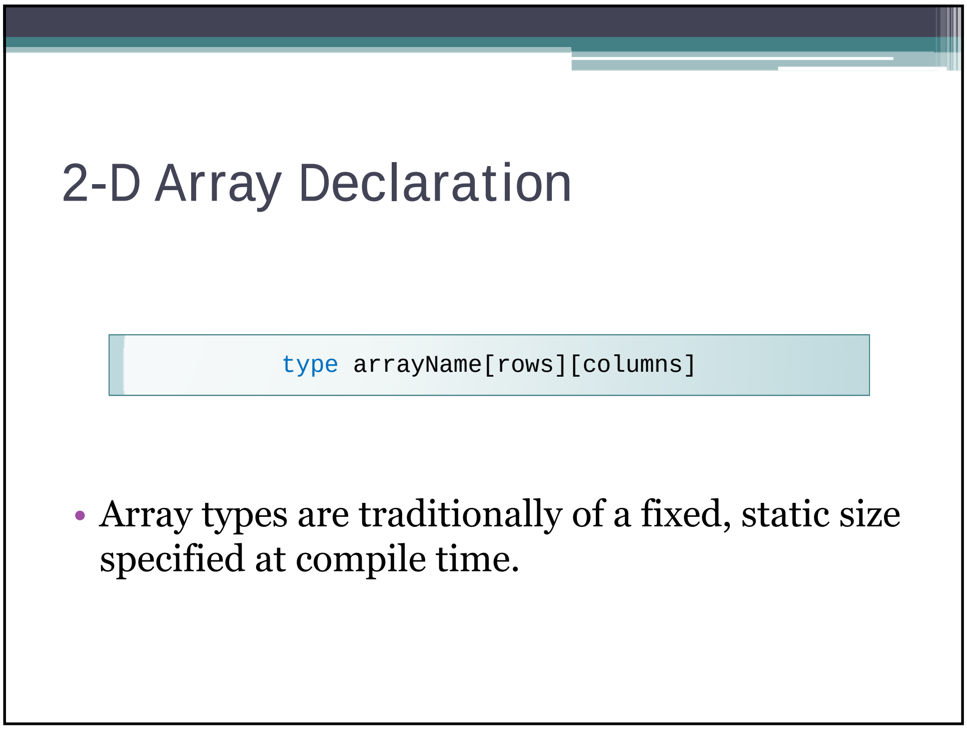
Example

```
#include<stdio.h>
int main()
{
        char sentence[22]="Welcome to my country";
        char word[9]={'T','h','a','i','l','a','n','d','\0'};
        char not_word[4]={'l','o','v','e'};

        printf("Message1 = %s\n",sentence);
        printf("Message2 = %s\n",word);
        printf("Message3 = %s\n",not_word);

        return 0;
}
```

```
Message1 = Welcome to my country
Message2 = Thailand
Message3 = loveThailand
```



