Why Did You Choose CSE?

Introduce With Structural Programming Language

And Other Related Terms

Course Title :- Structured Programming Language Sessional

Course Code :- CSE-122

Level Term: 1-II-A(G1) & 1-II-B(G3,G4)

Khandaker Jannatul Ritu, Lecturer(CSE), BAIUST

Content Of This Lecture

Resources will be followed:

Books:

1. **C: the complete reference**
2. **Programming in ANSI-C**
3. **C programming by tamim shahriar subin**
4. **52 Programming Problem Tamim Shahriar Subeen**

Websites:

1. https://www.w3schools.com/c/index.php
2. https://www.programiz.com/c-programming
3. https://www.javatpoint.com/c-programming-language-tutorial
4. https://www.geeksforgeeks.org/c-programming-language/
5. https://www.tutorialspoint.com/cprogramming/index.htm

YouTube Tutorials to follow(click on the link):-

C Programming Bangla Tutorials (সবার জন্য সি প্রোগ্রামিং) । Updated in 2023 - YouTube

Popular Programming languages

- 1. Javascript
  2. Python
  3. Go
  4. Java
  5. Kotlin
  6. PHP
  7. C#
  8. Swift
  9. R
  10. Ruby
  11. C and C++

Necessary Websites & Tools Related To Programming

GeeksforGeeks

javaTpoint

w3schools

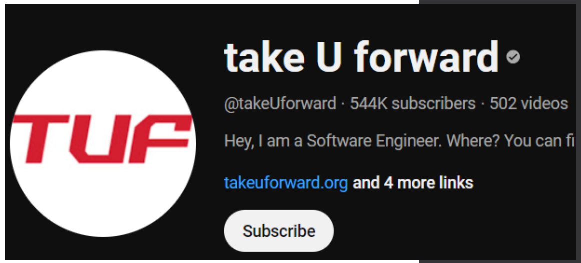
freeCodeCamp(🔥)

tutorialspoint
SIMPLY EASY LEARNING

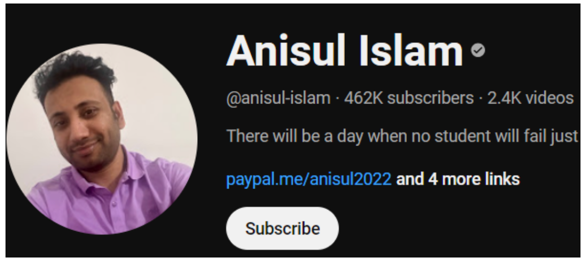
programiz

USACO

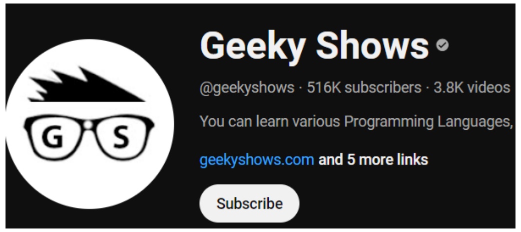
শাফায়েতের ব্লগ
প্রোগ্রামিং, অ্যালগরিদম, ব্যাকএন্ড ইঞ্জিনিয়ারিং



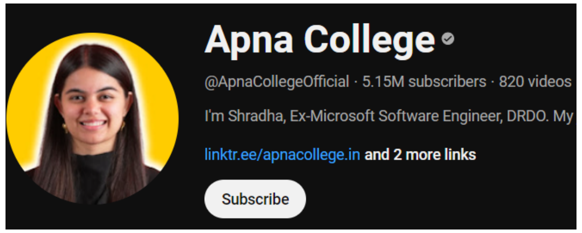
take U forward
@takeUforward · 544K subscribers · 502 videos
Hey, I am a Software Engineer. Where? You can fi
takeuforward.org and 4 more links
Subscribe



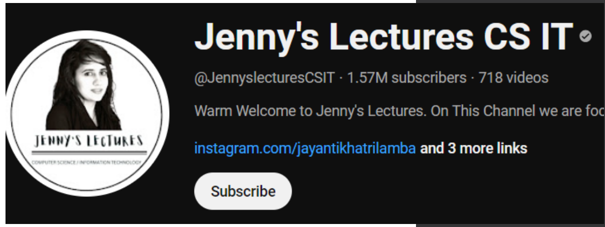
Anisul Islam
@anisul-islam · 462K subscribers · 2.4K videos
There will be a day when no student will fail just
paypal.me/anisul2022 and 4 more links
Subscribe



