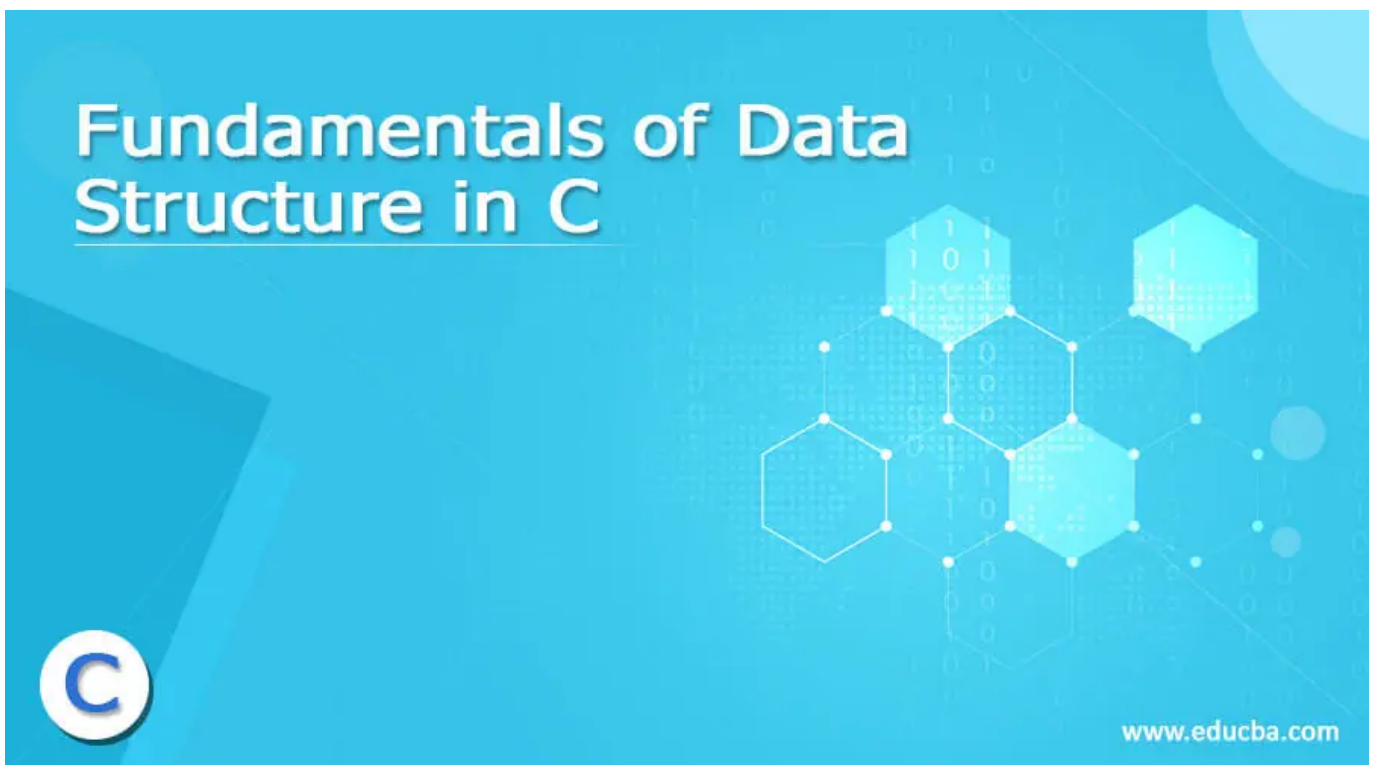




(<https://www.educba.com/software-development/>)

← (<https://www.educba.com/dfs-algorithm-in-c/>)

→ (<https://www.educba.com/linear-search-in-c/>)



## What are Fundamentals of Data Structure in C?

---

A data structure is a particular configuration of information for arranging and saving information so any client can easily get to and work inside specified data to run a program proficiently. PC memory data can be coordinated logically or numerically, and this interaction is





(<https://www.educba.com/software-development/>)

# Fundamentals of Data Structure in C

Now let's see the different fundamental concepts of data structure in c as follows.

