جامعة الملك سعود
King Saud University

Arrays

Lecture 11

Outline

- What is an Array
- Declaring and Using Arrays
- Declaring and Referencing Arrays
- Declaring and Using Arrays
- Reading from & Writing to Arrays
- Arrays in Memory
- What can we do with Arrays
- Initializing Arrays

What is an Array?

- A Collection of Data all of which is of the same type

Array of Numbers

Index	Score
0	100
1	75
2	80
3	66

Array of Letters

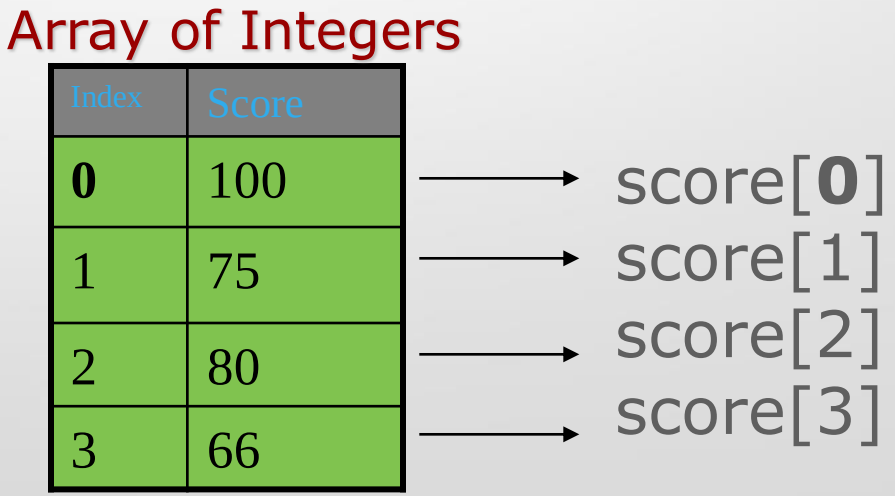
Index	Grade
0	A
1	C
2	B
3	D

Declaring and Using Arrays

- Declaration
  - Syntax:

    Type_name Array_Name[Declared_size]
  - Example:

    int score[4];



Declaring Arrays

Syntax

datatype name [size]; /* not initialized */

datatype name [size] = {initialization list} /* initialized */

optional

Examples

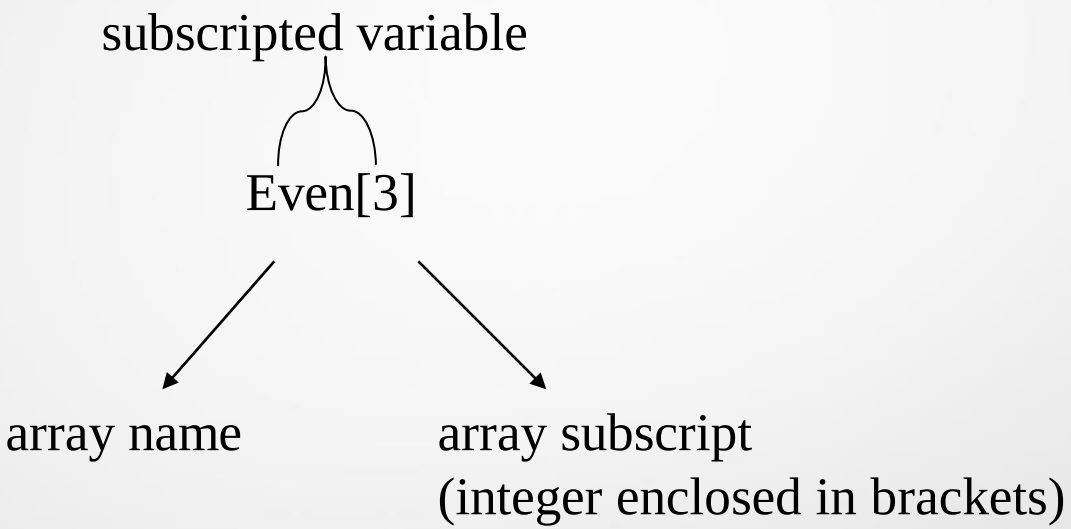
int even[5];

int even [5] = {2,4,6,8,10};

int even[5], num=0, x[10], y;

double j[] = {1.0,4.2,6.5,8.1,10.4}, d, k[3];

Referencing Arrays



Examples

even[3] ⟶ read as: even sub three

x[0] ⟶ read as: x sub zero

Declaring and Referencing Arrays

Int even[5]; → even[0] *subscripts*

even[1] *always:*

even[2] *starts from 0*

even[3] *until size-1*

even[4]

Int even[] = {2,4,6,8,10}; → even[0] = 2

even[1] = 4

even[2] = 6

even[3] = 8

even[4] = 10

Exercise 1

Char grades [5];

- *How many memory cells are reserved in memory?*

  5 memory cells

- *What type of data will be stored there?*

  character

- *How do we refer to the first array element?*

  grades[0]

- *How fo we refer to the last array element?*

  grades[4]

Exercise 2

- *Declare one array for storing the square roots of the integers*
- *from 0 through 10.*

Double sr[11];

- *Declare one array for storing the cubes of the same integers.*

Int cb[11];

Declaring and Using Arrays

- Using & Referencing Arrays
  - Elements of the array are called indexed variables
  - Can be used anyplace that an ordinary variable of the same

    type is used

Read: cin >> score[2];

