

Northern University, Nowshera

Programming Fundamentals (ECS-121)

Muhammad Athar

email id: athar@northern.edu.pk,

Whatsapp# 0333-5077664

(Week 15) Lecture 29 and 30

Objectives: Learning objectives of this lecture are

- **Pass Array as argument**
- **Constant parameter**
- **Char array as arguments**
- **Array of structure.**
- **Multi Dimension Arrays**



Northern University, Nowshera

Programming Fundamentals (ECS-121)

Muhammad Athar

email id: athar@northern.edu.pk,

Whatsapp# 0333-5077664

Lecture # 29 Passing array as argument

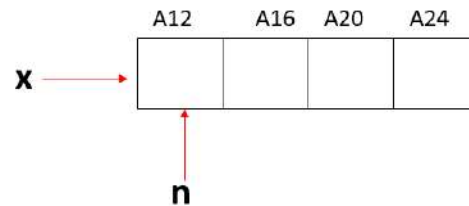
We have been discussing functions and so far we have discussed a lot about functions and have written program for different type of functions. In this lecture we will be discussing about how to pass arrays as argument to functions? First of all let us look at syntax of passing array to function.

```
void func( int x[ ] );
```

Can write size here

Syntax how a function should be written that is to receive array as argument. Now how to call this function? How to pass array? Whether array is passed as reference or pass as value? Figure given below provides answer to all these questions.

```
void func(int x[])
{
    // code here
}
void main()
{
    int n[4];
    func(n);
}
```



Array is always pass by reference.

Figure 1 Passing array to function.

In main array with name n is declared. While passing array to function just write name of array as mentioned in call of **func** in above figure. Note that at receiving end no new memory is created. As shown in above figure x is pointing towards same memory as that of n.

Northern University, Nowshera

Programming Fundamentals (ECS-121)

Muhammad Athar

email id: athar@northern.edu.pk,

Whatsapp# 0333-5077664

Program 1: Write a program that declares two integer array of size 5. Take input in array. After taking input display all the data of array.

```
#include "iostream"
using namespace std;
void main()
{
    int a1[5], a2[5];
    for (int i = 0; i < 5; i++)
    {
        cin >> a1[i];
    }
    for (int i = 0; i < 5; i++)
    {
        cin >> a2[i];
    }
    for (int i = 0; i < 5; i++)
    {
        cout << a1[i];
    }
    for (int i = 0; i < 5; i++)
    {
        cout << a2[i];
    }
}
```

Note that while taking input in a1 and a2 logic and number of time loop iteration is same. Why not write a function and call functions to input and output array.

Solution to above program with functions:

```
#include "iostream"
using namespace std;
void input(int x[])
{
    for (int i = 0; i < 5; i++)
        cin >> x[i];
}
void output(int x[])
{
    for (int i = 0; i < 5; i++)
        cout << x[i];
}
void main()
{
    int a1[5], a2[5];
    input(a1);
    input(a2);
    output(a1);
    output(a2);
}
```

Northern University, Nowshera

Programming Fundamentals (ECS-121)

Muhammad Athar

email id: athar@northern.edu.pk,

Whatsapp# 0333-5077664

In above case both the arrays are of same size what if a1 and a2 are of different? Than we have to modify the functions by passing size of array as argument. as given below.

```
#include "iostream"
using namespace std;
void input(int x[], int size)
{
    for (int i = 0; i < size; i++)
        cin >> x[i];
}
void output(int x[], int size)
{
    for (int i = 0; i < size; i++)
        cout << x[i];
}
void main()
{
    int a1[8], a2[10];
    input(a1, 8);
    input(a2, 10);
    output(a1, 8);
    output(a2, 10);
}
```

Now note one more thing inside input function what if we change value of size. As shown below.

```
void input(int x[], int size)
{
    size=size*10;
    for (int i = 0; i < size; i++)
        cin >> x[i];
}
void output(int x[], int size)
{
    size=size*2;
    for (int i = 0; i < size; i++)
        cout<< x[i];
}
```

It is clear that value of size must not be change we have to make this parameter as constant.