## Start Your Free Software Development Course

Web development, programming languages, Software testing & others

## Characteristics

- **Linear:** In linear data structure we arrange the data in a sequential manner like array structure.
  - **Non-Linear:** In nonlinear data structure we arrange the data in a non-sequential manner like graph and tree structure.
  - **Static:** It is a static data structure that depicts the size and structure of a collection of data values related to a memory area at assemble time that is fixed. For example- Array.
  - **Homogenous:** It is a quality of data structures addressing whether the data types of all components are the same for example Array.
  - **Non-Homogenous:** It is a quality of data structures addressing whether the data types of all components are not the same.
  - **Dynamic:** It is a dynamic data structure that characterizes the contracting and growing of data items at the run time or the program's execution. It is likewise identified with the usage of the memory area that can be changed at the program's run time for example Linked List.
- It has a few principles that characterize how the data items are identified with another.
  - It characterizes a few guidelines to show the relationship between data items and





(<https://www.educba.com/software-development/>)

- **Time Complexity:** It is used to define how much time is required for the execution of a program and it should be less as possible.
- **Space Complexity:** It is used to define how much memory space is required for the execution of a program and it should be less as possible.

## Types of Data Structure

Basically, there are two types of data structure.

1. Primitive Data Structure
2. Non Primitive Data Structure

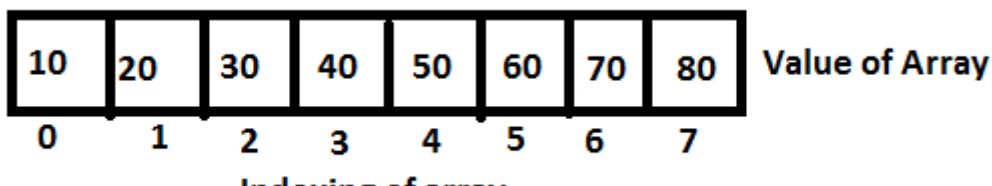
### Primitive Data Structure

Basically Primitive Data types directly work with the machine instruction and it has different data types for different machines such as integer, float, character, string, constant and pointer.

### Non Primitive Data Structure

This is a complex data structure as compared to the primitive data structure. It works on clusters or we can say that grouping the same or different data values and it includes the following data structure as follows.

**Array:** Normally an array is a collection of values that are stored in a sequential manner in a memory location. If we need to store the multiple values that have the same data type at that time we can use an array data structure. The array data structure we illustrate by using the following figure as follows.





(<https://www.educba.com/software-development/>)

Popular Course in this category



## C Programming Training (3 Courses, 5 Project)

3 Online Courses | 5 Hands-on Projects | 34+ Hours | Verifiable Certificate of Completion | Lifetime Access

★★★★★ 4.5 (8,644 ratings)

Course Price

**\$79** ~~\$399~~

[View Course](#)

(<https://www.educba.com/software-development/courses/c-programming-course/?btnz=edu-blg-inline-banner1>)

### Related Courses

C++ Training (4 Courses, 5 Projects, 4 Quizzes) (<https://www.educba.com/software-development/courses/c-course/?btnz=edu-blg-inline-banner1>)

Java Training (40 Courses, 29 Projects, 4 Quizzes) (<https://www.educba.com/software-development/courses/java-course/?btnz=edu-blg-inline-banner1>)

**List:** List we divide into two different categories as follows.

1. **Linear List**
2. **Non Linear List**

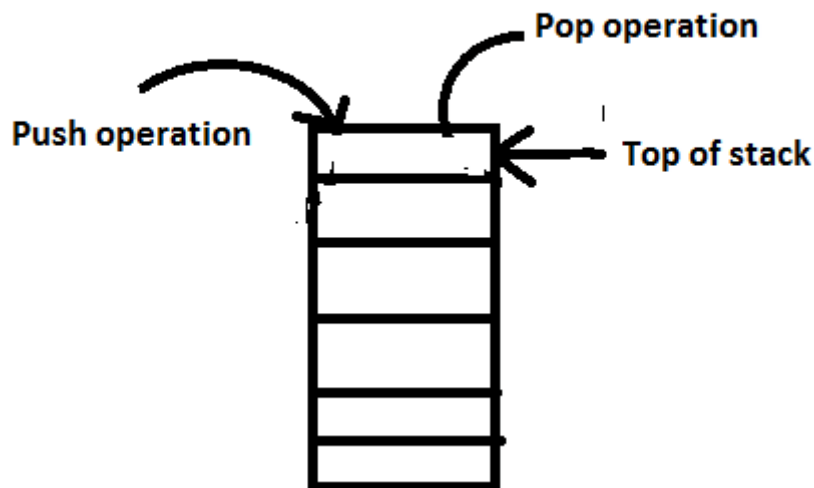
**Linear List:** Linear data structure can further be divided into two parts as follows.



