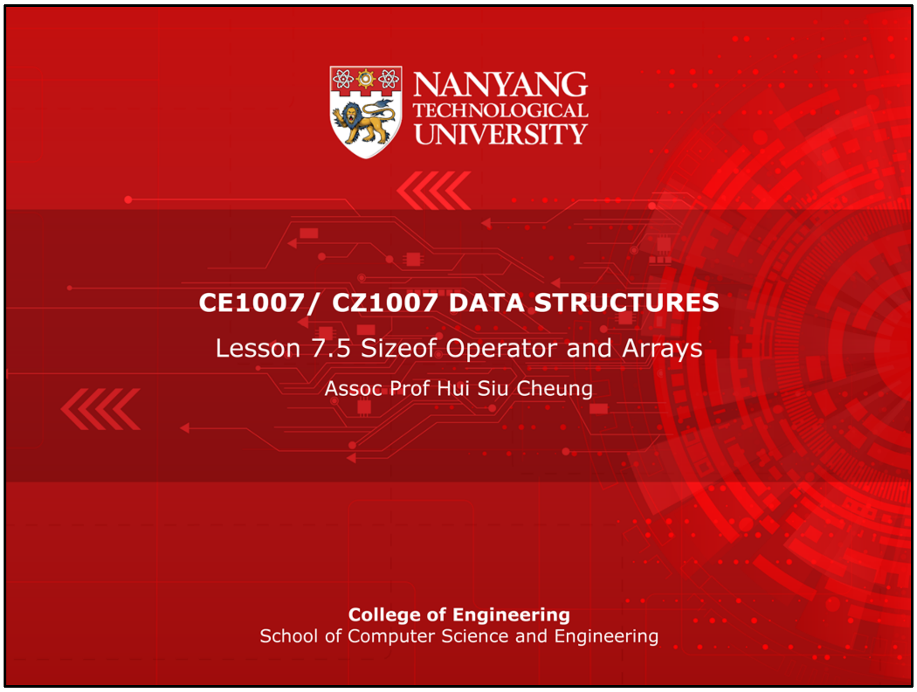


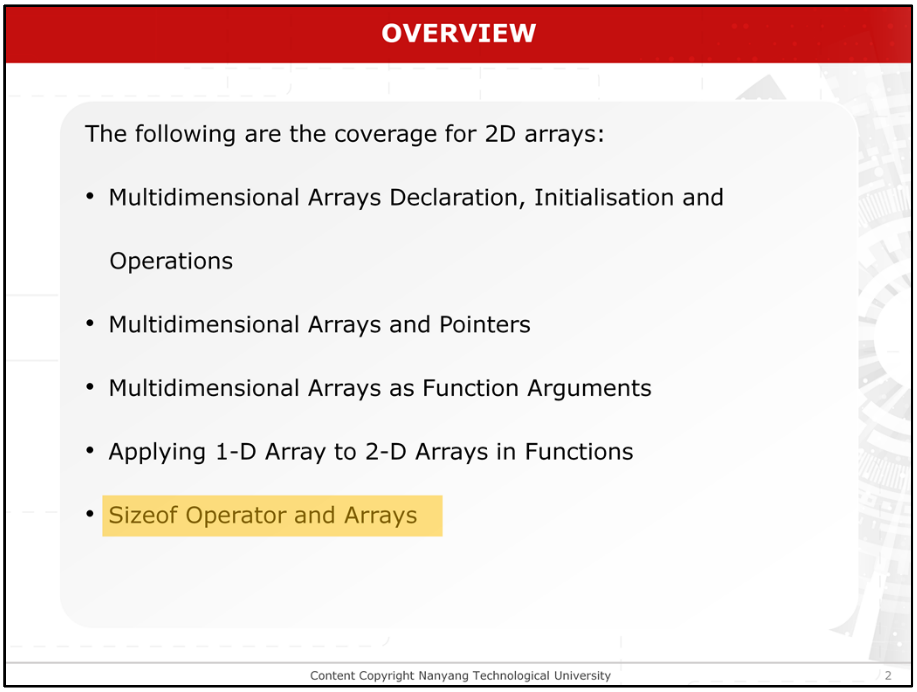
NANYANG TECHNOLOGICAL UNIVERSITY

CE1007/ CZ1007 DATA STRUCTURES

Lesson 7.5 Sizeof Operator and Arrays

Assoc Prof Hui Siu Cheung

College of Engineering
School of Computer Science and Engineering



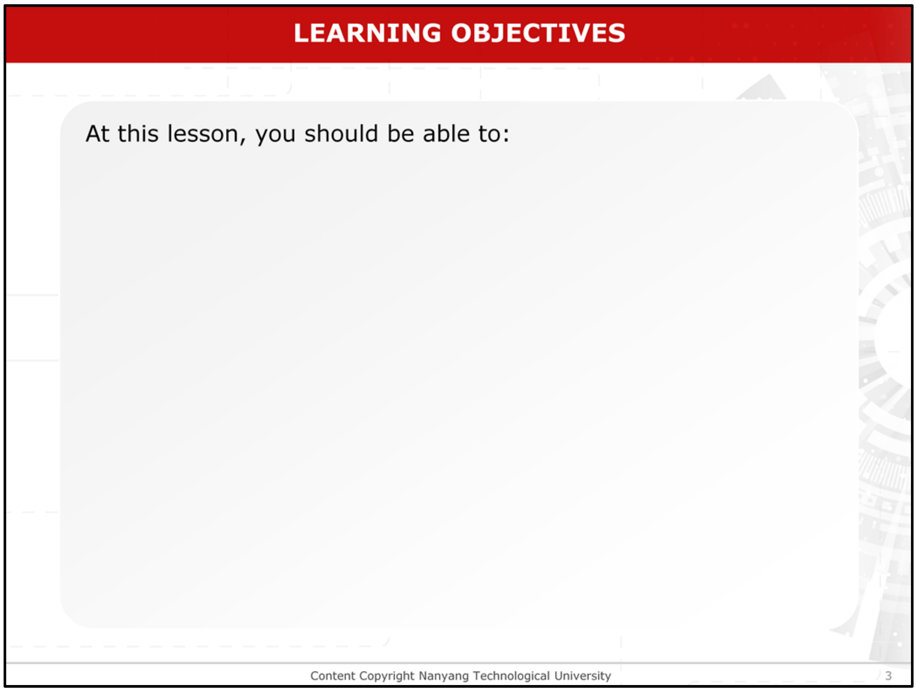
OVERVIEW

The following are the coverage for 2D arrays:

