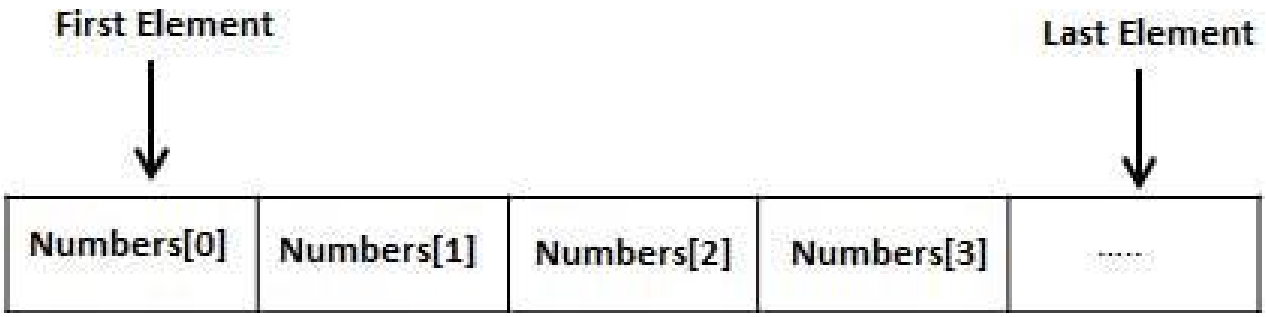
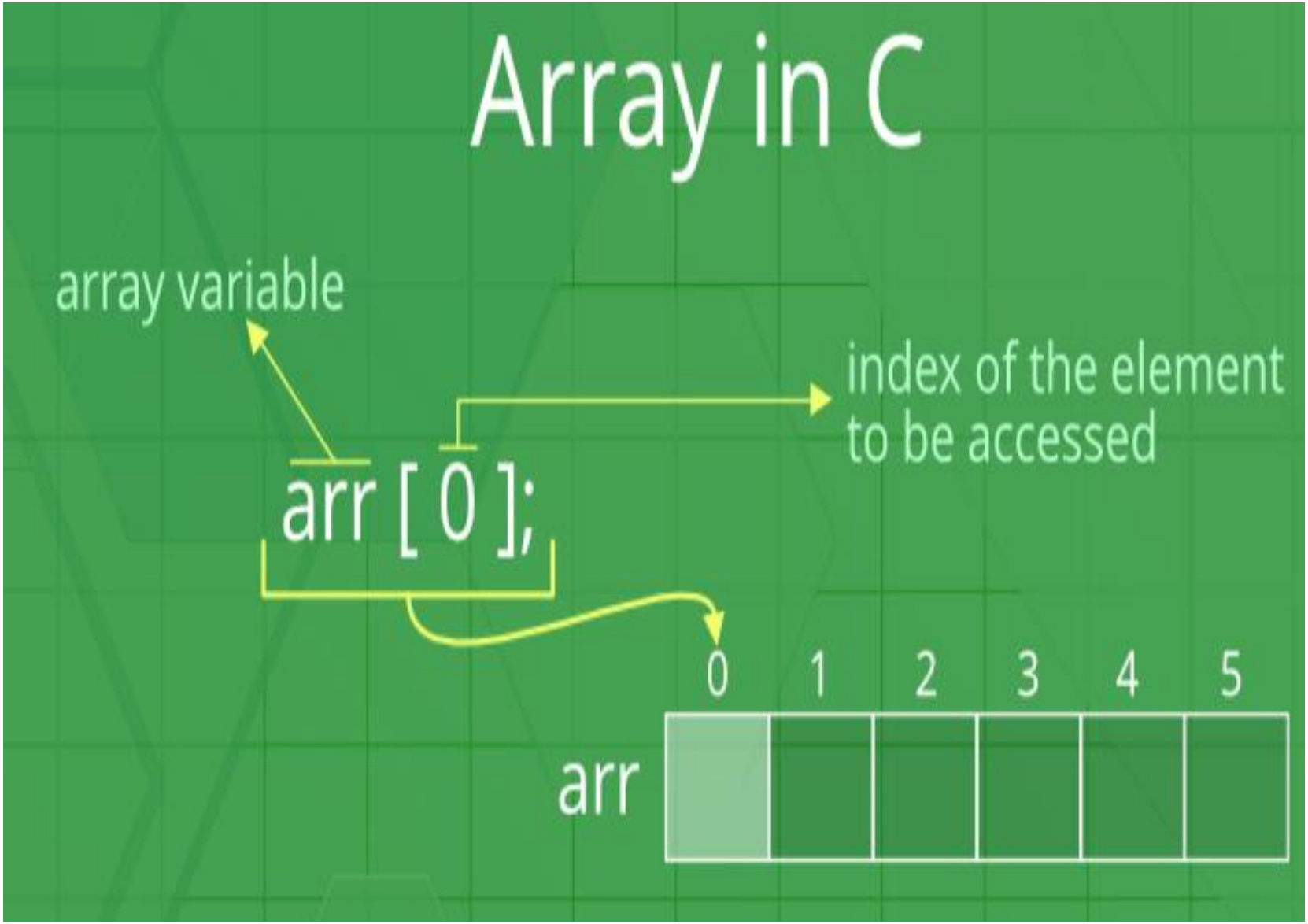
C Programming
Array

