Unit V – Part 01
(Array in C)

Lecture Slide

AS2022

Objectives

By the end of this session, students will be able to:

- Define the concept of array
- Determine how one dimensional array is declared and initialized
- Explain the concept of two dimensional array
- Discuss how 2-d array is declared and initialized
- Describe multi-dimensional array
- Differentiate between static and dynamic arrays
- Implement Passing array to function

Array

- One of the derived data types
- **Array** is a *fixed-size* sequenced collection of elements of the same data type
- It is also classified as one of the data structures in C
- Elements in an array shares common name

  Example: salary[10], salary [9], ….,salary[1]
- Individual values are **elements** and complete set is **array**
- ***Index*** or ***subscript*** in square bracket after the array name is used to access the individual elements in an array

  example: salary [**10**]

Array

- Array enables us to develop concise and efficient program
- The usage of array is not limited to representing simple list of values
- Types of arrays
  - One-dimensional arrays
  - Two-dimensional arrays
  - Multi-dimensional arrays

One-Dimensional array

- Can be also referred as '*single-subscripted variable*'
- List of items can be given one name using only one subscript
- The single scripted variable x_i can be expressed as

  x[0], x[1], x[2], ……., x[n]
- The *i* refers to the i^{th} element in an array
- The subscript of an array can be integer constants, integer variable or expression that yields integer
- *C performs **no bounds checking** and care should be exercise to ensure that the array indices are within the declared limits*

Declaring 1-D array

- Must be declared before their usage in the program
- General form of array declaration is

```
type variable_name[size]
```

- Consider an array to store list of number (2,3,1,4), the array can be declared as

```
int number[4]
```

- **Note:** When declaring character arrays, we must allow one extra element space for the null terminator

Initialization of 1-D array

- Contains garbage if it is not initialized after declaring
- Can be initialized at either of the following stages
  - Compile time
  - Run time
- **Compile time**
  - General form

```
type array_name[size]={list of values};
```

  - Example:

```
int number[3] ={1,2,3};
char name[10]={'j', 'o', 'e'};
```

Compile Initialization

- If the number of values in the list is less than the size of an array, then only that many elements will be initialized
- The remaining elements will be set to ***zero*** automatically incase of **float** and **int** type and **NULL** for **char** type
- The size may be omitted in such case
- It won't work if the number of initializer is more than the declared size
  - Example: `int number[2]={2,4,2};`

Runtime Initialization

- An array can be explicitly initialized
- Using *scanf( )* to get from user keyboard
- Example

```
int i, number[10];
for(i=0;i<10;i++){
   scanf("%d",&number[i]);
}
```

- **Demonstration:** WAP to store values in array named **scores** and access all the values of the array

Home Assignment

- WAP to sort the given value {2,7,4,9,1,0} in the ascending order
- WAP to search an element in the given array {‘a’, ‘b’, ‘c’, ‘d’, ‘e’, ‘f’, ‘g’, ‘h’, ‘i’}. If the search is successful, then the position of an element in the array has to be printed out.
- WAP to print the highest number of any 5 integer numbers
- WAP to compute the sum of {3,8,9,3,1}

FLIPPED CLASSROOM

ACTIVITY

- Work on the assigned question(10 mins)
- Present it to the class(5 mins)
- Q&A by other group members (5 mins)

Question 1:

Read one integer array with 10 elements. Read the array values using runtime initialization. Add the arrays values and display the sum of all array values.

Question 2:

Declare an array that can store 10 integer values. Read the array values using run-time initialization. Find the minimum value in an array, print the least value and its array index.

Question 3:

Declare an array that can store 12 characters. Initialize using runtime initialization. Read a character and check if its available in the array list, if so print the array index/subscript where the matched character is available.

Thank you