- Multidimensional Arrays Declaration, Initialisation and Operations
- Multidimensional Arrays and Pointers
- Multidimensional Arrays as Function Arguments
- Applying 1-D Array to 2-D Arrays in Functions
- Sizeof Operator and Arrays

2

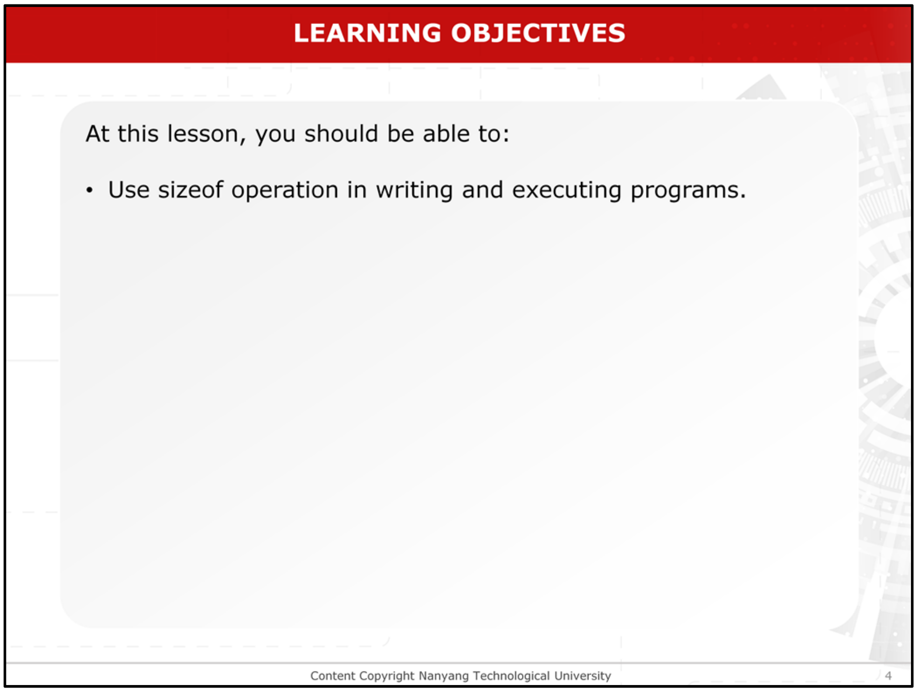
The following are the coverage for 2 dimensional ARRAYS. this video focuses on size of operator and arrays