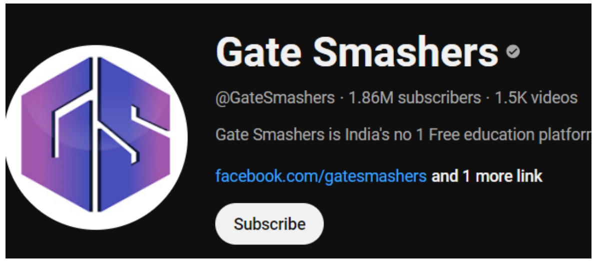
Geeky Shows
@geekyshows · 516K subscribers · 3.8K videos
You can learn various Programming Languages,
geekyshows.com and 5 more links
Subscribe



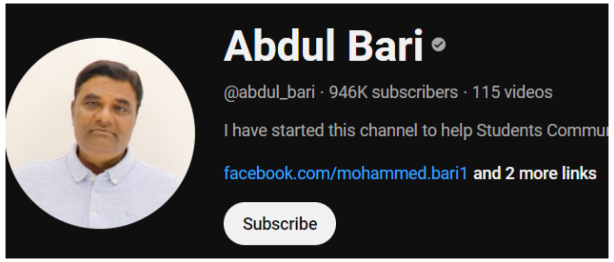
Apna College
@ApnaCollegeOfficial · 5.15M subscribers · 820 videos
I'm Shradha, Ex-Microsoft Software Engineer, DRDO. My
linktr.ee/apnacollege.in and 2 more links
Subscribe



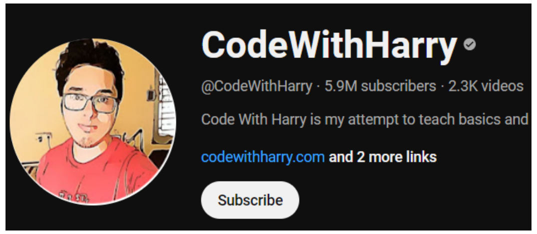
Jenny's Lectures CS IT
@JennyslecturesCSIT · 1.57M subscribers · 718 videos
Warm Welcome to Jenny's Lectures. On This Channel we are foc
instagram.com/jayantikhatrilamba and 3 more links
Subscribe



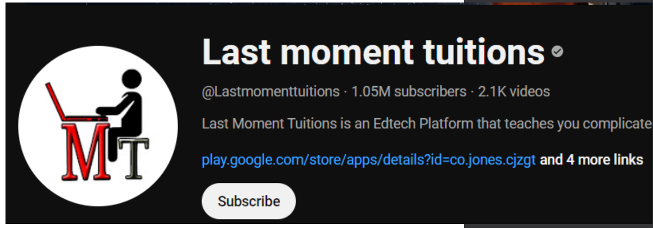
Gate Smashers
@GateSmashers · 1.86M subscribers · 1.5K videos
Gate Smashers is India's no 1 Free education platforr
facebook.com/gatesmashers and 1 more link
Subscribe



Abdul Bari
@abdul_bari · 946K subscribers · 115 videos
I have started this channel to help Students Commu
facebook.com/mohammed.bari1 and 2 more links
Subscribe



CodeWithHarry
@CodeWithHarry · 5.9M subscribers · 2.3K videos
Code With Harry is my attempt to teach basics and
codewithharry.com and 2 more links
Subscribe



Last moment tuitions
@Lastmomenttuitions · 1.05M subscribers · 2.1K videos
Last Moment Tuitions is an Edtech Platform that teaches you complicate
play.google.com/store/apps/details?id=co.jones.cjzgt and 4 more links
Subscribe

Competitive Programming OJ

CODEFORCES

LeetCode

AtCoder

beecrowd

hackerearth

CODECHEF

An unacademy Educational Initiative

SPOJ.com

Dimik OJ

LightOJ

Toph

Necessary Websites & Tools Related To Programming

GitHub

STALK

stackoverflow

Quora

LinkedIn

Gmail

ideone.com

Your great ideas will be born here

Programiz

C Online Compiler

R^G

Google Scholar

Structural programming language

- **Structured Programming**
- Structured Programming is a type of programming that generally converts large or complex programs into more manageable and small pieces of code.
- These small pieces of codes are usually known as functions or modules or sub-programs of large complex programs.
- It is known as modular programming and minimizes the chances of function affecting another.
- Exmaples: C,C++, Java, and Python

```
// C program to demonstrate the structured
programming
#include <stdio.h>

// Function for addition
int sum(int a, int b){
        return a + b;
}

int main(){
        int a = 10, b = 5; // Variable initialisation
        int add, minus;

        add = sum(a, b); // Function Call

        printf("Addition = %d\n", add);
}
```