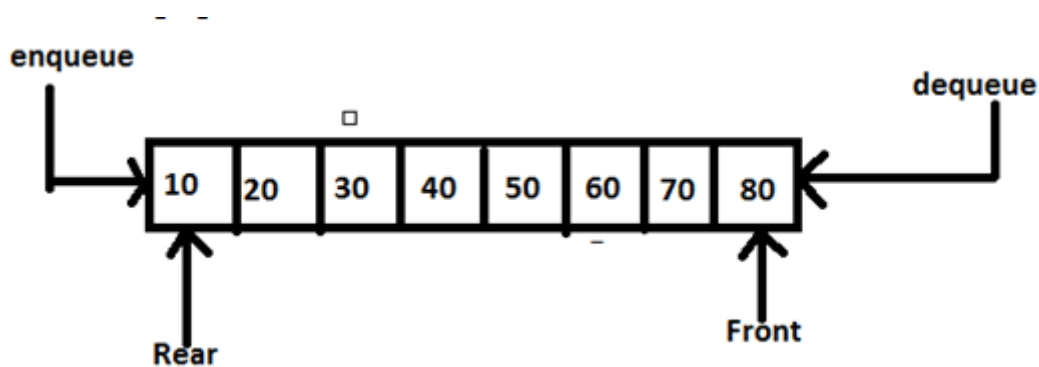
- **Stack:** Stack is one type of data structure in which we can store the data element. On the stack, we can perform the two types of operation such as push and pop. Pushing push



(<https://www.educba.com/software-development/>)



2. **Queue:** Queue is also a linear data structure in which we can add elements from the **rear side** and we can remove elements from the **front side** of the queue. The queue works as **FIFO** manner means (First In First Out). The queue data structure we illustrate by using the following figure as follows.



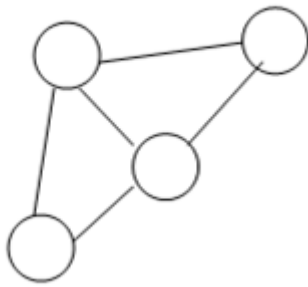
**Non Linear List:** Non linear list means there is no sequence to store the data; in other words, we can say that every element has multiple paths. The Non linear list also has two categories as follows.



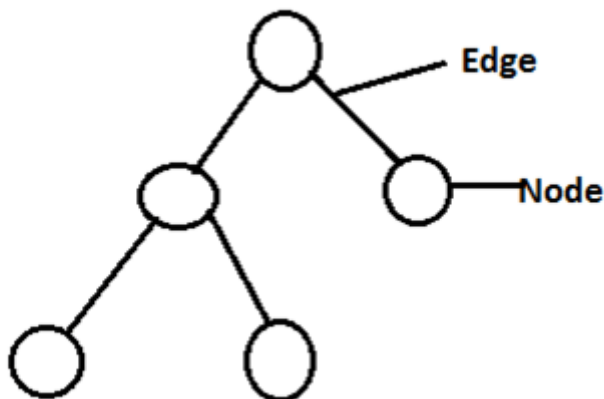


[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

Graph has different terminology. The graph data structure we illustrate by using the following figure as follows.



- **Trees:** A tree is a nonlinear data structure, in which nodes are connected by using edges and it maintains the hierarchical data structure. See other data structures work in a sequential manner, suppose we need to perform any operation that increases the complexity of structure, so that is the reason we use tree data structure. The tree data structure we illustrate by using the following figure as follows.



## Example of Fundamentals of Data Structure in



Now let's see a basic example in a data structure as follows.



[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

```
#include <stdio.h>

#define MAX 50

void queue_insert();
void queue_delete();
void queue_display();
int que_array[MAX];
int que_rear = - 1;
int que_front = - 1;

main()
{
    int ch;
    while (1)
    {
        printf("1.Add Data Element \n");
        printf("2.Delete Data element  \n");
        printf("3.Display Data elements \n");
        printf("4.Quit \n");
        printf("Enter choice : ");
        scanf("%d", &ch);
        switch (ch)
        {
            case 1:

                queue_insert();
                break;

            case 2:
```





[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

```
break,  
case 4:  
exit(1);  
default:  
printf("Choice is incorrect \n");  
}  
}  
}  
void queue_insert()  
{  
int add_element;  
if (que_rear == MAX - 1)  
printf("Now Queue is Overflow \n");  
else  
{  
if (que_front == - 1)  
que_front = 0;  
printf("Enter data element : ");  
scanf("%d", &add_element);  
que_rear = que_rear + 1;  
que_array[que_rear] = add_element;  
}  
}  
void queue_delete()  
{  
if (que_front == - 1 || que_front > que_rear)
```







(<https://www.educba.com/software-development/>)

```

}
{
printf("Data Element deleted: %d\n", que_array[que_front]);
que_front = que_front + 1;
}
}
void queue_display()
{
int a;
if (que_front == - 1)
printf("Empty Queue \n");
else
{
printf("Queue: \n");
for (a = que_front; a <= que_rear; a++)
printf("%d ", que_array[a]);
printf("\n");
}
}
}