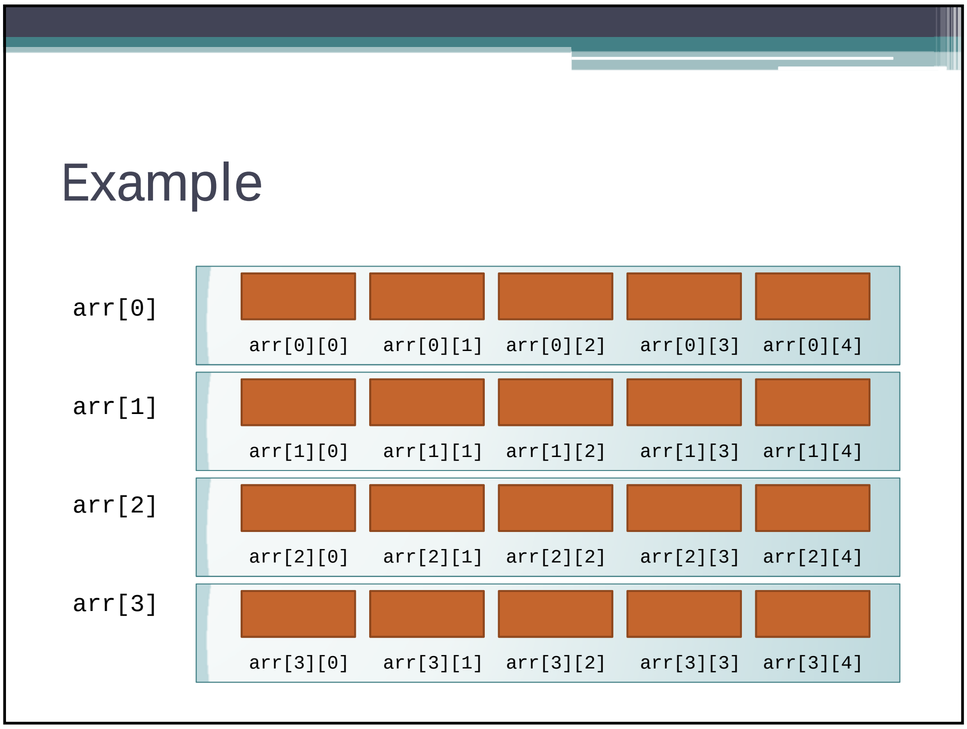
Two-Dimensional Array



2-D Array Declaration

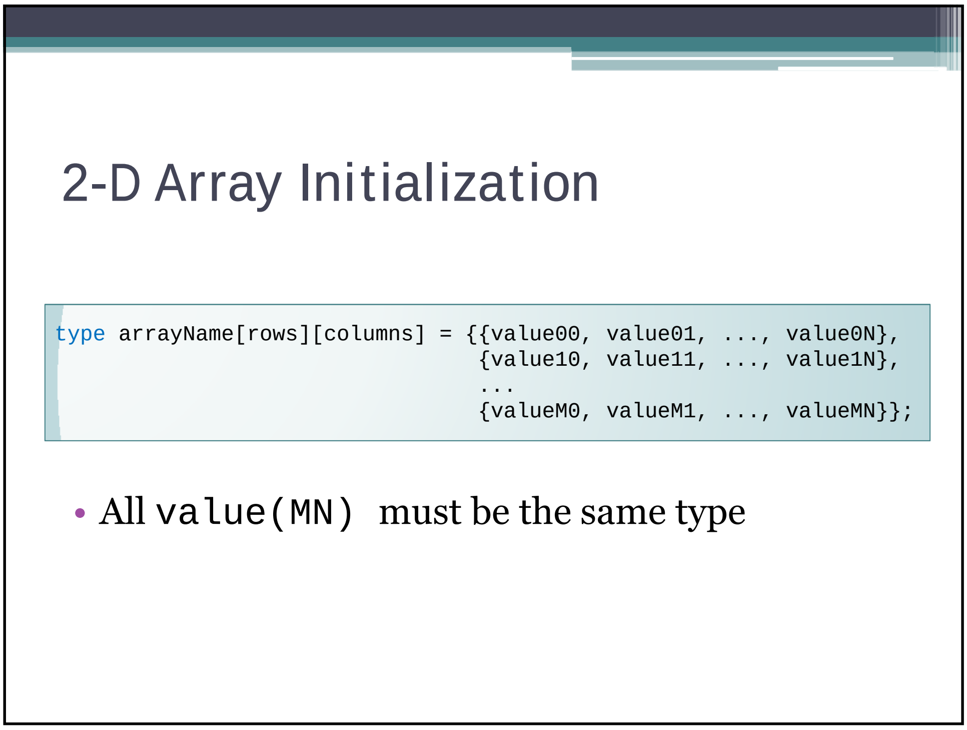
```
type arrayName[rows][columns]
```

- Array types are traditionally of a fixed, static size specified at compile time.