Northern University, Nowshera

Programming Fundamentals (ECS-121)

Muhammad Athar

email id: athar@northern.edu.pk,

Whatsapp# 0333-5077664

```
void input(int x[], const int size)
{
    size=size*10; // error cannot change const
    for (int i = 0; i < size; i++)
        cin >> x[i];
}
void output(int x[], const int size)
{
    size=size*2; // error cannot change const
    for (int i = 0; i < size; i++)
        cout<< x[i];
}
void main()
{
    int a1[8], a2[10];
    input(a1,8);
    input(a2,10);
    output(a1,8);
    output(a2,10);
}
```

Now value of size cannot be changed as it is made constant now, its value will be same that is provided at call time.

Pogram 2: You are required to input average temperature of 12 months in array of float. After taking input pass this temperature to a function maxtemp that will return maximum temperature.

Solution:

```
#include "iostream"
using namespace std;
float maxtemp(const float x[], const int size)
{
    float max = x[0];
    for (int i = 1; i < size; i++)
    {
```

Northern University, Nowshera

Programming Fundamentals (ECS-121)

Muhammad Athar

email id: athar@northern.edu.pk,

Whatsapp# 0333-5077664

```
        if (max < x[i])
            max = x[i];
    }
    return max;
}
void main()
{
    float temp[12];
    for (int i = 0; i < 12; i++)
        cin >> temp[i];
    cout << "Max temp " << maxtemp(temp, 12);
}
```

Program 3: In an organization there works several employees. You have to input total employees in an organization. You are required to store income tax paid by employee in array. After storing display total income tax paid by employees of organization. Use functions for input and computing sum.

Solution:

```
#include "iostream"
using namespace std;
void input(int taxamount[], const int total)
{
    for (int i = 0; i < total; i++)
        cin >> taxamount[i];
}
int totaltaxpaid(int taxamount[], const int total)
{
    int amount = 0;
    for (int i = 0; i < total; i++)
        amount += taxamount[i];
}
void main()
{
    int totalemp;
    cout << "Enter total employee ";
    cin >> totalemp;
    int* tax = new int[totalemp];
    input(tax, totalemp);
    totaltaxpaid(tax, totalemp);
}
```

Program 4: Input student roll number in an array of size 10. After taking input you are supposed to ask for a roll number to search from array. If roll number found display message *found* other wise display message *not found*.

```
#include "iostream"
using namespace std;
```

Northern University, Nowshera

Programming Fundamentals (ECS-121)

Muhammad Athar

email id: athar@northern.edu.pk,

Whatsapp# 0333-5077664

```
bool serachstudent(int rnum[], const int total, const int search)
{
    for (int i = 0; i < total; i++)
    {
        if (rnum[i] == search)
            return true;
    }
    return false;
}

void main()
{
    int rollnum[10];
    for (int i = 0; i < 10; i++)
        cin >> rollnum[i];
    int search;
    cout << "Enter roll number to search ";
    cin >> search;
    if (serachstudent(rollnum, 10, search))
        cout << "Found ";
    else
        cout << "Not found ";
}
```

Program 5: You are supposed to input employee numbers in a array of size 10. Employee number must not be repeated in array. Use function to ensure that employee number is not repeated.

Solution:

```
#include "iostream"
using namespace std;

bool empnumexist(int emnum[], const int total, const int search)
{
    for (int i = 0; i < total; i++)
    {
        if (emnum[i] == search)
            return true;
    }
    return false;
}

void main()
{
    int emp_num[10];
    for (int i = 0; i < 10; i++)
    {
        cout << "Enter employee number ";
        cin >> emp_num[i];
        while (empnumexist(emp_num, i, emp_num[i]))
        {
            cout << "It already exists re-enter ";
            cin >> emp_num[i];
        }
    }
}
```

Northern University, Nowshera

Programming Fundamentals (ECS-121)

Muhammad Athar

email id: athar@northern.edu.pk,

Whatsapp# 0333-5077664

```
}  
}
```

Program 6: In a competition exam there appears 10 students. You have to input marks of obtained in array of size 10. Sort the given array in descending order. Use function for sorting.

