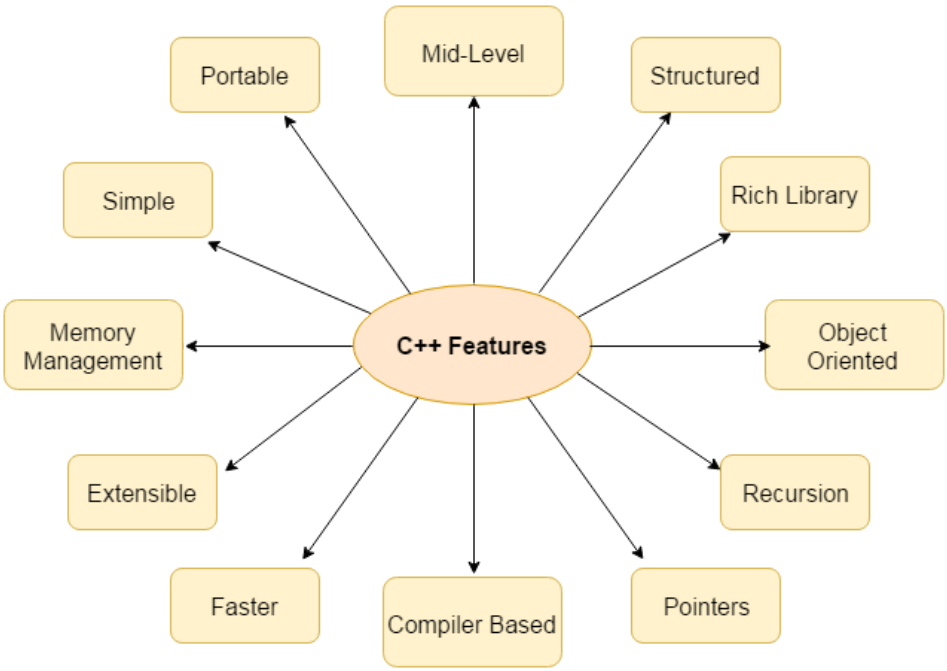
C vs. C++

No.	C	C++
1)	C follows the **procedural style programming.**	C++ is multi-paradigm. It supports both **procedural and object oriented.**
2)	Data is less secured in C.	In C++, you can use modifiers for class members to make it inaccessible for outside users.
3)	C follows the **top-down approach.**	C++ follows the **bottom-up approach.**
4)	C does not support function overloading.	C++ supports function overloading.
5)	In C, you can't use functions in structure.	In C++, you can use functions in structure.
6)	C does not support reference variables.	C++ supports reference variables.
7)	In C, **scanf() and printf()** are mainly used for input/output.	C++ mainly uses stream **cin and cout** to perform input and output operations.
8)	Operator overloading is not possible in C.	Operator overloading is possible in C++.
9)	C programs are divided into **procedures and modules**	C++ programs are divided into **functions and classes.**
10)	C does not provide the feature of namespace.	C++ supports the feature of namespace.
11)	Exception handling is not easy in C. It has to perform using other functions.	C++ provides exception handling using Try and Catch block.
12)	C does not support the inheritance.	C++ supports inheritance.

C++ Features



C++ Basic Program

```
1. #include <iostream>   // standard input output file (cin,cout)
2. using namespace std;
3. int main() {
4.    cout << "Hello C++ Programming";
5.    return 0;
6. }
```

There are 4 types of data types in C++ language.

Types	Data Types
Basic Data Type	int, char, float, double, etc
Derived Data Type	array, pointer, etc
Enumeration Data Type	enum
User Defined Data Type	structure

C++ Keywords

← Prev Next →

A keyword is a reserved word. You cannot use it as a variable name, constant name etc. **A list of 32 Keywords in C++ Language which are also available in C language are given below.**

auto	break	case	char	const	continue	default	do
double	else	enum	extern	float	for	goto	if
int	long	register	return	short	signed	sizeof	static
struct	switch	typedef	union	unsigned	void	volatile	while

What are the keywords?

Keywords are the reserved words that have a special meaning to the compiler. They are reserved for a special purpose, which cannot be used as the identifiers. For example, 'for', 'break', 'while', 'if', 'else', etc. are the predefined words where predefined words are those words whose meaning is already known by the compiler. Whereas, the identifiers are the names which are defined by the programmer to the program elements such as variables, functions, arrays, objects, classes.

Identifiers	Keywords
Identifiers are the names defined by the programmer to the basic elements of a program.	Keywords are the reserved words whose meaning is known by the compiler.
It is used to identify the name of the variable.	It is used to specify the type of entity.
It can consist of letters, digits, and underscore.	It contains only letters.
It can use both lowercase and uppercase letters.	It uses only lowercase letters.
No special character can be used except the underscore.	It cannot contain any special character.
The starting letter of identifiers can be lowercase, uppercase or underscore.	It can be started only with the lowercase letter.
It can be classified as internal and external identifiers.	It cannot be further classified.
Examples are test, result, sum, power, etc.	Examples are 'for', 'if', 'else', 'break', etc.