Dr. Asif Uddin Khan

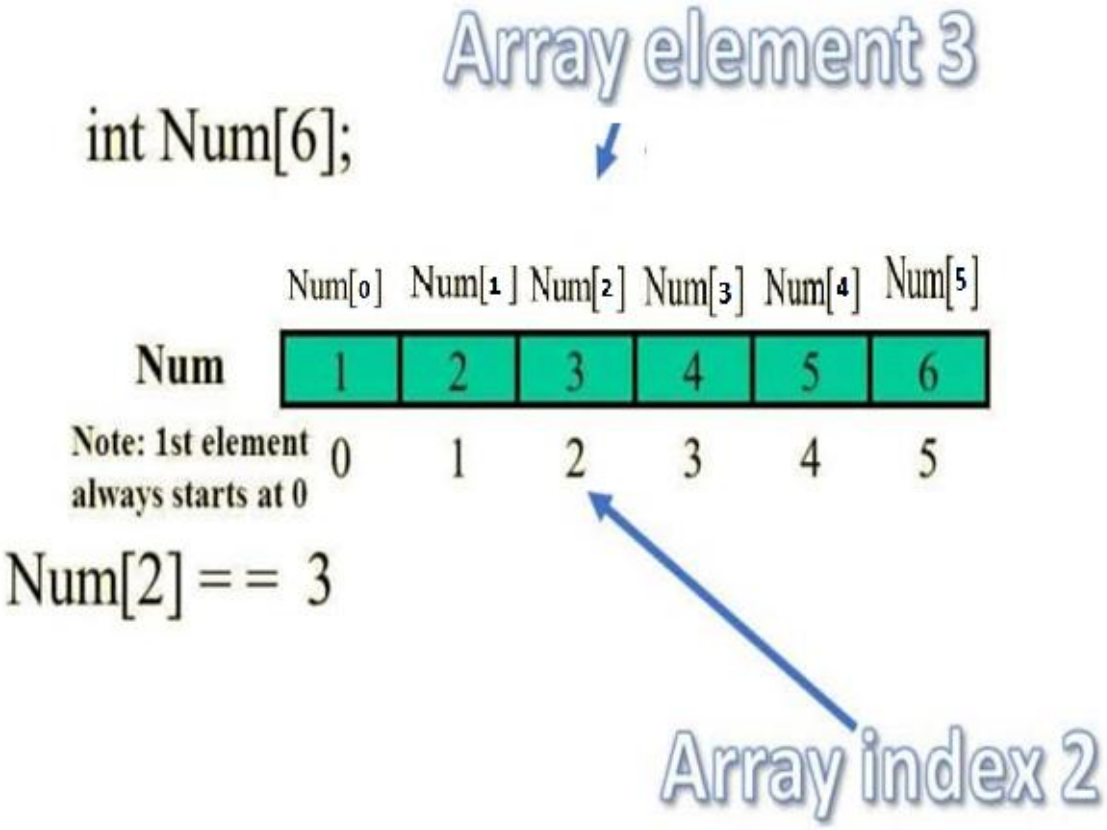
Array

- An array is a collection of variables of the same type.
- All arrays consist of contiguous memory locations.
- The lowest address corresponds to the first element and the highest address to the last element.
- Example: Instead of declaring individual variables, such as number0, number1, ..., and number99, we can declare one array variable such as numbers and use numbers[0], numbers[1], and ..., numbers[99] to represent individual variables




Array in C
array variable
index of the element
to be accessed
arr [0];
0
1
2
3
4
5
arr

Array



Array Name: Num
Type: int
Size: 6
Dimension: 1

Array element 3 is indexed at position 2

Declaring Arrays

Syntax

- type arrayName [arraySize];

Example

- double balance[10];

Here *balance* is an array which is sufficient to hold up to 10 double numbers.

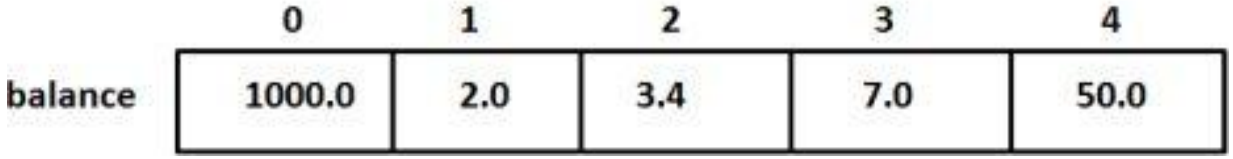
Initializing Arrays

Type-1

- double balance[5] = {1000.0, 2.0, 3.4, 7.0, 50.0};

Type-2

- double balance[] = {1000.0, 2.0, 3.4, 7.0, 50.0};



Assigning individual value

balance[4] = 50.0; // assigns 50 to 5th element of array

Accessing Array Elements

- double salary = balance[3];

Change Value of Array elements

- int mark[5] = {19, 10, 8, 17, 9}
- // make the value of the third element to -1
- mark[2] = -1;
- // make the value of the fifth element to 0
- mark[4] = 0;

Input and Output Array Elements

- Here's how you can take input from the user and store it in an array element
- // take input and store it in the 3rd element
- scanf("%d", &mark[2]);
- // take input and store it in the ith element
- scanf("%d", &mark[i-1]);

Here's how you can print an individual element of an array.

