Object-Oriented Programming in C++

Arrays and Vectors

Professor: 최학남, xncui@inha.ac.kr
Office: high-tech 401

Contents

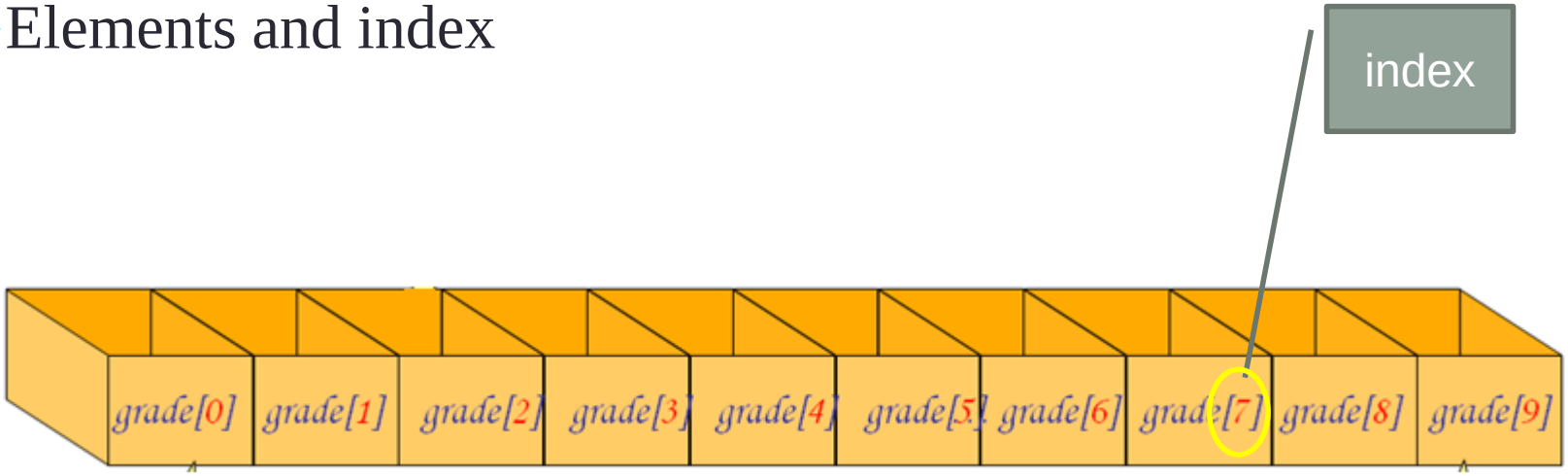
- What are arrays
- A simple program using array
  - ✓ Array declaration
  - ✓ Array initialization
- 1D-Array
- 2D-Array
- ND-Array

What are arrays

- Array: store more than one value at a time in a single variable.
  - ✓ Suppose we wish to arrange the scores obtained by 30 students. In such a case we have two options to store these scores in memory:
    - Construct 30 variables to store scores obtained by 30 different students, i.e. each variable containing one student's score.
    - Construct one variable (called array or subscripted variable) capable of storing or holding all the 30 values.