Example

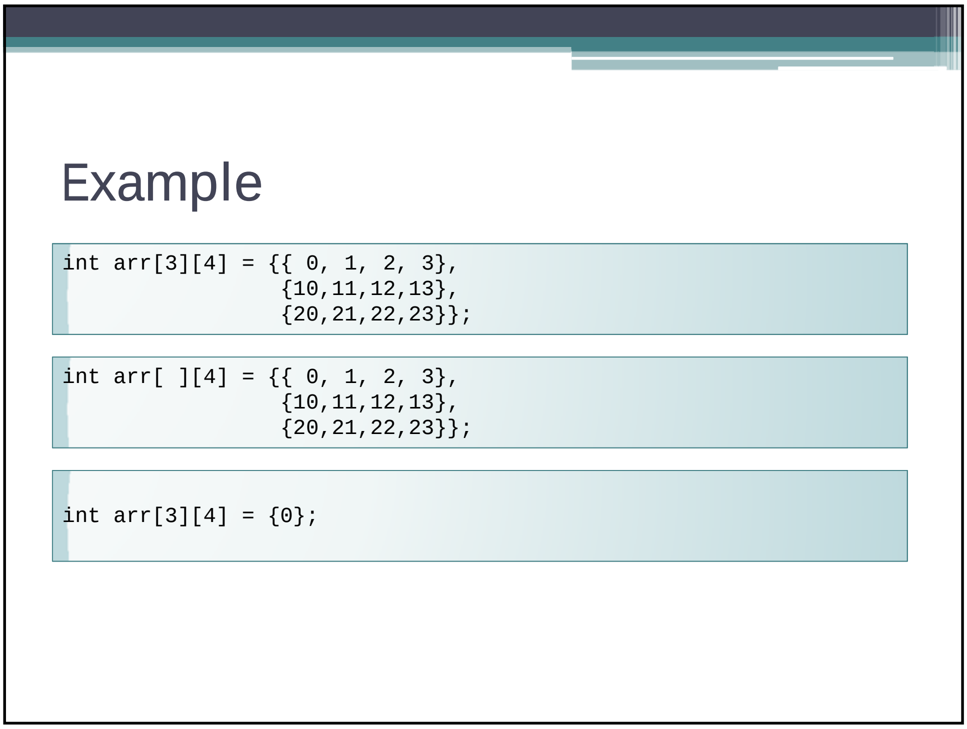
```
arr[0]   arr[0][0]  arr[0][1]  arr[0][2]  arr[0][3]  arr[0][4]
arr[1]   arr[1][0]  arr[1][1]  arr[1][2]  arr[1][3]  arr[1][4]
arr[2]   arr[2][0]  arr[2][1]  arr[2][2]  arr[2][3]  arr[2][4]
arr[3]   arr[3][0]  arr[3][1]  arr[3][2]  arr[3][3]  arr[3][4]
```



2-D Array Initialization

```
type arrayName[rows][columns] = {{value00, value01, ..., value0N},
                                 {value10, value11, ..., value1N},
                                 ...
                                 {valueM0, valueM1, ..., valueMN}};
```

