Programming Fundamental

Lab Manual - Week 09

Introduction

Welcome Back to your favorite Programming Lab students. In this lab manual, we shall work together to learn and implement new programming concepts.

Skills to be learned:

- Declare and initialize arrays of different data types.

Let's do some coding.

Skill: Declare and initialize arrays of different data types.

Introduction

By this week, you have learned how to write a program that contains functions, loops, and conditional structures. In this class, we will learn about another very powerful concept known as Arrays.

An array is a collection of **similar data items** stored at **contiguous memory locations** and elements that can be accessed using the **indices of an array.**
Following are the types of arrays that are often used in programs.

- Integer arrays
- Float arrays
- Char arrays
- String arrays

Consider a task in which you want to store the mid-term marks of Programming Fundamentals of 200 students. Let's create an Array to store the marks of 200 students of the Computer Science Department of UET Lahore.

Array Declaration

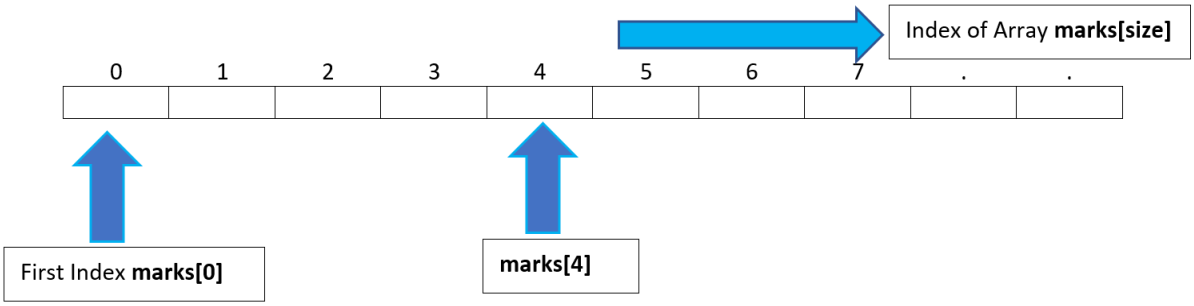
Following is the Syntax of an Array **dataType ArrayName[size];**	For example, int marks[200];

Skill: Declare and initialize arrays of different data types

This C++ statement will declare an array of **size** in the memory of the computer. Now, we can use the indices of the array to store the marks of the students at different addresses.

Index of Array

An array is a collection of items on contiguous memory locations. Every item in the array exists on a unique address in memory. Consider the following diagram for a better understanding.



We can use the assignment operator to assign a value using the index of the array item.

Consider the following program for better understanding

Task 01(WP): Write a program that declares an array of 5 integer elements, initializes them one by one and displays them.

Skill: Declare and initialize arrays of different data types

```cpp
main()
{
    int numbers[5];

    numbers[0] = 15;
    numbers[1] = 12;
    numbers[2] = 13;
    numbers[3] = 19;
    numbers[4] = 10;

    cout << "The 1st element at location numbers[0] is: " << numbers[0] << endl;
    cout << "The 2nd element at location numbers[0] is: " << numbers[1] << endl;
    cout << "The 3rd element at location numbers[0] is: " << numbers[2] << endl;
    cout << "The 4th element at location numbers[0] is: " << numbers[3] << endl;
    cout << "The 5th element at location numbers[0] is: " << numbers[4] << endl;

}
```

We use the assignment operator to assign values to array item using indices of an array.

```
The 1st element at location numbers[0] is: 15
The 2nd element at location numbers[0] is: 12
The 3rd element at location numbers[0] is: 13
The 4th element at location numbers[0] is: 19
The 5th element at location numbers[0] is: 10
```

Task 02(CA): Write a program that declares an array of 5 elements, initializes them one by one and displays only the 2nd and 4th elements of an array.

Similarly, we can store the value entered by the user in the array by using the following statement.
cin >> arrayName[index];

Task 03(WP): Write a program that declares an array of 5 elements, initializes them one by one by user input, and displays 1st and last elements of the array.