What are arrays

- Elements and index

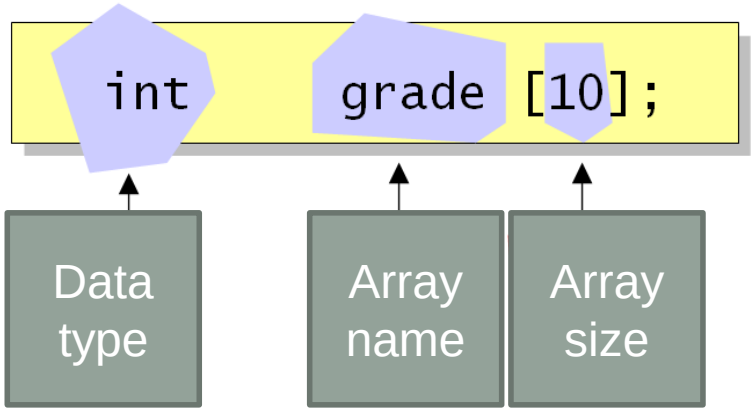


0th element

9th element

What are arrays

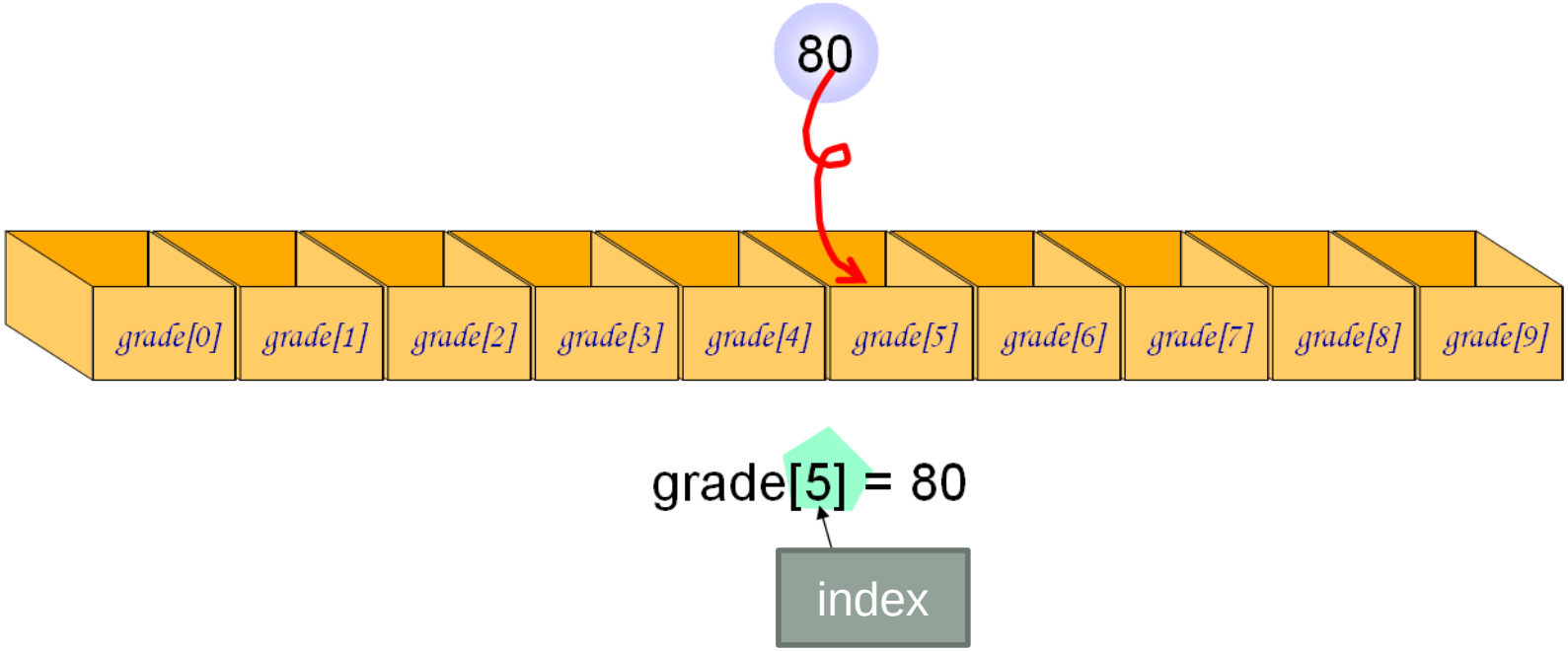
- Declaration
  - ✓The index from 0 not 1.




```
int  score[60];            // #of 60 elements, data type: int, array name: grade
float  cost[12];           // #of 12elements, data type: float, array name: cost
char  name[50];            // #of 50elements, data type: char, array name: name
char  src[10], dst[10];    // declare the two char type arrays simultaneously
int   index, days[7];      // declare the variable and array simultaneously
```