- // print the first element of the array
- printf("%d", mark[0]);
- // print the third element of the array
- printf("%d", mark[2]);
- // print ith element of the array
- printf("%d", mark[i-1]);

Program for declaration and initialization of array at compile time and displaying using loop

```c
#include<stdio.h>
int main(){
int i=0;
int marks[5];//declaration of array
marks[0]=80;//initialization of array
marks[1]=60;
marks[2]=70;
marks[3]=85;
marks[4]=75;
//traversal of array
for(i=0;i<5;i++){
printf("%d \n",marks[i]);
}//end of for loop
return 0;
}
```

Example: Program for declaration and initialization of array at compile time and displaying using loop

```c
#include<stdio.h>
int main(){
int i=0;
int marks[5]={20,30,40,50,60};
 //traversal of array
for(i=0;i<5;i++){
printf("%d \n",marks[i]);
}
return 0;
}
```

Example: Program for initialization of array at run time and displaying using loop

```c
#include<stdio.h>
int main(){
int i=0;
int marks[5];//declaration of array
printf("Enter the marks:");
scanf("%d",&marks[0]); //initialization of array
scanf("%d",&marks[1]);
scanf("%d",&marks[2]);
scanf("%d",&marks[3]);
scanf("%d",&marks[4]);
//traversal of array
for(i=0;i<5;i++){
printf("%d \n",marks[i]);
}//end of for loop
return 0;
}
```

Example of run time initialization and printing output using loop

```c
#include <stdio.h>
int main () {
   int n[ 10 ]; /* n is an array of 10 integers */
   int i,j;

   /* initialize elements of array n to 0 */
   for ( i = 0; i < 10; i++ ) {
      n[ i ] = i + 100; /* set element at location i to i + 100 */
   }
   /* output each array element's value */
   for (j = 0; j < 10; j++ ) {
      printf("Element[%d] = %d\n", j, n[j] );
   }
   return 0;
}
```

```
Output
Element[0] = 100
Element[1] = 101
Element[2] = 102
Element[3] = 103
Element[4] = 104
Element[5] = 105
Element[6] = 106
Element[7] = 107
Element[8] = 108
Element[9] = 109
```

Example 1: Array Input/Output

```
// Program to take 5 values from the user and store them in an array
// Print the elements stored in the array
#include <stdio.h>
int main() {
int values[5];
printf("Enter 5 integers: ");
// taking input and storing it in an array
for(int i = 0; i < 5; ++i) {
scanf("%d", &values[i]);
}
printf("Displaying integers: ");
// printing elements of an array
for(int i = 0; i < 5; ++i) {
printf("%d\n", values[i]);
}
return 0;
}
```

Example 2: Calculate Average

```c
// Program to find the average of n numbers using arrays

#include <stdio.h>
int main()
{
    int marks[10], i, n, sum = 0, average;

    printf("Enter number of elements: ");
    scanf("%d", &n);

    for(i=0; i<n; ++i)
    {
        printf("Enter number%d: ",i+1);
        scanf("%d", &marks[i]);

        // adding integers entered by the user to the sum variable
        sum += marks[i];
    }

    average = sum/n;
    printf("Average = %d", average);

    return 0;
}
```

Access elements out of its bound!

- Suppose you declared an array of 10 elements.

  int testArray[10];

- Now let's say if you try to access testArray[12].
- This may cause unexpected output (undefined behavior).
- Sometimes you might get an error and some other time your program may run correctly.
- Hence, you should never access elements of an array outside of its bound.

Basic Operations on Array

- **Traverse** − Access the array elements so that the data can be checked or used as part of a process.
- **Insertion** − Adds an element at the given index.
- **Deletion** − Deletes an element at the given index.
- **Search** − Searches an element using the given index or by the value.
- **Update** − Updates an element at the given index.

Traverse and Print

```
#include<stdio.h>
int main(){
int i=0;
int marks[5]={20,30,40,50,60};
 //traverse and print array elements
for(i=0;i<5;i++){
printf("%d \n",marks[i]);
}
return 0;
}
```

References

1. C programming by E Balaguruswami
2. Programming C by Y. kanitkar
3. Programming C by Denis Ritchie
4. NPTEL Lecture note of Dr. Partha Pratim Das, Department of Computer Science and Engineering, Indian Institute of Technology, Kharagpur.
5. https://docs.oracle.com/cd/E18752_01/html/817-6223/chp-typeopexpr-2.html
6. https://data-flair.training/blogs/escape-sequence-in-c/
7. Internet source