Skill: Declare and initialize arrays of different data types

```cpp
main()
{
    int numbers[5];
    cout << "Enter 1st Number: ";
    cin >> numbers[0];
    cout << "Enter 2nd Number: ";
    cin >> numbers[1];
    cout << "Enter 3rd Number: ";
    cin >> numbers[2];
    cout << "Enter 4th Number: ";
    cin >> numbers[3];
    cout << "Enter 5th Number: ";
    cin >> numbers[4];

    cout << "The 1st element in array is: " << numbers[0] << endl;
    cout << "The last element in array is: " << numbers[4] << endl;

}
```

```
Enter 1st Number: 12
Enter 2nd Number: 14
Enter 3rd Number: 15
Enter 4th Number: 13
Enter 5th Number: 17
The 1st element in array is: 12
The last element in array is: 17
```

🚧 Problem 🚧

We have used the same variable name to store multiple values but our code is repeating for taking input.

Is there any better way ❓

💡 Solution 💡

We can use the counting loop for taking input in the array from the user

Consider the same question with loops below.

Task 04(WP): Write a program that declares an array of 5 elements, initialize them one by one by user input and display 1st and last elements of the array.
Instruction: Use a loop for taking input from the user

Skill: Declare and initialize arrays of different data types

```cpp
main()
{
    int numbers[5];

    for (int count = 0; count < 5; count = count + 1)
    {
        cout << "Enter Number: ";
        cin >> numbers[count];
    }
    cout << "The 1st element in array is: " << numbers[0] << endl;
    cout << "The last element in array is: " << numbers[4] << endl;
}
```

```
Enter 1st Number: 12
Enter 2nd Number: 14
Enter 3rd Number: 15
Enter 4th Number: 13
Enter 5th Number: 17
The 1st element in array is: 12
The last element in array is: 17
```

Task 05(OP): Write a program that takes **n numbers** from the user, stores them in an array, and displays those numbers on the screen.

Instruction: First of all the user will tell how many numbers he/she wants to enter. Then take those numbers as input from the user and store them in the array.

Similarly, we can use the array for solving more complex problems.
Consider the following task for better understanding.

Task 06(CL): Write a program that prints the sum and average of the first 5 natural numbers on the screen.

Skill: Declare and initialize arrays of different data types

```
main()
{
    int sum = 0;
    float average = 0;
    int numbers[5] = {1, 2, 3, 4, 5};
    for (int idx = 0; idx < 5; idx = idx + 1)
    {
        sum = sum + numbers[idx];
    }
    average = sum/5;
    cout << "Sum: " << sum << endl;
    cout << "Avg: " << average << endl;
}
```

```
Sum: 15
Avg: 3
```

String as Character Array

String values are stored as a **character array** inside memory and the end of the string is stored as '**\0**' referred to as a null character.

Consider the following task for better understanding.

Task 07(WP): Write a program that prints whether a specific character is present in a string or not.

```
#include <iostream>
using namespace std;
bool check(string word, char letter)
{
    bool isFound = false;
    for (int idx = 0; word[idx] != '\0'; idx++)
    {
        if (word[idx] == letter)
        {
            isFound = true;
            break;
        }
    }
    return isFound;
}
```

```
main()
{
    string word;
    char letter;
    cout << "Enter a Word: ";
    cin >> word;
    cout << "Enter the character you want to find: ";
    cin >> letter;
    if (check(word, letter))
    {
        cout << letter << " is found in " << word;
    }
    else
    {
        cout << letter << " is not found in " << word;
    }
}
```

```
Enter a Word: programming
Enter the character you want to find: m
m is found in programming
```

Point to ponder:

We have used a counter loop in the check function.

Skill: Declare and initialize arrays of different data types

Programming Fundamental

Lab Manual - Week 09

Did you already knew how many times we had to iterate the loop..??
Which loop would be better in this case?

Consider the following amazing facts about the string variables.

Do you know? 💡
You can use the **getline(cin, stringVariableName)** function to take string input that includes a space character.