LEARNING OBJECTIVES

At this lesson, you should be able to:

LEARNING OBJECTIVES: At this lesson, you should be able to:

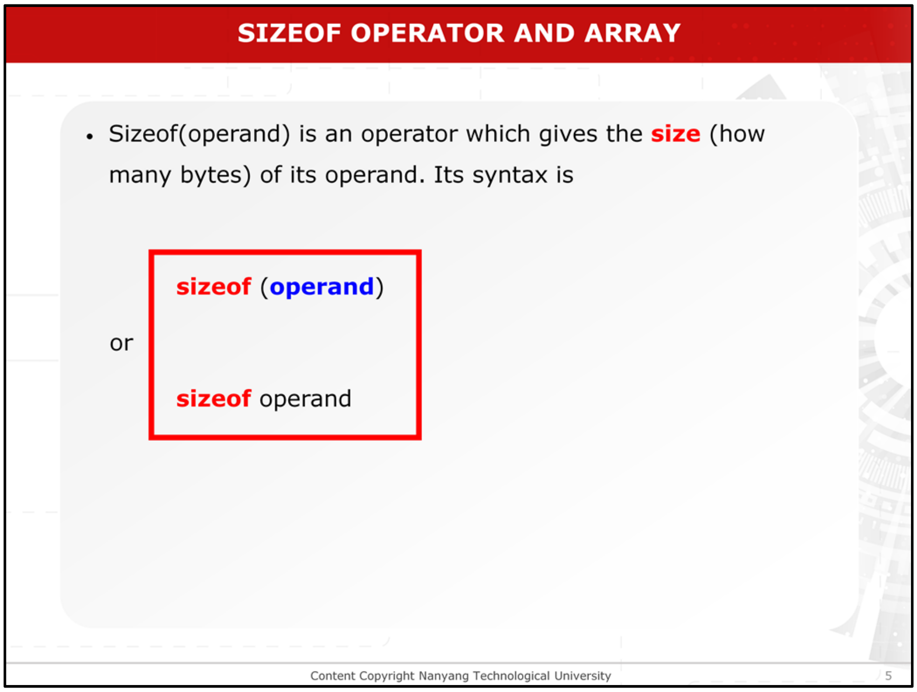


LEARNING OBJECTIVES

At this lesson, you should be able to:

- Use sizeof operation in writing and executing programs.

Use size of operation in writing and executing programs.