Solution:

```
#include "iostream"  
using namespace std;  
  
void swap(int& x, int& y)  
{  
    int temp = x;  
    x = y;  
    y = temp;  
}  
  
void sort(int arr[], int total)  
{  
    for (int i = 0; i < total; i++)  
    {  
        int maxind = i;  
        for (int j = i + 1; j < total; j++)  
        {  
            if (arr[maxind] < arr[j])  
                maxind = j;  
        }  
        swap(arr[i], arr[maxind]);  
    }  
}  
  
void main()  
{  
    int mark[10];  
    for (int i = 0; i < 10; i++)  
        cin >> mark[i];  
    sort(mark, 10);  
}
```

Northern University, Nowshera

Programming Fundamentals (ECS-121)

Muhammad Athar

email id: athar@northern.edu.pk,

Whatsapp# 0333-5077664

Lecture # 30 Char array and Structures as argument

In this lecture we will pass see some programs in which char array and structures variable is passed as arguments to functions. Before moving on to user defined function let us have look at a built in function that receives char array as argument.

```
void main()
{
    char ch[20];
    gets_s(ch);
}
```

gets_s is a built in function that receives char array as argument and inputs with spaces. Why not make our own function that does same as gets_s. As discussed earlier that cin does not input with spaces. So in our function we will be using getchar(); function that returns char after taking input. Our function looks like.

```
void input_char_array(char ch[])
{
    int i = 0;
    char c;
    while (true)
    {
        ch[i] = getchar();

        if (ch[i] == '\n') //13 ascii value for enter
        {
            ch[i] = '\0';
            return;
        }
        i++;
    }
}

void main()
{
    char ch[20];
    input_char_array(ch);
    cout << ch << endl;
}
```

Northern University, Nowshera

Programming Fundamentals (ECS-121)

Muhammad Athar

email id: athar@northern.edu.pk,

Whatsapp# 0333-5077664

Program 1: Make a function name has_all_alphabets that receives char array as argument. If all letters are alphabets than return true other wise return false.

Solution:

```
bool has_all_alphabets(char ch[])
{
    int i = 0;
    while (ch[i] != '\0')
    {
        if (ch[i] >= 'A' && ch[i] <= 'Z')
            i++;
        else if (ch[i] >= 'a' && ch[i] <= 'z')
            i++;
        else
            return false;
    }
    return true;
}
```

Program 2: Write a program in which you are supposed to input name of a person. Ensure that name contains all alphabets.

Solution:

```
#include "iostream"
using namespace std;

bool has_all_alphabets(char ch[])
{
    int i = 0;
    while (ch[i] != '\0')
    {
        if (ch[i] >= 'A' && ch[i] <= 'Z')
            i++;
        else if (ch[i] >= 'a' && ch[i] <= 'z')
            i++;
        else
            return false;
    }
    return true;
}

void main()
{
    char name[10];
    cin >> name;
```

Northern University, Nowshera

Programming Fundamentals (ECS-121)

Muhammad Athar

email id: athar@northern.edu.pk,

Whatsapp# 0333-5077664

```
while (!has_all_alphabets(name))
{
    cout << "re-enter invalid name ";
    cin >> name;
}

}
```

Char Array Comparison Using functions.

Logic:

- Find length of both arrays.
- If length is not equal than arrays are not equal.
- If length is equal than compare values of each index if same values found at every index than equal
- In case of any different value arrays are not equal.