- All `value(MN)` must be the same type

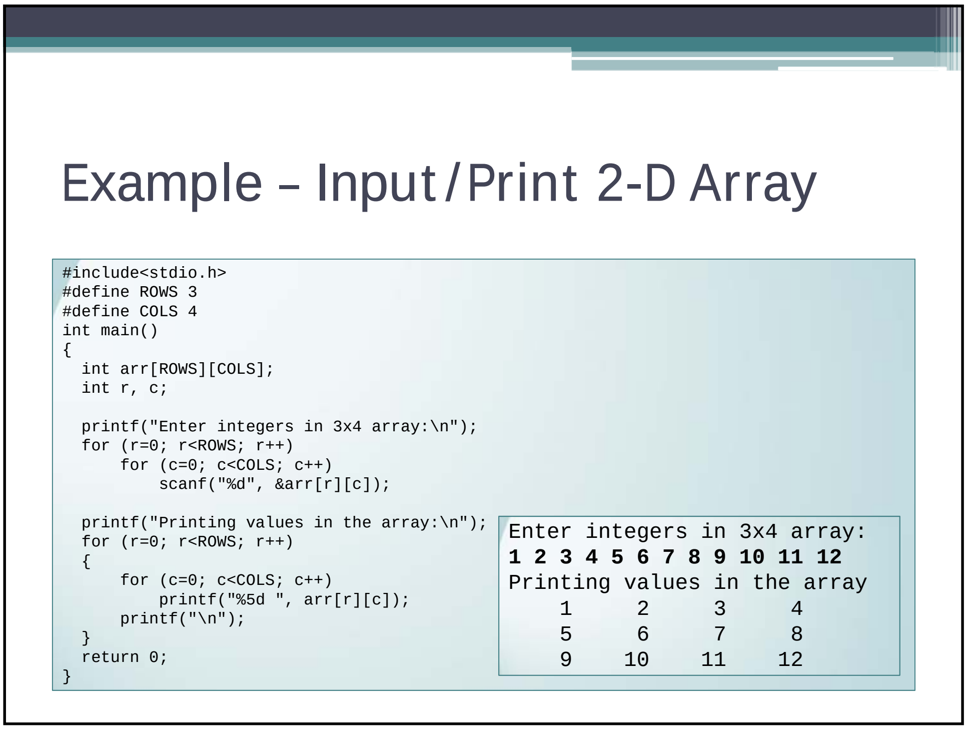


Example

```
int arr[3][4] = {{ 0, 1, 2, 3},
                 {10,11,12,13},
                 {20,21,22,23}};
```

```
int arr[ ][4] = {{ 0, 1, 2, 3},
                 {10,11,12,13},
                 {20,21,22,23}};
```