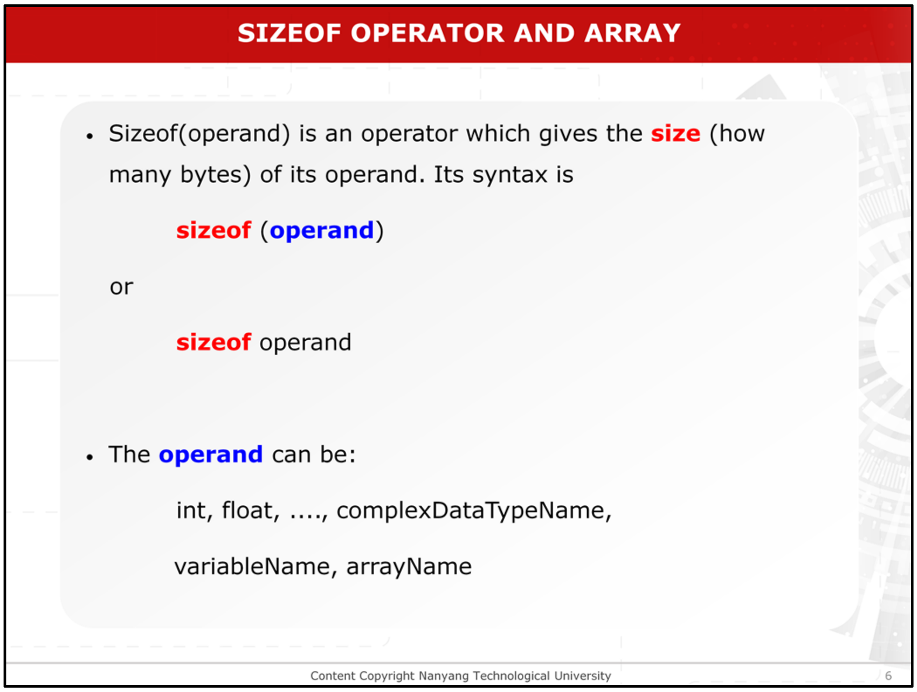
SIZEOF OPERATOR AND ARRAY

- Sizeof(operand) is an operator which gives the **size** (how many bytes) of its operand. Its syntax is

```
sizeof (operand)
or
sizeof operand
```

Sizeof Operator and Array

Size of is an operator which gives the size (in bytes) of its operand. Its syntax is as shown. The **operand** can either be a type enclosed in parenthesis or an expression.



SIZEOF OPERATOR AND ARRAY

- Sizeof(operand) is an operator which gives the **size** (how many bytes) of its operand. Its syntax is

  sizeof (**operand**)

  or

  sizeof operand

- The **operand** can be:

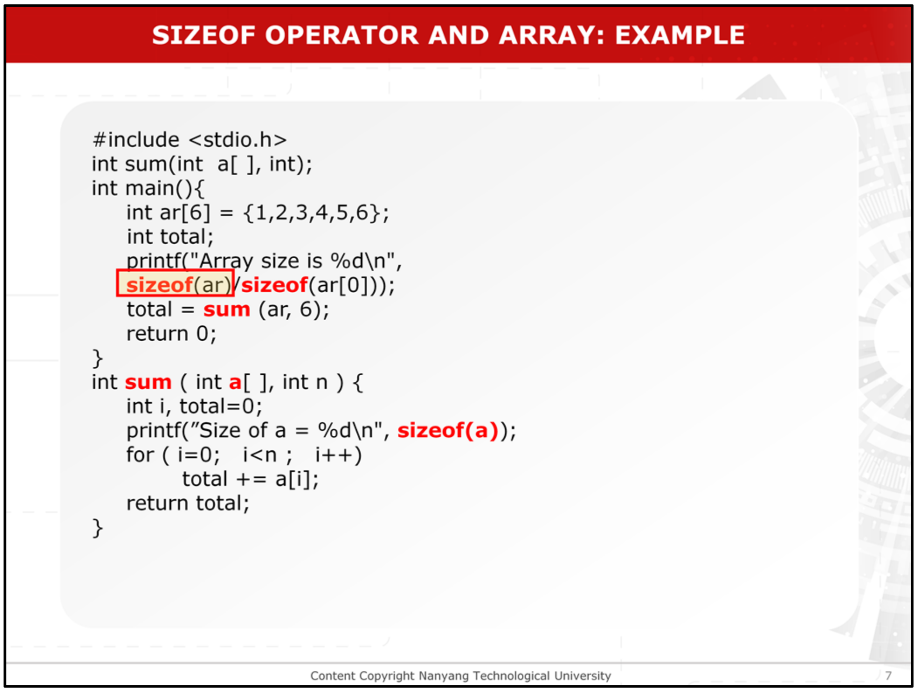
  int, float, …., complexDataTypeName,

  variableName, arrayName

Content Copyright Nanyang Technological University

6

We can also use it with arrays.