Program:

```
#include "iostream"
using namespace std;

int length(char ch[])
{
    int i = 0;
    while (ch[i] != '\0')
        i++;

    return i;
}

bool compare(char ch1[], char ch2[])
{
    int len1 = length(ch1);
    int len2 = length(ch2);
    if (len1 != len2)
        return false;
    for (int i = 0; i < len1; i++)
    {
        if (ch1[i] != ch2[i])
            return false;
    }
    return true;
}

void main()
{
```

Northern University, Nowshera

Programming Fundamentals (ECS-121)

Muhammad Athar

email id: athar@northern.edu.pk,

Whatsapp# 0333-5077664

```
char arr1[10], arr2[20];
cin >> arr1;
cin >> arr2;
if (compare(arr1, arr2))
    cout << "equal \n";
else
    cout << "Not equal\n";
}
```

Program 3: Make two functions with name convert_to_upper and convert_to_lower. convert_to_upper converts all alphabets to upper case and convert_to_lower convert all alphabets to lower case. Both functions receives char array as argument.

Solution:

```
void convertupper(char ch[])
{
    int i = 0;
    while (ch[i] != '\0')
    {
        if (ch[i] >= 'a' && ch[i] <= 'z')
            ch[i] -= 32;
        i++;
    }
}

void converttolower(char ch[])
{
    int i = 0;
    while (ch[i] != '\0')
    {
        if (ch[i] >= 'A' && ch[i] <= 'Z')
            ch[i] += 32;
        i++;
    }
}
```

Program 4: Make three functions with name trimstart, trimend and trim. All functions receives char array as argument. trimstart removes spaces from start. trimend removes spaces from end and trim removes spaces from start and end.

Solution:

```
void trimstart(char ch[])
{
    int i = 0;
    while (ch[i] == ' ')
    {

```

Northern University, Nowshera

Programming Fundamentals (ECS-121)

Muhammad Athar

email id: athar@northern.edu.pk,

Whatsapp# 0333-5077664

```
        i = 0;
        while (ch[i] != '\0')
        {
            ch[i] = ch[i + 1];
            i++;
        }
    }
}

void trimend(char ch[])
{
    int i = 0;
    while (ch[i] != '\0')
    {
        i++;
    }
    i = i - 1;
    while (ch[i] == ' ')
        ch[i--] = '\0';
}

void trim(char ch[])
{
    trimstart(ch);
    trimend(ch);
}
```

Program 5: Make a structure NIC that contains following data members.

- ❖ Name (must contains alphabets).
- ❖ Father name (must contains alphabets).
- ❖ Contact number (Must have all digits and have length 11)
- ❖ NIC number must have two formats
 - ❖ 1230155222569 (all digits and have length 13)
 - ❖ 12301-5522256-9 (length 15 have digits except 5 and 13 index)

Note: Trim all arrays to remove extra spaces from start and end.

Solution:

```
#include "iostream"
using namespace std;
struct NIC
{
    char name[12], fname[12];
    char cnic[16], contact[12];
}
```

Northern University, Nowshera

Programming Fundamentals (ECS-121)

Muhammad Athar

email id: athar@northern.edu.pk,

Whatsapp# 0333-5077664

```
};
void trimstart(char ch[])
{
    int i = 0;
    while (ch[0] == ' ')
    {
        i = 0;
        while (ch[i] == '\0')
        {
            if (ch[i] >= 'a' && ch[i] <= 'z')
                ch[i] -= 32;
            i++;
        }
    }
}

void trimend(char ch[])
{
    int i = 0;
    while (ch[i] != '\0')
    {
        i++;
    }
    i = i - 1;
    while (ch[i] == ' ')
        ch[i--] = '\0';
}

void trim(char ch[])
{
    trimstart(ch);
    trimend(ch);
}

bool has_all_alphabets(char ch[])
{
    int i = 0;
    while (ch[i] != '\0')
    {
        if (ch[i] >= 'A' && ch[i] <= 'Z')
            i++;
        else if (ch[i] >= 'a' && ch[i] <= 'z')
            i++;
        else
            return false;
    }
    return true;
}

int countdigit(char ch[])
{
    int c = 0;
    int i = 0;
    while (ch[i] != '\0')
    {
        if (ch[i] >= '0' && ch[i] <= '9')
            c++;
    }
}
```

Programming Fundamentals (ECS-121)
Muhammad Athar email id: athar@northern.edu.pk, Whatsapp# 0333-5077664