Do you know? 💡
You can declare a string by using any of the following methods. Pretty cool, isn't ?

```
main()
{
    string word = "C++";
    char word[4] = "C++";
    char word[] = "C++";
    char word[]= {'C','+','+','\0'};
    char word[4]= {'C','+','+','\0'};
    char word[]= {'C','+','+'};
    char word[100] = "C++";
}
```

Task 01(OP): (Word Postmortem)

Write a program that takes a word from the user and stores it in a character array and passes it to a function that displays the location of all alphabets in the array.

```
G:\Semesters\Programming Fundamentals (Fall 2023)\Week 9\Lab Tasks>Task1.exe
Enter a word: hello
h found at position 0
e found at position 1
l found at position 2
l found at position 3
o found at position 4
```

Task 02(OP): (Reverse Word)

Skill: Declare and initialize arrays of different data types

Programming Fundamental

Lab Manual - Week 09

Write a function that takes a string as input parameter and displays it in reverse order.

```
G:\Semesters\Programming Fundamentals (Fall 2023)\Week 9\Lab Tasks>Task2.exe
Enter a string: hello
Reversed String: olleh
```

Task 03(OP): (Next Letter)

Write a C++ function to change every letter in a given string with the letter following it in the alphabet (ie. a becomes b, p becomes q, z becomes a).

For Example:

Input: aslam

Output: btmbn

```
G:\Semesters\Programming Fundamentals (Fall 2023)\Week 9\Lab Tasks>Task3.exe
Enter a String: hello
Shifted String: ifmmp

G:\Semesters\Programming Fundamentals (Fall 2023)\Week 9\Lab Tasks>Task3.exe
Enter a String: Zakuta Jin
Shifted String: Ablvub Kjo
```

Hint for solving this problem in an easy way:

ASCII (American Standard Code for Information Interchange) codes in C++ refer to the numeric representations of characters in the ASCII character set. The ASCII character set includes various characters, such as letters, digits, punctuation marks, and control characters, each of which is assigned a unique numerical value.

In C++, you can obtain the ASCII code of a character by storing it in an int variable. Here's a simple example:

```
char character = 'A';  // Change this to the character you want to find ASCII code for
int asciiCode = character;

cout << "The ASCII code for " << character << " is " << asciiCode;
```

Skill: Declare and initialize arrays of different data types

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[END OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	|
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

Ignore the Hex column, just consider the **Decimal and Char** column.

Task 04(OP): (Reverse Numbers)

Write a program that takes **n** numbers from the user, stores them in an array, and passes them to a function that prints them in reverse order.

```
G:\Semesters\Programming Fundamentals (Fall 2023)\Week 9\Lab Tasks>Task4.exe
Enter the number of elements: 3
Enter 3 numbers, one per line:
1
2
3
Numbers in reverse order: 3 2 1

G:\Semesters\Programming Fundamentals (Fall 2023)\Week 9\Lab Tasks>Task4.exe
Enter the number of elements: -5
Invalid input. Number of elements must be greater than 0.
```

Solution:

Skill: Declare and initialize arrays of different data types

Programming Fundamental

Lab Manual - Week 09

```cpp
#include <iostream>
using namespace std;
// Function to print an array in reverse order
void printReverseArray(int arr[], int n)
{
    for (int i = n - 1; i >= 0; i--)
    {
        cout << arr[i] << " ";
    }
    cout << endl;
}

int main()
{
    int n;
    cout << "Enter the number of elements: ";
    cin >> n;

    if (n <= 0) {
        cout << "Invalid input. Number of elements must be greater than 0." << endl;
        return 0; // Terminate the Program
    }

    int arr[n]; // Declare an array to store the numbers

    cout << "Enter " << n << " numbers, one per line:" << endl;
    for (int i = 0; i < n; i++)
    {
        cin >> arr[i];
    }

    cout << "Numbers in reverse order: ";
    printReverseArray(arr, n);

    return 0; // Terminate the Program
}
```

When you have to pass an array to a function you have to just give the name of the array.

```cpp
printReverseArray(arr, n);
```

But when to define a function that is receiving the array. You must give 2 things.