What are arrays

- Access the array elements




```
grade[5] = 80;
grade[1] =  grade[0];
grade[i] = 100;          // i is integer variable
grade[i+2] = 100;         // index can be expression.
grade[index[3]] = 100;   // index[]is integer array
```

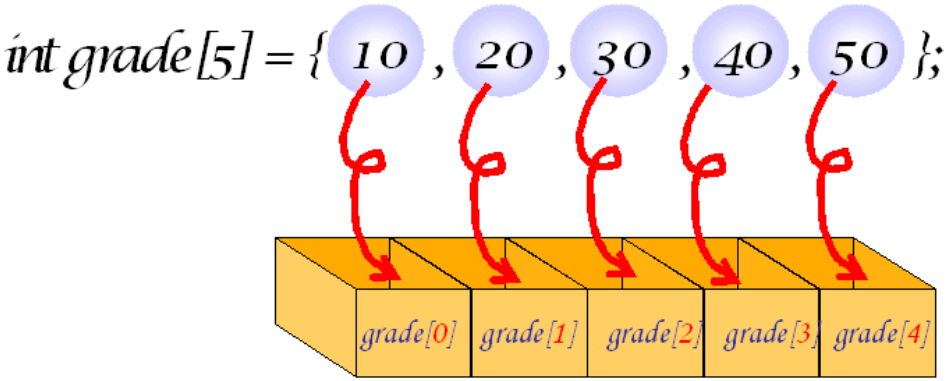
Array initialization

- Array initialization
  - ✓Till the array elements are not given any specific values, they are supposed to contain garbage values.

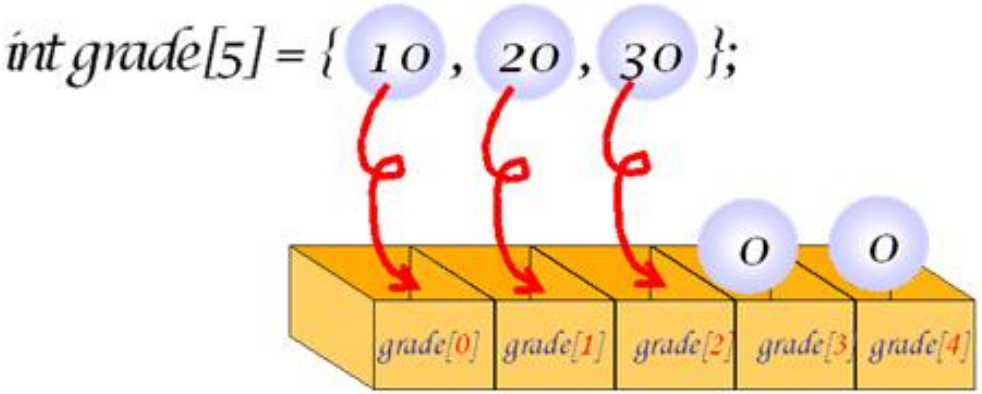
```
int num[6] = { 2, 4, 12, 5, 45, 5 } ;
int n[ ] = { 2, 4, 12, 5, 45, 5 } ;
float press[ ] = { 12.3, 34.2 -23.4, -11.3 } ;
```