Whatsapp# 0333-5077664

15))

Northern University, Nowshera

Programming Fundamentals (ECS-121)

Muhammad Athar

email id: athar@northern.edu.pk,

Whatsapp# 0333-5077664

```
} }
```

Multidimensional Arrays

```
#include <iostream>                                     // Array example 3, Multidimensional Arrays
using namespace std;
#include <iomanip>

const int DISTRICTS = 4;                                //array dimensions
const int MONTHS = 3;

int main ( )
{
    int d, m;
    double sales [DISTRICTS] [MONTHS];

    for (d = 0 ; d < DISTRICTS ; d++)
        for (m = 0; m < MONTHS; m++)
        {
            cout << "Enter sales for district " << d+1 ;
            cout << ", month " << m+1 << ": ";
            cin >> sales [d] [m];                        //store number in array sales
        }

    for (d=0 ; d < DISTRICTS ; d++)
    {
        cout << "\nDistrict " << d+1;
        for (m = 0 ; m < MONTHS; m++)
        {
            cout << setiosflags(ios::fixed)                //not exponential
                << setiosflags(ios::showpoint)             //always use point
                << setprecision(2)                          //digits to right
                << setw( 10 )                               //field width
                << sales [d] [m] ;                          //get number from array
        }
        cout << endl;
        return 0;
    }
}

//-----
-

#include <iostream>                                     // Array example 4, 3 Dimensional Array
using namespace std;

int main ( )
{
```

Northern University, Nowshera

Programming Fundamentals (ECS-121)

Muhammad Athar

email id: athar@northern.edu.pk,

Whatsapp# 0333-5077664

```
int x, y, z;
int ary1[5][4][3]; //Array of 5 x 4 x 3 integer elements

for (x = 0 ; x < 5 ; x++)
    for (y = 0 ; y < 4 ; y++)
        for (z = 0 ; z < 3 ; z++)
            ary1[x][y][z] = 0;
return 0;
}
//-----

#include <iostream> // Array example 5, 2 Dimensional Array
using namespace std;

int main ( )
{
    int x, y; //Array of 3 x 2 integer elements
    int ary [3][2] = { { 1, 2}, { 3, 4}, { 5, 6} } ;

    for (x = 0 ; x < 3 ; x++)
        for (y = 0 ; y < 2 ; y++)
            cout << x << "," << y << " : " << ary[x][y] << " ; ";

    return 0;
}
//-----

#include <iostream> // Array example 6, 3 Dimensional Array
using namespace std;

int main ( )
{
    int x, y, z , item; //Array of 4 x 3 x 2 integer elements
    int loc_x = 100, loc_y = 100, loc_z = 100;
    int ary[4][3][2] = {
        { { 1, 2}, { 3, 4}, { 5, 6} },
        { { 7, 8}, { 9, 10}, {11, 12} },
        { {13, 14}, {15, 16}, {17, 18} },
        { {19, 20}, {21, 22}, {23, 24} } };

    for (x = 0 ; x < 4 ; x++)
    { cout << endl;
        for (y = 0 ; y < 3 ; y++)
            for (z = 0 ; z < 2 ; z++)
                cout << x << "," << y << "," << z << " : " << ary[x][y][z] << " ; ";
    }
}
```

Northern University, Nowshera

Programming Fundamentals (ECS-121)

Muhammad Athar

email id: athar@northern.edu.pk,

Whatsapp# 0333-5077664

```
cout << "\n\n Enter an item to search:" ; cin >> item;

for (x = 0 ; x < 4 ; x++)
    for (y = 0 ; y < 3 ; y++)
        for (z = 0 ; z < 2 ; z++)
            if (item == ary[x] [y] [z])
                {
                    loc_x = x;
                    loc_y = y;
                    loc_z = z;
                }
cout << "Location of item is: " << loc_x << "," << loc_y << "," << loc_z << endl;
return 0;
}
```