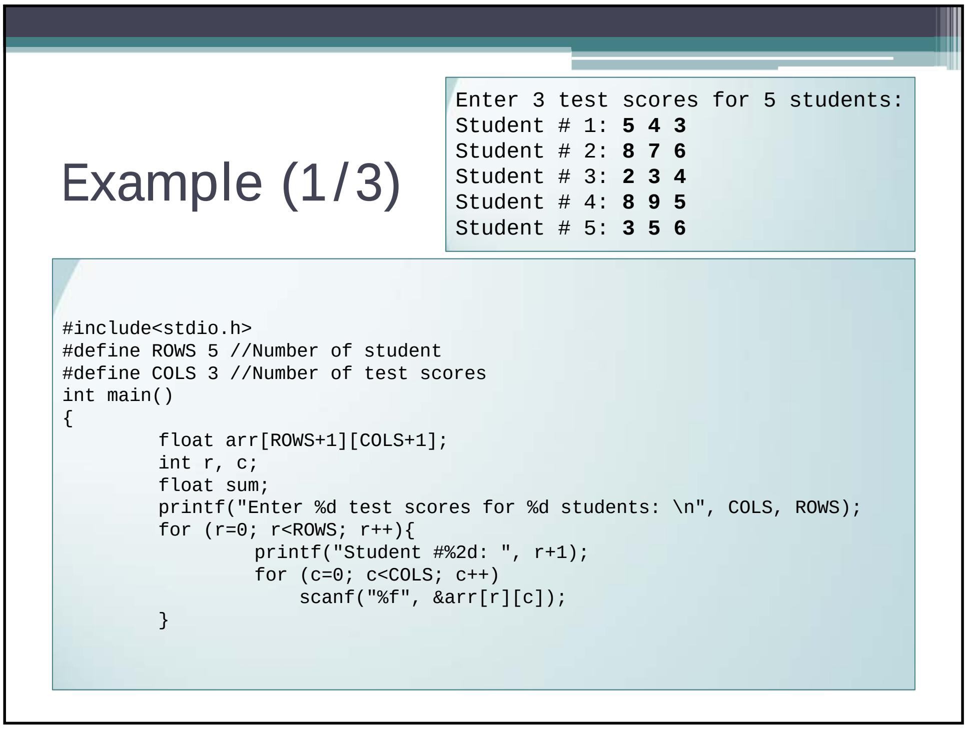
```
int arr[3][4] = {0};
```



Example – Input/Print 2-D Array

```
#include<stdio.h>
#define ROWS 3
#define COLS 4
int main()
{
  int arr[ROWS][COLS];
  int r, c;

  printf("Enter integers in 3x4 array:\n");
  for (r=0; r<ROWS; r++)
      for (c=0; c<COLS; c++)
          scanf("%d", &arr[r][c]);

  printf("Printing values in the array:\n");
  for (r=0; r<ROWS; r++)
  {
      for (c=0; c<COLS; c++)
          printf("%5d ", arr[r][c]);
      printf("\n");
  }
  return 0;
}
```

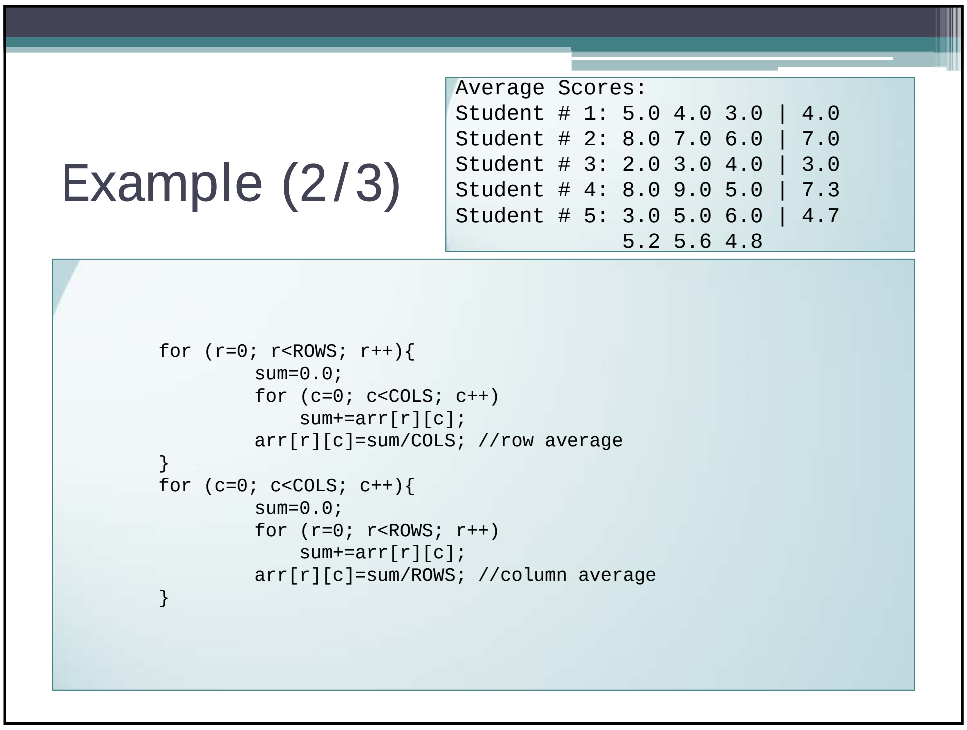
```
Enter integers in 3x4 array:
1 2 3 4 5 6 7 8 9 10 11 12
Printing values in the array
    1      2      3      4
    5      6      7      8
    9     10     11     12
```



Example (1/3)

```
Enter 3 test scores for 5 students:
Student # 1: 5 4 3
Student # 2: 8 7 6
Student # 3: 2 3 4
Student # 4: 8 9 5
Student # 5: 3 5 6
```