Array initialization

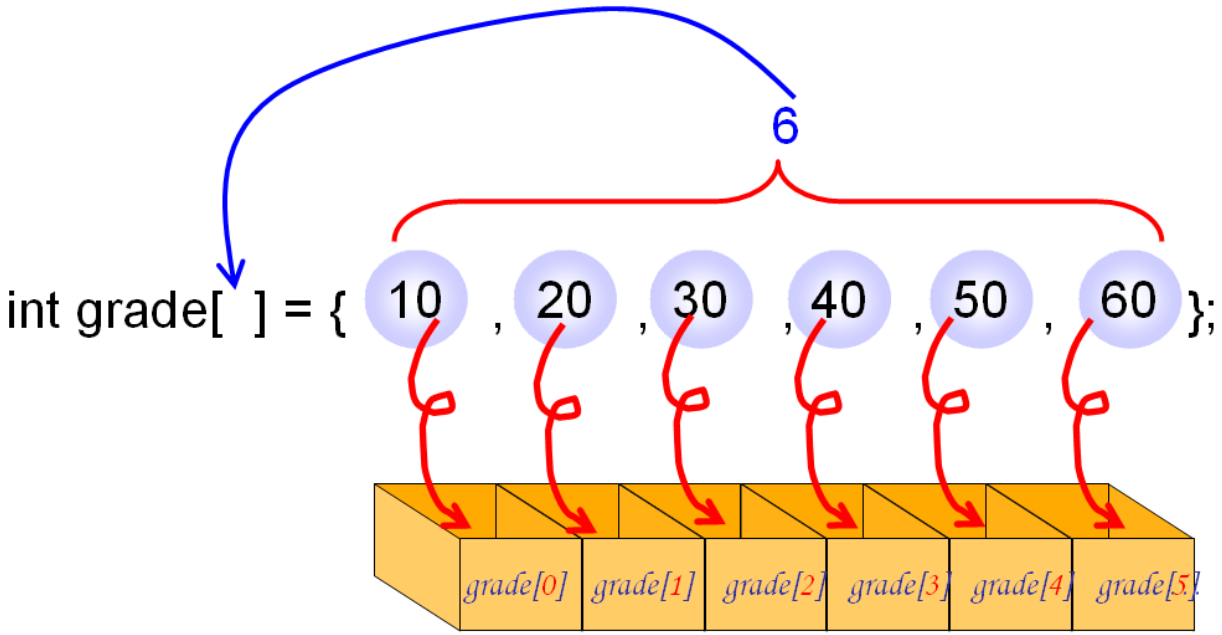
- int grade[5] = { 10,20,30,40,50 };



- int grade[5] = { 10,20,30 };



Array initialization



A simple program using array

```
1  # include <stdio.h>
2  ////////////////////////////////////////////////////////////ex2
3  void main( )
4  {
5      int avg, sum = 0 ;
6      int i ;
7      int scores[30] ; /* array declaration */
8      for ( i = 0 ; i <= 29 ; i++ )
9      {
10         printf ( "\nEnter scores " ) ;
11         scanf ( "%d", &scores[i] ) ; /* store data in array */
12     }
13     for ( i = 0 ; i <= 29 ; i++ )
14     {
15         sum = sum + scores[i] ; /* read data from an array*/
16         avg = sum / 30 ;
17     }
18     printf ( "\nAverage score = %d", avg ) ;
19 }
```

A simple program using array

- Entering data into an array

```
8     for ( i = 0 ; i <= 29 ; i++ )
9     {
10        printf ( "\nEnter scores " ) ;
11        scanf ( "%d", &scores[i] ) ; /* store data in array */
12    }
```

A simple program using array

- Reading data from an array
  - ✓ An array is a collection of similar elements.
  - ✓ The first element in the array is numbered 0, so the last element is 1 less than the size of the array.
  - ✓ An array is also known as a subscripted variable.
  - ✓ Before using an array its type and dimension must be declared.
  - ✓ However big an array its elements are always stored in contiguous memory locations. This is a very important point which we would discuss in more detail later on.

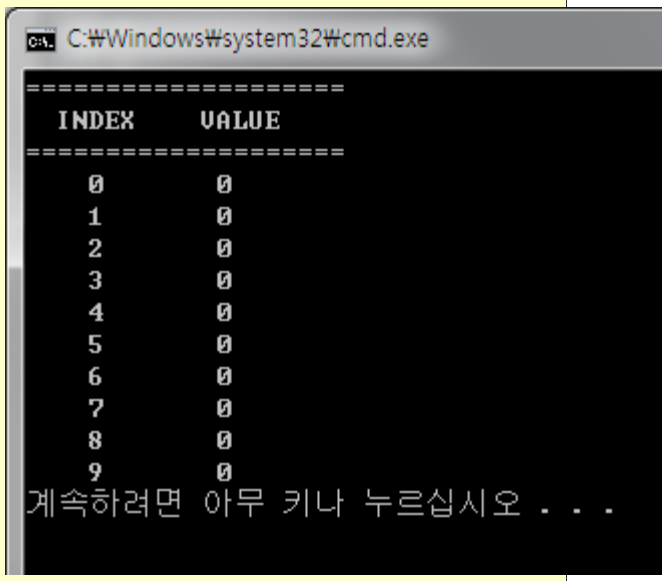
```
13      for ( i = 0 ; i <= 29 ; i++ )
14      {
15          sum = sum + scores[i] ; /* read data from an array*/
16          avg = sum / 30 ;
17      }
```