1. Complete Array (just add the bracket signs but do not give the size of the array in the brackets)
2. Array Size

```cpp
void printReverseArray(int arr[], int n)
```

Task 05(OP): (Validate Input)

Write a program that takes **n** numbers from the user and stores them in an array.
The program should print “Already Entered” if the user has already entered that number.

Skill: Declare and initialize arrays of different data types

```
G:\Semesters\Programming Fundamentals (Fall 2023)\Week 9\Lab Tasks>Task5.exe
Enter the number of elements: 3
Enter 3 numbers, one per line:
1
1
Already Entered: 1
2
Unique numbers entered: 1 2
```

Instructions:

You must make a function to check if the number is already present in the array or not.

```
bool isAlreadyEntered(int arr[], int size, int number)
```

Task 06(OP): (Find Largest)

Write a program that takes **n** numbers from the user and stores them in an array and call a function that returns the largest number entered by the user.

Function Prototype should be:

```
int findLargestNumber(int arr[], int size)
```

```
G:\Semesters\Programming Fundamentals (Fall 2023)\Week 9\Lab Tasks>Task6.exe
Enter the number of elements: 5
Enter 5 numbers, one per line:
4
8
33
-34
6
The largest number entered is: 33
```

Task 07(CP): (Resistors)

When resistors are connected together in series, the same current passes through each resistor in the chain and the total resistance, RT, of the circuit must be equal to the sum of all the individual resistors added together. That is

RT = R1 + R2 + R3 ...

Create a program that takes an array of values resistance that are connected in series, and calculates the total resistance of the circuit in ohms. The ohm is the standard unit of electrical resistance in the International System of Units (SI).

Create the function with the following prototype

Skill: Declare and initialize arrays of different data types

```
double calculateTotalResistance(double resistance[], int size)
```

Test Cases:

[1, 5, 6, 3] ➞ "15"

[16, 3.5, 6] ➞ "25.5"

[0.5, 0.5] ➞ "1.0"

```
G:\Semesters\Programming Fundamentals (Fall 2023)\Week 9\Lab Tasks>Task7.exe
Enter the number of resistors in the series circuit: 3
Enter the resistance values (in ohms) of the 3 resistors, one per line:
16
3.5
6
The total resistance of the series circuit is 25.5 ohms.
```

Task 08(CP): (Insert Array in Middle)

Create a program that takes two arrays and inserts the second array in the middle of the first array. The first array always has two elements.

Function prototype should be

```
void insertArrayInMiddle(int firstArray[], int firstArraySize, int secondArray[], int secondArraySize)
```

Test Cases:

[1, 10], [2, 3, 4, 5, 6, 7, 8, 9] ➞ [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

[15,150], [45, 75, 35] ➞ [15, 45, 75, 35, 150]

```
G:\Semesters\Programming Fundamentals (Fall 2023)\Week 9\Lab Tasks>Task8.exe
Enter the number of elements for the first array (must be 2): 2
Enter 2 elements for the first array, one per line:
1
6
Enter the number of elements for the second array: 4
Enter 4 elements for the second array, one per line:
2
3
4
5
Resulting array: [1, 2, 3, 4, 5, 6]
```

Task 09(CP): (Pay the Bill)

Skill: Declare and initialize arrays of different data types

Given a total due and an array representing the amount of change in your pocket, determine whether or not you are able to pay for the item. The change will always be represented in the following order: quarters, dimes, nickels, and pennies.
[25, 20, 5, 0], 4.25 should yield true, since having 25 quarters, 20 dimes, 5 nickels, and 0 pennies gives you 6.25 + 2 + .25 + 0 = 8.50.