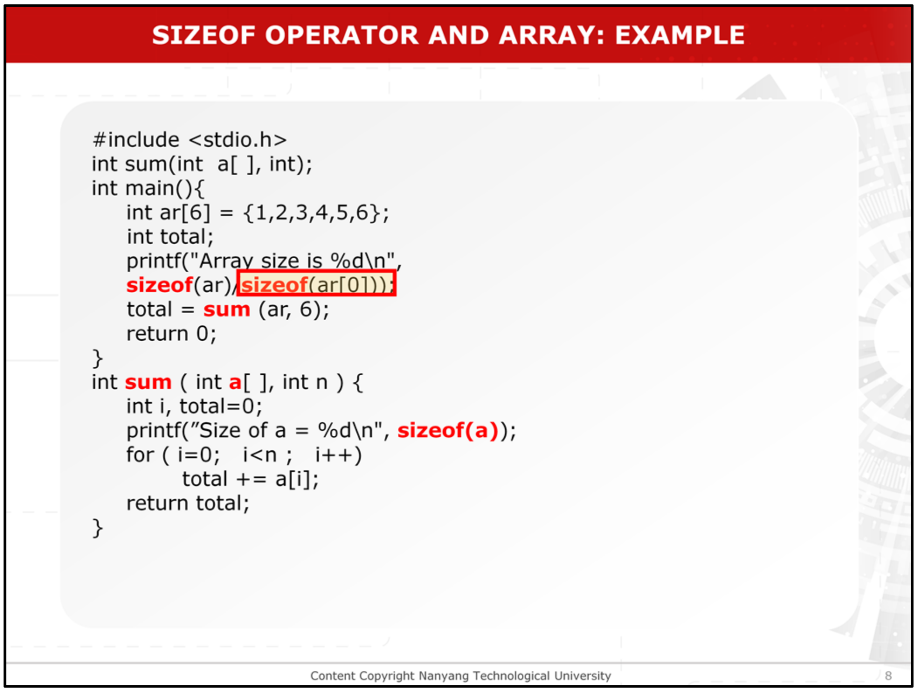
SIZEOF OPERATOR AND ARRAY: EXAMPLE

```
#include <stdio.h>
int sum(int  a[ ], int);
int main(){
    int ar[6] = {1,2,3,4,5,6};
    int total;
    printf("Array size is %d\n",
    sizeof(ar)/sizeof(ar[0]));
    total = sum (ar, 6);
    return 0;
}
int sum ( int a[ ], int n ) {
    int i, total=0;
    printf(”Size of a = %d\n", sizeof(a));
    for ( i=0;   i<n ;   i++)
          total += a[i];
    return total;
}
```