Arrays as Arguments to Functions

```
#include <iostream>
using namespace std;

const int X_size = 4;
const int Y_size = 3;
const int Z_size = 2;

void display ( int a1 [ X_size] [Y_size] [Z_size] );
// void display ( int [ ] [Y_size] [Z_size] );
void search ( int item, int a2 [ X_size] [Y_size] [Z_size] );
// void search ( int, int [ ] [Y_size] [Z_size] );

int main ( )
{
    int item;
    int ary[X_size] [Y_size] [Z_size] = { { { 1, 2}, { 3, 4}, { 5, 6} },
                                           { { 7, 8}, { 9, 10}, {11, 12} },
                                           { {13, 14}, {15, 16}, {17, 18} },
                                           { {19, 20}, {21, 22}, {23, 24} } };

    display (ary);
    cout << "\n\n Enter an item to search:" ; cin >> item;
```

// Array example 7, Function Arguments

//Array Dimensions

//declarations

Northern University, Nowshera

Programming Fundamentals (ECS-121)

Muhammad Athar

email id: athar@northern.edu.pk,

Whatsapp# 0333-5077664

```
    search ( item, ary);
    return 0;
}

void display ( int a1 [ X_size] [Y_size] [Z_size] )
{
    for (int x = 0 ; x < X_size ; x++)
        for (int y = 0 ; y < Y_size ; y++)
            for (int z = 0 ; z < Z_size; z++)
                cout << endl << x << “,”<< y << “,” << z << “ : ” << a1[x] [y] [z] ;
}

void search ( int item, int a2 [ X_size] [Y_size] [Z_size] )
{
    int loc_x = 100, loc_y = 100, loc_z = 100;
    for (int x = 0 ; x < X_size ; x++)
        for (int y = 0 ; y < Y_size ; y++)
            for (int z = 0 ; z < Z_size; z++)
                if (item == a2[x] [y] [z])
                {
                    loc_x = x;
                    loc_y = y;
                    loc_z = z;
                }
    cout << “Location of item is: ” << loc_x << “,” << loc_y << “,” << loc_z << endl;
}
//-----
```

Arrays of Structures

```
#include <iostream>
```

```
//Array Example 8: Arrays and Structures
```

```
using namespace std;
```

```
struct part
```

```
{
```

```
    int model_number;           //ID number of part
```

```
    int part_number;           //ID number of subparts
```

Northern University, Nowshera

Programming Fundamentals (ECS-121)

Muhammad Athar

email id: athar@northern.edu.pk,

Whatsapp# 0333-5077664

```
float cost;

};

int main ( )
{
    const int X_size = 5;           //constant X_size for x dimension of array
    const int Y_size = 3;           //constant Y_size for y dimension

    int i, j;

    part dummy_part = {0, 0, 0.0F}; //defining dummy part
    part p[X_size] [Y_size];         //defining a 2 dim array of part type

    for (i = 0; i < X_size ; i++)     //Initializing array with dummy part
        for ( j = 0 ; j < Y_size ; j++)
            p[i] [j] = dummy_part;

    for (i = 0; i < X_size ; i++)
        for (j = 0; j < Y_size ; j++)
            cout << endl << i << "," << j << " : " << p[i] [j].model_number << ","
                << p[i] [j].part_number << "," << p[i] [j].cost;

    for (i = 0; i < X_size ; i++)
        for (j = 0; j < Y_size ; j++)
        {
            p[i] [j].model_number = i + j;
            p[i] [j].part_number = i + j + 100 ;
            p[i] [j].cost = 50.00F;
        }
}
```

Northern University, Nowshera

Programming Fundamentals (ECS-121)

Muhammad Athar

email id: athar@northern.edu.pk,

Whatsapp# 0333-5077664

```
}
```

```
for (i = 0; i < X_size ; i++)
```

```
for (j = 0; j < Y_size ; j++)
```

```
cout << endl << i << "," << j << " : " << p[i] [j].model_number << ","
```

```
<< p[i] [j].part_number << "," << p[i] [j].cost;
```

```
return 0;
```

```
}
```