A simple program using array

- Array declaration
  - ✓ int scores[30]
    - The elements of this array are
      - scores[0], scores[1],…, scores[29]
      - Note that the array elements are denoted by [ ], to distinguish it from function which uses( );

EX #1 Declaration

```
#include <stdio.h>
int main(void)
{
        int grade[10];
        int i;
        for(i = 0; i < 10; i++)
                grade[i] = 0;
        printf("====================\n");
        printf("  INDEX    VALUE\n");
        printf("====================\n");
        for(i = 0; i < 10; i++)
                printf("%5d    %5d\n", i, grade[i]);
        return 0;
}
```



EX #2 Initialization

```c
#include <stdio.h>
int main(void)
{
    int grade[10]={ 31, 63, 62, 87, 14, 25, 92, 70, 75, 53 };
    int i;

    printf("======================\n");
    printf("  INDEX    VALUE\n");
    printf("======================\n");
    for(i = 0; i < 10; i++)
        printf("%5d    %5d\n", i, grade[i]);
    return 0;
}
```

```
======================
  INDEX    VALUE
======================
    0       31
    1       63
    2       62
    3       87
    4       14
    5       25
    6       92
    7       70
    8       75
    9       53
계속하려면 아무 키나 누르십시오 . . .
```

EX #3 Array elements

```c
#include <stdio.h>

#define STUDENTS 5

int main(void)
{
	int grade[STUDENTS];
	int sum = 0;
	int i, average;

	for(i = 0; i < STUDENTS; i++)
	{
		printf("enter the scores: ");
		scanf("%d", &grade[i]);
	}

	for(i = 0; i < STUDENTS; i++)
		sum += grade[i];

	average = sum / STUDENTS;
	printf("average score= %d\n", average);
	return 0;
}
```

```
enter the scores: : 10
enter the scores: : 20
enter the scores: : 30
enter the scores: : 40
enter the scores: : 50
average score = 30
```

EX #5 Access the wrong index

```c
#include <stdio.h>
#define SIZE 5

int main(void)
{
	int array[SIZE] = {1, 2, 3, 4, 5};
	int i;

	for(i = 0; i <= SIZE; i++)
		printf("array[%d]	%d\n", i, array[i]);

	return 0;
}
```

```
array[0]	1
array[1]	2
array[2]	3
array[3]	4
array[4]	5
array[5]	1245120
```

EX #6 Array copy

```
int grade[SIZE];
int score[SIZE];

score = grade;              // wrong !
```

```
#include <stdio.h>
#define SIZE 5

int main(void)
{
        int i;
        int a[SIZE] = {1, 2, 3, 4, 5};
        int b[SIZE];

        for(i = 0; i < SIZE; i++)
                b[i] = a[i];

        return 0;
}
```