NOTE:

- quarter: 25 cents / $0.25
- dime: 10 cents / $0.10
- nickel: 5 cents / $0.05
- penny: 1 cent / $0.01

Test Cases:

[2, 100, 0, 0], 14.11 ➞ false

[0, 0, 20, 5], 0.75 ➞ true

[30, 40, 20, 5], 12.55 ➞ true

[10, 0, 0, 50], 3.85 ➞ false

[1, 0, 5, 219], 19.99 ➞ false

Function Prototype should be:

```
bool canPayWithChange(double change[], double totalDue)
```

```
G:\Semesters\Programming Fundamentals (Fall 2023)\Week 9\Lab Tasks>Task9.exe
Enter quarters: 30
Enter dimes: 40
Enter nickels: 20
Enter pennies: 5
Enter the total amount due: $12.55
Can you pay the amount? Yes

G:\Semesters\Programming Fundamentals (Fall 2023)\Week 9\Lab Tasks>Task9.exe
Enter quarters: 10
Enter dimes: 0
Enter nickels: 0
Enter pennies: 50
Enter the total amount due: $3.85
Can you pay the amount? No
```

Task 10(CP): (Something)

Write a program that prints the string "something" joined with a space " " and the given argument a.

Skill: Declare and initialize arrays of different data types

Test Cases:

"is better than nothing" ➞ "something is better than nothing"

"Bob Jane" ➞ "something Bob Jane"

"something" ➞ "something something"

```
G:\Semesters\Programming Fundamentals (Fall 2023)\Week 9\Lab Tasks>Task10.exe
Enter the argument 'a': Hello World
Result: something Hello World
```

Task 11(CP): (Vowels Removed)

Create a program that takes a string and returns a new string with all vowels removed.

Test Cases:

Input:

"I have never seen a thin person drinking Diet Coke."

Output:

" hv nvr sn thn prsn drnkng Dt Ck."

```
G:\Semesters\Programming Fundamentals (Fall 2023)\Week 9\Lab Tasks>Task11.exe
Enter a string: HEllo World
String with vowels removed: Hll Wrld
```

Task 12(CP): (Special Array or Not)

An array is **special** if every **even index** contains an **even number** and every **odd index** contains an **odd number**. Create a function that returns `true` if an array is **special**, and `false` otherwise.

Function prototype should be

```
bool isSpecialArray(int arr[], int size)
```

Examples

isSpecialArray([2, 7, 4, 9, 6, 1, 6, 3]) ➞ true

// Even indices: [2, 4, 6, 6]; Odd indices: [7, 9, 1, 3]

isSpecialArray([2, 7, 9, 1, 6, 1, 6, 3]) ➞ false

// Index 2 has an odd number 9.

isSpecialArray([2, 7, 8, 8, 6, 1, 6, 3]) ➞ false

Skill: Declare and initialize arrays of different data types

// Index 3 has an even number 8.

```
G:\Semesters\Programming Fundamentals (Fall 2023)\Week 9\Lab Tasks>Task12.exe
Enter the size of the array: 5
Enter 5 elements of the array:
1
2
3
4
5
The array is not special

G:\Semesters\Programming Fundamentals (Fall 2023)\Week 9\Lab Tasks>Task12.exe
Enter the size of the array: 5
Enter 5 elements of the array:
0
1
2
3
4
The array is special
```

Task 13(CP): (Jazzify)

Create a function which concatenates the number **7** to the end of every chord in an array. Ignore all chords which already end with 7.

Function Prototype should be:

```
void jazzifyChords(string chords[], int numChords)
```

Examples

jazzify(["G", "F", "C"]) ➞ ["G7", "F7", "C7"]

jazzify(["Dm", "G", "E", "A"]) ➞ ["Dm7", "G7", "E7", "A7"]

jazzify(["F7", "E7", "A7", "Ab7", "Gm7", "C7"]) ➞ ["F7", "E7", "A7", "Ab7", "Gm7", "C7"]

Skill: Declare and initialize arrays of different data types

```
G:\Semesters\Programming Fundamentals (Fall 2023)\Week 9\Lab Tasks>Task13.exe
Enter the number of chords: 5
Enter 5 chords, one per line:
Dm
G
E
F7
e7
Jazzified chords: [Dm7, G7, E7, F7, e7]
```

Good Luck and Best Wishes !!

Happy Coding ahead :)

Skill: Declare and initialize arrays of different data types