7

Sizeof Operator and Array: Example

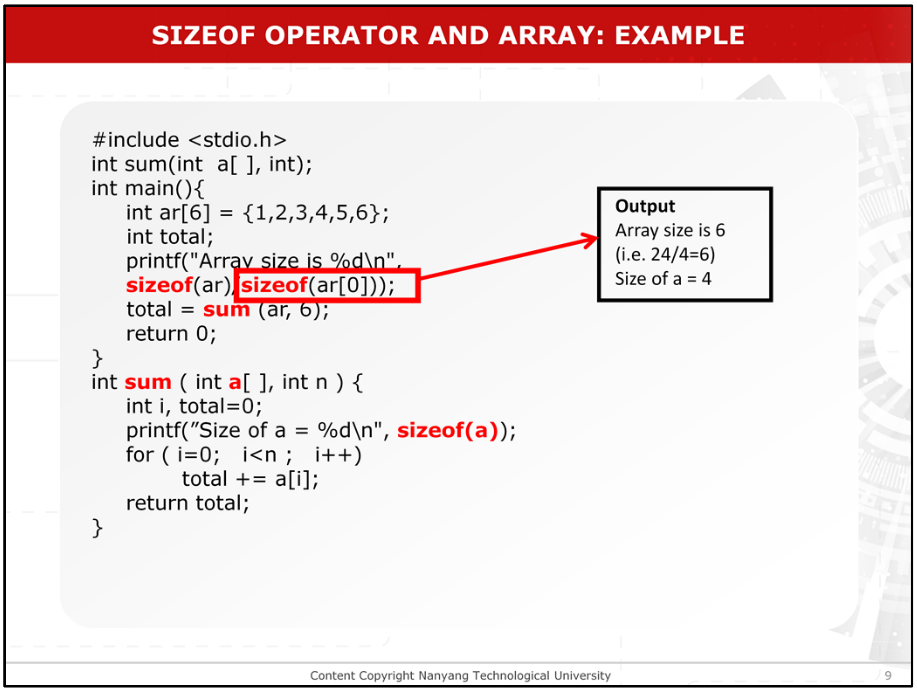
In the **main** function of the program, the **size of** operator returns the number of bytes of the array.



SIZEOF OPERATOR AND ARRAY: EXAMPLE

```
#include <stdio.h>
int sum(int  a[ ], int);
int main(){
    int ar[6] = {1,2,3,4,5,6};
    int total;
    printf("Array size is %d\n",
    sizeof(ar)/sizeof(ar[0]));
    total = sum (ar, 6);
    return 0;
}
int sum ( int a[ ], int n ) {
    int i, total=0;
    printf(”Size of a = %d\n", sizeof(a));
    for ( i=0;   i<n ;   i++)
          total += a[i];
    return total;
}
```

The second **size of** operator returns the number of bytes of each element in the array.

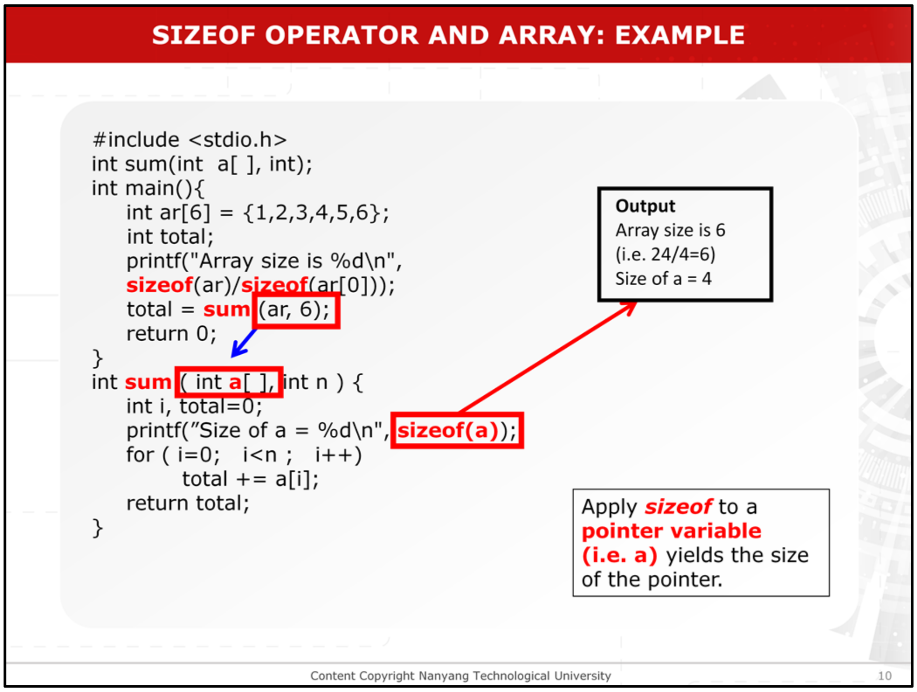


SIZEOF OPERATOR AND ARRAY: EXAMPLE

```
#include <stdio.h>
int sum(int  a[ ], int);
int main(){
    int ar[6] = {1,2,3,4,5,6};
    int total;
    printf("Array size is %d\n",
    sizeof(ar)/sizeof(ar[0]));
    total = sum (ar, 6);
    return 0;
}
int sum ( int a[ ], int n ) {
    int i, total=0;
    printf(”Size of a = %d\n", sizeof(a));
    for ( i=0;   i<n ;   i++)
          total += a[i];
    return total;
}
```