True way

EX #7 Array comparison

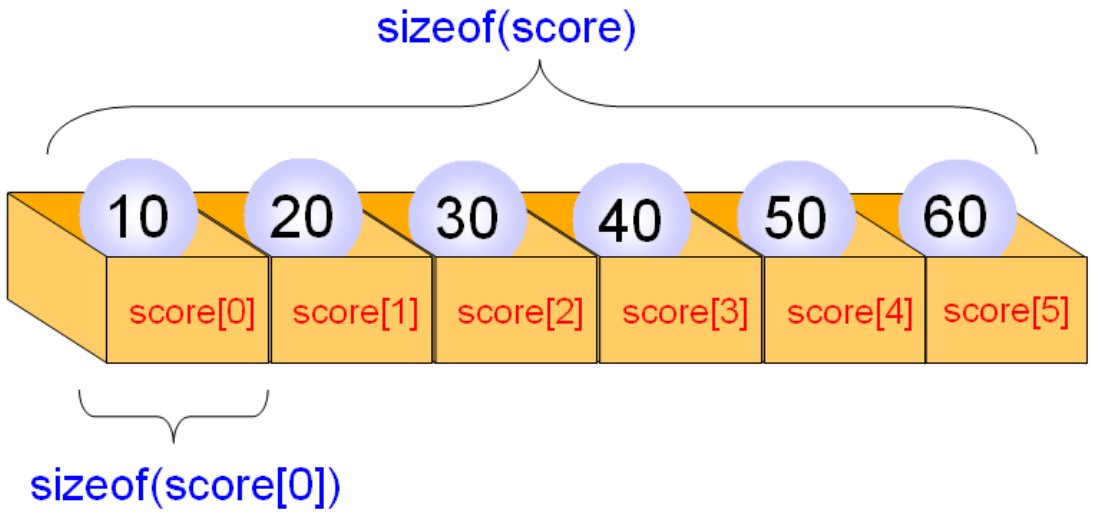
```
#include <stdio.h>
#define SIZE 5

int main(void)
{
	int i;
	int a[SIZE] = { 1, 2, 3, 4, 5 };
	int b[SIZE] = { 1, 2, 3, 4, 5 };

	if( a == b )		// ① wrong way to comparison of the array
		printf("wrong reuslts.\n");
	else
		printf(" wrong reuslts.\n");


	for(i = 0; i < SIZE ; i++) // ② true way
	{
		if ( a[i] != b[i] )
		{
			printf("a[] != b[]\n");
			return 0;
		}
	}
	printf("a[]=b[]\n");
	return 0;
}
```

EX #8 Array size




```
int grade[] = { 1, 2, 3, 4, 5, 6 };
int i, size;

size = sizeof(grade) / sizeof(grade[0]);

for(i = 0; i < size ; i++)
        printf("%d ", grade[i]);
```

Compute the size of array automatically

EX #9 Minimum value

```
#include <stdio.h>
#define SIZE 10

int main(void)
{
	int grade[SIZE];
	int i, min;

	for(i = 0; i < SIZE; i++)
	{
		printf("enter the scores: ");
		scanf("%d", &grade[i]);
	}

	min = grade[0];

	for(i = 1; i < SIZE; i++)
	{
		if( grade[i] < min )
			min = grade[i];
	}

	printf("minimum value is %d.\n", min);
	return 0;
}
```

```
enter the scores : 50
enter the scores : 40
enter the scores : 30
enter the scores : 20
enter the scores : 10
enter the scores : 20
enter the scores : 30
enter the scores : 40
enter the scores : 60
enter the scores : 70
minimum value is 10.
```

EX #10 Function and Array

```
#include <stdio.h>
#define SIZE 7

void square_array(int a[], int size);
void print_array(int a[], int size);
void square_element(int e);

int main(void)
{
        int list[SIZE] = { 1, 2, 3, 4, 5, 6, 7 } ;

        print_array(list, SIZE);
        square_array(list, SIZE);
        print_array(list, SIZE);

        printf("%3d\n", list[6]);
        square_element(list[6]);
        printf("%3d\n", list[6]);

        return 0;
}
```

```
void square_array(int a[], int size)
{
        int i;

        for(i = 0; i < size; i++)
                a[i] = a[i] * a[i];
}
void square_element(int e)
{
        e = e * e;
}
void print_array(int a[], int size)
{
        int i;

        for(i = 0; i < size; i++)
                printf("%3d ", a[i]);
        printf("\n");
}
```

The function parameter refer the original array if the parameter is an array