C++ IF-else-if ladder Statement

The C++ if-else-if ladder statement executes one condition from multiple statements.

```
1. if(condition1){
2. //code to be executed if condition1 is true
3. }else if(condition2){
4. //code to be executed if condition2 is true
5. }
6. else if(condition3){
7. //code to be executed if condition3 is true
8. }
9. ...
10.else{
11.//code to be executed if all the conditions are false
12.}
```

C++ For LOOP

```
for(initialization; condition; incr/decr){
//code to be executed    }
```

CPP While LOOP

```
while(condition){
//code to be executed    }
```

C++ Do-While Loop

The C++ do-while loop is used to iterate a part of the program several times. If the number of iteration is not fixed and you must have to execute the loop at least once, it is recommended to use do-while loop.

The C++ do-while loop is executed at least once because condition is checked after loop body.

```
#include <iostream>
using namespace std;
int main() {
    int i = 1;
        do{
            cout<<i<<"\n";
            i++;
        } while (i <= 10) ;
}
```

C++ Goto Statement

The C++ goto statement is also known as jump statement. It is used to transfer control to the other part of the program. It unconditionally jumps to the specified label.

It can be used to transfer control from deeply nested loop or switch case label.

```
#include <iostream>
using namespace std;
int main()
{
ineligible:
      cout<<"You are not eligible to vote!\n";
    cout<<"Enter your age:\n";
    int age;
    cin>>age;
    if (age < 18){
        goto ineligible;
    }
    else
    {
        cout<<"You are eligible to vote!";
    }
}
```

Output:

```
You are not eligible to vote!
Enter your age:
16
You are not eligible to vote!
Enter your age:
7
You are not eligible to vote!
Enter your age:
22
You are eligible to vote!
```

Difference between call by value and call by reference in C++

No.	Call by value	Call by reference
1	A copy of value is passed to the function	An address of value is passed to the function
2	Changes made inside the function is not reflected on other functions	Changes made inside the function is reflected outside the function also
3	Actual and formal arguments will be created in different memory location	Actual and formal arguments will be created in same memory location

← Prev Next →

C++ Storage Classes

Storage class is used to define the lifetime and visibility of a variable and/or function within a C++ program.

Lifetime refers to the period during which the variable remains active and visibility refers to the module of a program in which the variable is accessible.

There are five types of storage classes, which can be used in a C++ program

1. Automatic
2. Register
3. Static
4. External
5. Mutable

Storage Class	Keyword	Lifetime	Visibility	Initial Value
Automatic	auto	Function Block	Local	Garbage
Register	register	Function Block	Local	Garbage
Mutable	mutable	Class	Local	Garbage
External	extern	Whole Program	Global	Zero
Static	static	Whole Program	Local	Zero

ARRAYS IN CPP

```
1.  int main()
2.  {
3.   int arr[5]={10, 0, 20, 0, 30};  //creating and initializing array
4.         for (int i = 0; i < 5; i++)
5.         {   cout<<arr[i]<<"\n";  }
6.  }
```

TRAVERSING USING FOR EACH LOOP

```
1.  int main()
2.  {
3.   int arr[5]={10, 0, 20, 0, 30}; //creating and initializing array
4.        for (int i: arr)
5.         {   cout<<i<<"\n";    }
6.  }
```

2D ARRAYS IN C++

```
int test[3][3];  //declaration of 2D array
```

```
1.    int test[3][3] =
2.     {  {2, 5, 5},
3.        {4, 0, 3},
4.        {9, 1, 8}  };  //declaration and initialization of 2D array
```

Passing Arrays in functions

```
int fun(int arr[]){ // square brackets are used here.
        //Code
}
arr = {1,2,3,4,5}
fun(arr)
```

NOTE : Square brackets '[]' are not used at thetime of function call.

EXTRAS:

If arr[10] = {1,2,3,4,5}
Then at index 5 to 9 we have 0 values if accessed.

Size of Operator (tells total space holded by array.)
int size = sizeof(myArray);

C++ Pointers

The pointer in C++ language is a variable, it is also known as locator or indicator that points to an address of a value.

Advantage of pointer

1) Pointer reduces the code and improves the performance, it is used to retrieving strings, trees etc. and used with arrays, structures and functions.

2) We can return multiple values from a function using pointer.

3) It makes you able to access any memory location in the computer's memory.

Usage of pointers

1) Dynamic memory allocation

In C language, we can dynamically allocate memory using malloc() and calloc() functions where a pointer is used.

2) Arrays, Functions and Structures

Pointers in c language are widely used in arrays, functions and structures. It reduces the code and improves the performance.