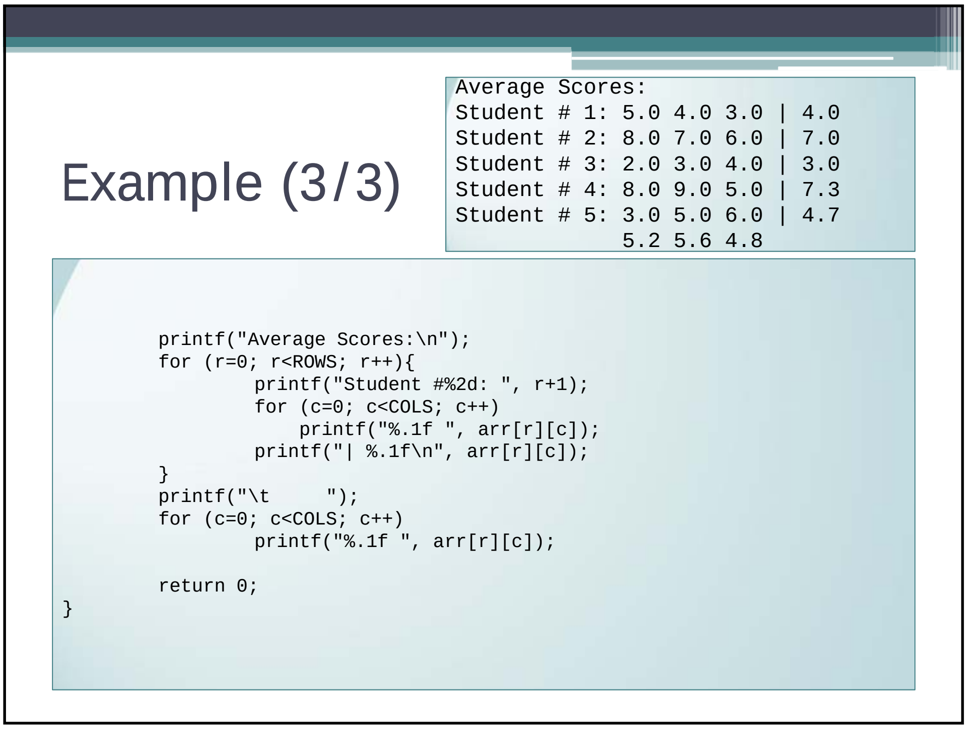
```
#include<stdio.h>
#define ROWS 5 //Number of student
#define COLS 3 //Number of test scores
int main()
{
        float arr[ROWS+1][COLS+1];
        int r, c;
        float sum;
        printf("Enter %d test scores for %d students: \n", COLS, ROWS);
        for (r=0; r<ROWS; r++){
                printf("Student #%2d: ", r+1);
                for (c=0; c<COLS; c++)
                    scanf("%f", &arr[r][c]);
        }
```



Example (2/3)

```
Average Scores:
Student # 1: 5.0 4.0 3.0 | 4.0
Student # 2: 8.0 7.0 6.0 | 7.0
Student # 3: 2.0 3.0 4.0 | 3.0
Student # 4: 8.0 9.0 5.0 | 7.3
Student # 5: 3.0 5.0 6.0 | 4.7
             5.2 5.6 4.8
```

```
        for (r=0; r<ROWS; r++){
                sum=0.0;
                for (c=0; c<COLS; c++)
                    sum+=arr[r][c];
                arr[r][c]=sum/COLS; //row average
        }
        for (c=0; c<COLS; c++){
                sum=0.0;
                for (r=0; r<ROWS; r++)
                    sum+=arr[r][c];
                arr[r][c]=sum/ROWS; //column average
        }
```



Example (3/3)

```
Average Scores:
Student # 1: 5.0 4.0 3.0 | 4.0
Student # 2: 8.0 7.0 6.0 | 7.0
Student # 3: 2.0 3.0 4.0 | 3.0
Student # 4: 8.0 9.0 5.0 | 7.3
Student # 5: 3.0 5.0 6.0 | 4.7
             5.2 5.6 4.8
```

```
        printf("Average Scores:\n");
        for (r=0; r<ROWS; r++){
                printf("Student #%2d: ", r+1);
                for (c=0; c<COLS; c++)
                    printf("%.1f ", arr[r][c]);
                printf("| %.1f\n", arr[r][c]);
        }
        printf("\t     ");
        for (c=0; c<COLS; c++)
                printf("%.1f ", arr[r][c]);

        return 0;
}
```