Write: cout << score[1];

Assign: score[3]= score[1] + score[2];

score[student_id] = score[2];

score

Index	Score
0	100
1	75
2	80
3	66

Reading from & Writing to Arrays

- For (i=0; i<size, i++)
  cout << , x[i];

- For (i=0; i<=size-1, i++)
  cin >> x[i];

- For (i=0; i<size, i++)
  x[i] = i *5;

Example 1

```
/* program to compute the sum and the sum of the squares of all
data elements in an array */

#include <iostream>
#include <cmath>
 using namespace std;
int main (void)
{ int x[5] = {1,2,3,4,5}, i, sum=0, sum_sqr=0;
   for (i=0; i<5; i++)
         { sum+=x[i];
            sum_sqr+=pow(x[i],2); }
   cout << "The sum of the array elements are“ <<sum << endl;
   cout << "The sum of their squares are “ << sum_sqr << endl;
   return 0;}
```

Example 2

```
/* program that reads 10 integers and prints them in
reverse order */
#include<iostream>
 using namespace std;
int main()
{
 int number [10], i;

 for (i=0; i<=9; i++)
      cin >> number[i]);

 for (i=9; i>=0; i--)
      cout <<  number[i]);
 return(0);
}
```

Example 3

```
/* Program that calculates the price of four items */

#include<iostream>
 using namespace std;
int main()
{ int i, quantity;
   float total=0, price[4]={5.77, 3.15, 2.50, 1.35};
   for (i=0; i<=3; i++)
         {
           cout << "Enter the quantity of item : “ << i << endl;
           cin >> quantity; /*read quantity of item i */
           total+=quantity*price[i];
         }
   cout << "The total cost is “<<total <<endl;
   return 0;}
```

Example 4

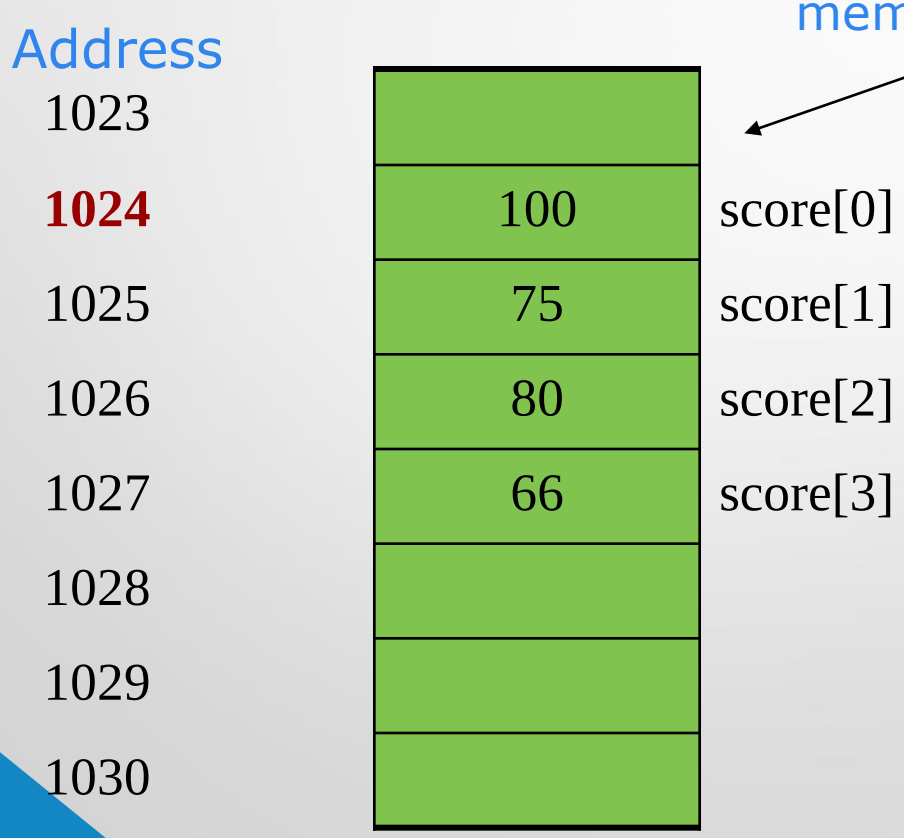
```
/* Program that determines the maximum element in an array */

#include <iostream>
 using namespace std;
int main()
{
 int x [5] = {12, 5, 21, 6, 4},i,max;
 max = x[0]; /* assume that max is the first element */
 for (i=1; i<=4; i++) /* finding the maximum */
     if (x[i] > max)
             max = x[i];
 cout << "The maximum is“ << max << endl;
 return 0;
}
```

Arrays in Memory

- Contiguous locations:

```
int score[4];
```



score

Index	Score
0	100
1	75
2	80
3	66

What can we do with Arrays?

- Read/Write
- Initialize
- Search
- Sort
- Manipulate values

Initializing Arrays

- int score_w[4] = {60, 60, 60, 60};
- char score_x[4] = {‘A’, ‘A’,’A’,’A’};
- int score_y[4] = {100, 100};
- int score_z[] = {100, 100, 100};

score_w

Index	Score
0	60
1	60
2	60
3	60

score_x

Index	Score
0	A
1	A
2	A
3	A

score_y

Index	Score
0	100
1	100
2	0
3	0

score_z

Index	Score
0	100
1	100
2	100

Thank You