Unstructured Programming:

- Unstructured Programming is a type of programming that generally executes in sequential order i.e., these programs just not jumped from any line of code and each line gets executed sequentially.
- It is also known as non-structured programming that is capable of creating turning-complete algorithms.

```
// C program to demonstrate the unstructured
programming

#include <stdio.h>

int main()
{
        // Variable initialisation
        int a = 10, b = 5;
        int add, minus;

        // Operations performed
        add = a + b;
        minus = a - b;

        printf("Addition = %d\n", add);
        printf("Subtraction = %d\n", minus);
        return 0;
}
```

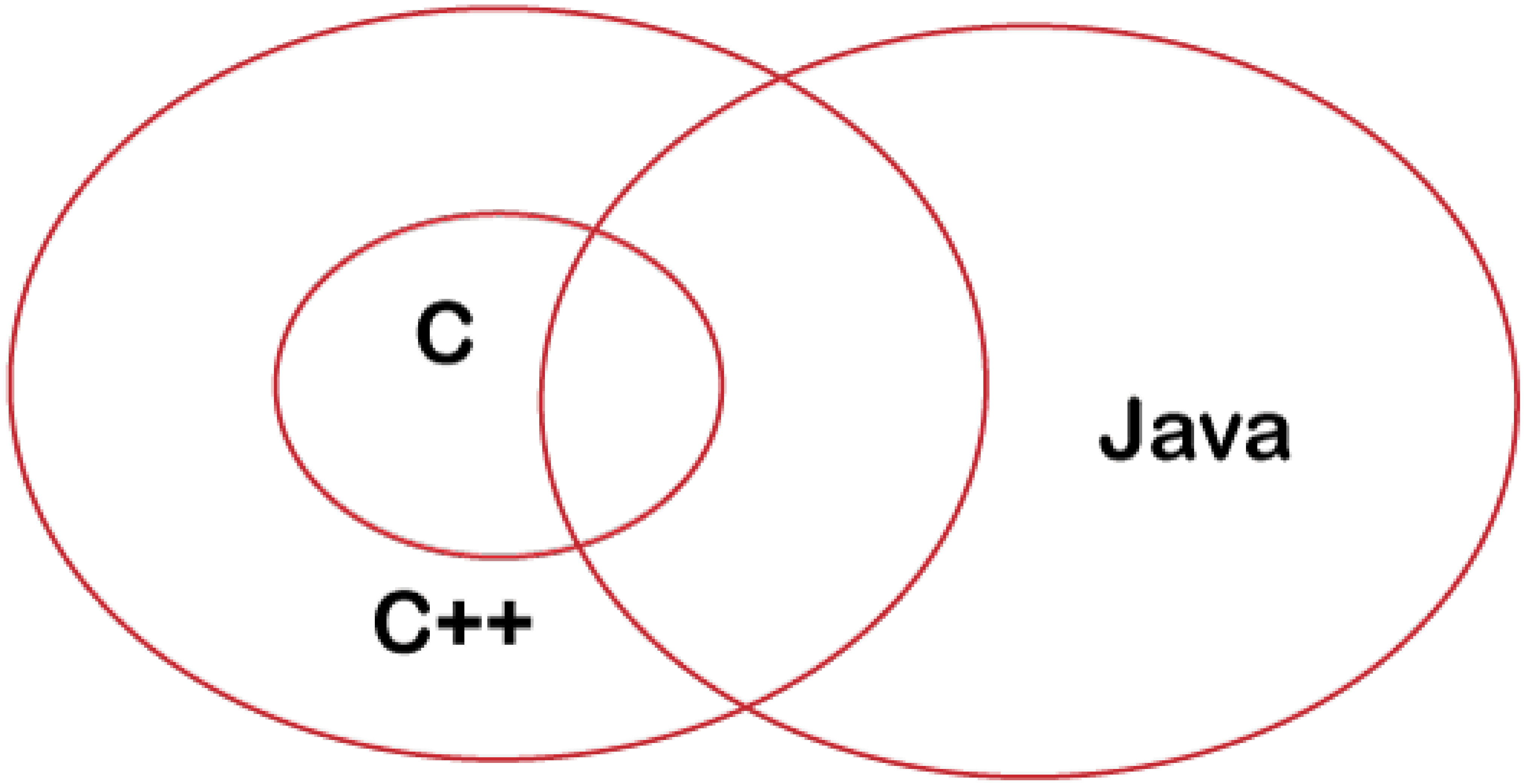
Tabular Difference Between Structured Vs Unstructured Programming:

Structured Programming	**Unstructured Programming**
It is basically a subset of procedural programs.	It is basically a procedural program.
In this, programmers are allowed to code a program simply by dividing the program into modules or smaller units.	In this, programmers are not allowed code divide programs into small units. Instead, the program should be written as a single continuous block without any breakage.
It is more user-friendly and easy to understand as compared to unstructured programming.	It is less user-friendly and little hard to understand as compared to structured programming.
It is easier to learn and follow.	It is difficult to learn and follow

Differences Between C, C++ and Java

S.N.	Basis	C	C++	Java
1	Origin	The C language is based on BCPL.	The C++ language is based on the C language.	The Java programming language is based on both C and C++.
2	Programming Pattern	It is a procedural language.	It is an object-oriented programming language.	It is a pure object-oriented programming language.
3	Approach	It uses the top-down approach.	It uses the bottom-up approach.	It also uses the bottom-up approach.
4	Dynamic or Static	It is a static programming language.	It is also a static programming language.	It is a dynamic programming language.
5	Code Execution	The code is executed directly.	The code is executed directly.	The code is executed by the JVM.
6	Platform Dependency	It is platform dependent.	It is platform dependent.	It is platform-independent because of byte code.
7	Translator	It uses a compiler only to translate the code into machine language.	It also uses a compiler only to translate the code into machine language.	Java uses both compiler and interpreter and it is also known as an interpreted language.
8	File Generation	It generates the .exe, and .bak, files.	It generates .exe file.	It generates .class file.
9	Number of Keyword	There are 32 keywords in the C language.	There are 60 keywords in the C++ language.	There are 52 keywords in the Java language.
10	Source File Extension	The source file has a .c extension.	The source file has a .cpp extension.	The source file has a .java extension.

Differences Between C, C++ and Java



C Programming Language

- The C Language is developed by Dennis Ritchie for creating system applications that directly interact with the hardware devices such as drivers, kernels, etc.
- C programming is considered as the base for other programming languages, that is why it is known as mother language.
- *It can be defined by the following ways:*

1. Mother language
2. System programming language
3. Procedure-oriented programming language
4. Structured programming language
5. Mid-level programming language

1) C as a mother language

C language is considered as the mother language of all the modern programming languages because **most of the compilers, JVMs, Kernels, etc. are written in C language**, and most of the programming languages follow C syntax, for example, C++, Java, C#, etc.
It provides the core concepts like the array, strings, functions, file handling, etc. that are being used in many languages like C++, Java, C#, etc.

2) C as a system programming language

A system programming language is used to create system software. C language is a system programming language because it **can be used to do low-level programming (for example driver and kernel)**. It is generally used to create hardware devices, OS, drivers, kernels, etc. For example, Linux kernel is written in C.
It can't be used for internet programming like Java, .Net, PHP, etc.

C Programming Language

3) C as a procedural language

A procedure is known as a function, method, routine, subroutine, etc. A procedural language **specifies a series of steps for the program to solve the problem**.

A procedural language breaks the program into functions, data structures, etc.

C is a procedural language. In C, variables and function prototypes must be declared before being used.

4) C as a structured programming language

A structured programming language is a subset of the procedural language. **Structure means to break a program into parts or blocks** so that it may be easy to understand.

In the C language, we break the program into parts using functions. It makes the program easier to understand and modify.

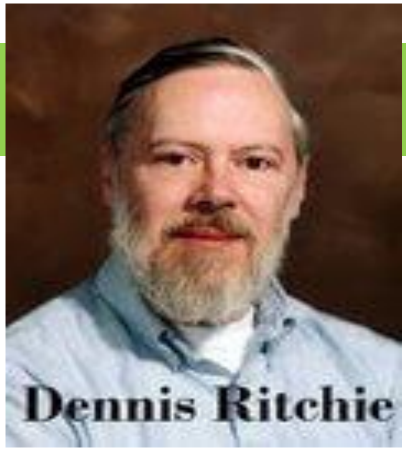
5) C as a mid-level programming language

C is considered as a middle-level language because it **supports the feature of both low-level and high-level languages**. C language program is converted into assembly code, it supports pointer arithmetic (low-level), but it is machine independent (a feature of high-level).

A **Low-level language** is specific to one machine, i.e., machine dependent. It is machine dependent, fast to run. But it is not easy to understand.

A **High-Level language** is not specific to one machine, i.e., machine independent. It is easy to understand.