```

**Explanation:** By using the above example we try to implement the queue in the data structure by using C programming. The final output of the above statement we illustrate by using the following snapshot.

```

1.Add Data Element
2.Delete Data element
3.Display Data elements
4.Quit
Enter choice : 1
Enter data element : 10

```





(<https://www.educba.com/software-development/>)

```
3.Display Data elements
4.Quit
Enter choice : 1
Enter data element : 30
1.Add Data Element
2.Delete Data element
3.Display Data elements
4.Quit
Enter choice : 3
Queue:
10 20 30
1.Add Data Element
2.Delete Data element
3.Display Data elements
4.Quit
```

```
Enter choice : 2
Data Element deleted: 10
1.Add Data Element
2.Delete Data element
3.Display Data elements
4.Quit
Enter choice : 3
Queue:
20 30
1.Add Data Element
2.Delete Data element
3.Display Data elements
4.Quit
Enter choice : 4

...Program finished with exit code 1
Press ENTER to exit console.
```

## Conclusion

We hope from this article you learn the Fundamentals of data structure in C. From the article, we have learned the basic theory of Fundamentals of data structure and we also see different examples of Fundamentals of data structure. From this article, we learned how and





[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

different fundamental concepts of data structure in C along with examples and its code implementation. You may also have a look at the following articles to learn more –

1. [Hash Table in C \(https://www.educba.com/hash-table-in-c/\)](https://www.educba.com/hash-table-in-c/)
2. [Selection sort in C \(https://www.educba.com/selection-sort-in-c/\)](https://www.educba.com/selection-sort-in-c/)
3. [Bubble Sort in C \(https://www.educba.com/bubble-sort-in-c/\)](https://www.educba.com/bubble-sort-in-c/)
4. [new in C++ \(https://www.educba.com/new-in-c-plus-plus/\)](https://www.educba.com/new-in-c-plus-plus/)

## ALL IN ONE SOFTWARE DEVELOPMENT BUNDLE (600+ COURSES, 50+ PROJECTS)

- ☒ 600+ Online Courses
- ☒ 50+ projects
- ☒ 3000+ Hours
- ☒ Verifiable Certificates
- ☒ Lifetime Access

**Learn More**

<https://www.educba.com/software-development/courses/software-development-course/?btnz=edu-blg-inline-banner3>





[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

Who is EDUCBA? (<https://www.educba.com/about-us/?source=footer>)

Sign Up (<https://www.educba.com/software-development/signup/?source=footer>)

Corporate Training (<https://www.educba.com/corporate/?source=footer>)

Certificate from Top Institutions (<https://www.educba.com/educbalive/?source=footer>)

Contact Us (<https://www.educba.com/contact-us/?source=footer>)

Verifiable Certificate (<https://www.educba.com/software-development/verifiable-certificate/?source=footer>)

Reviews (<https://www.educba.com/software-development/reviews/?source=footer>)

Terms and Conditions (<https://www.educba.com/terms-and-conditions/?source=footer>)

Privacy Policy (<https://www.educba.com/privacy-policy/?source=footer>)

## Apps

iPhone & iPad (<https://itunes.apple.com/in/app/educba-learning-app/id1341654580?mt=8>)

Android (<https://play.google.com/store/apps/details?id=com.educba.www>)

## Resources

Free Courses (<https://www.educba.com/software-development/free-courses/?source=footer>)

Java Tutorials (<https://www.educba.com/software-development/software-development-tutorials/java-tutorial/?source=footer>)

Python Tutorials (<https://www.educba.com/software-development/software-development-tutorials/python-tutorial/?source=footer>)





[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

source=footer)

Software Development Course - All in One Bundle  
(<https://www.educba.com/software-development/courses/software-development-course/?source=footer>)

Become a Python Developer (<https://www.educba.com/software-development/courses/python-certification-course/?source=footer>)

Java Course (<https://www.educba.com/software-development/courses/java-course/?source=footer>)

Become a Selenium Automation Tester (<https://www.educba.com/software-development/courses/selenium-training-certification/?source=footer>)

Become an IoT Developer (<https://www.educba.com/software-development/courses/iot-course/?source=footer>)

ASP.NET Course (<https://www.educba.com/software-development/courses/asp-net-course/?source=footer>)

VB.NET Course (<https://www.educba.com/software-development/courses/vb-net-course/?source=footer>)

PHP Course (<https://www.educba.com/software-development/courses/php-course/?source=footer>)

© 2022 - EDUCBA. ALL RIGHTS RESERVED. THE CERTIFICATION NAMES ARE THE TRADEMARKS OF THEIR RESPECTIVE OWNERS.