Output
Array size is 6
(i.e. 24/4=6)
Size of a = 4

Therefore, the number of elements can be calculated by dividing the size of the array by the size of each element in the array. In this case, the array size is 24 by 6 which gives the value of 6.



SIZEOF OPERATOR AND ARRAY: EXAMPLE

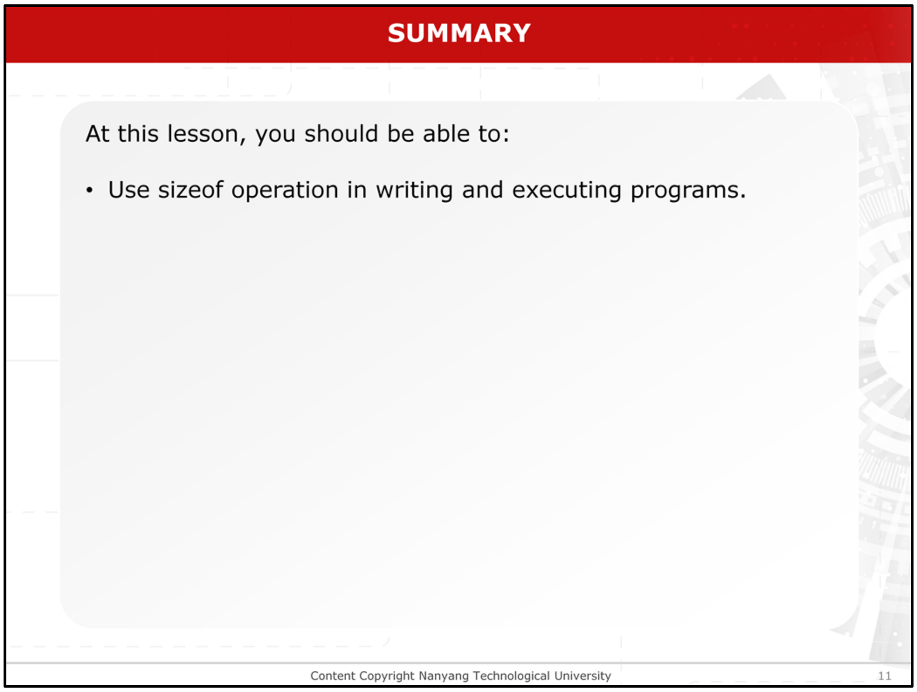
```
#include <stdio.h>
int sum(int  a[ ], int);
int main(){
    int ar[6] = {1,2,3,4,5,6};
    int total;
    printf("Array size is %d\n",
    sizeof(ar)/sizeof(ar[0]));
    total = sum(ar, 6);
    return 0;
}
int sum( int a[ ], int n ) {
    int i, total=0;
    printf(”Size of a = %d\n", sizeof(a));
    for ( i=0;   i<n ;   i++)
          total += a[i];
    return total;
}
```

Output
```
Array size is 6
(i.e. 24/4=6)
Size of a = 4
```

Apply ***sizeof*** to a **pointer variable (i.e. a)** yields the size of the pointer.

Content Copyright Nanyang Technological University 10

However, in the function **sum**, the **size of** operator returns the number of bytes for the array **a**. It is in fact a pointer which contains the address of the argument passed in from the calling function. As a pointer has 4 bytes, the size of **a** is 4.



SUMMARY

At this lesson, you should be able to:

- Use sizeof operation in writing and executing programs.

Content Copyright Nanyang Technological University 11

In summary, after viewing this video lesson, your should be able to do the listed.