History of C Language



- **C programming language** was developed in 1972 by Dennis Ritchie at bell laboratories of AT&T (American Telephone & Telegraph), located in the U.S.A.
- **Dennis Ritchie** is known as the **founder of the c language**.
- It was developed to overcome the problems of previous languages such as B, BCPL, etc.
- Initially, C language was developed to be used in **UNIX operating system**. It inherits many features of previous languages such as B and BCPL.
- Let's see the programming languages that were developed before C language.

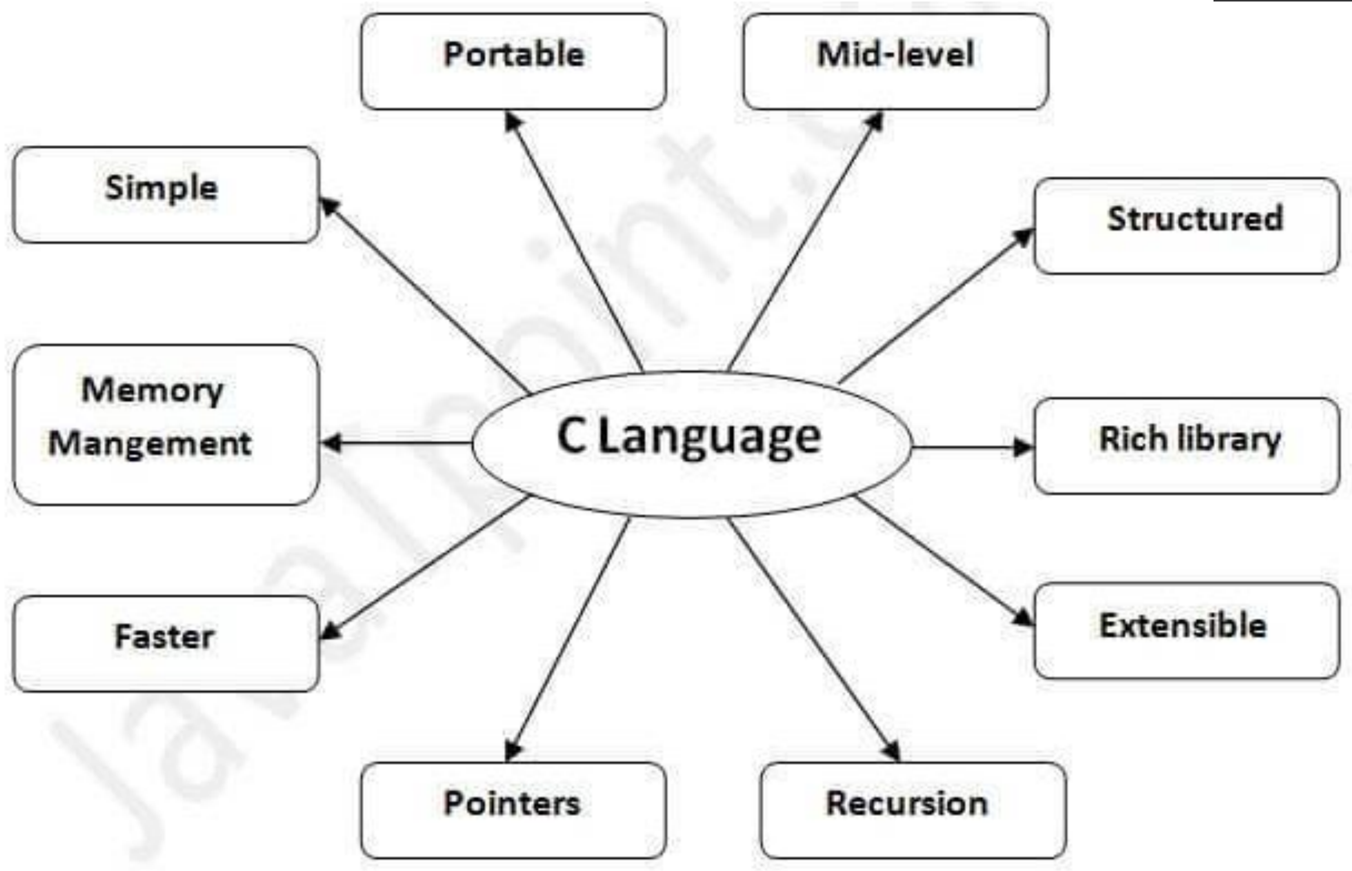
Language	Year	Developed By
Algol	1960	International Group
BCPL	1967	Martin Richard
B	1970	Ken Thompson
Traditional C	1972	Dennis Ritchie
K & R C	1978	Kernighan & Dennis Ritchie
ANSI C	1989	ANSI Committee
ANSI/ISO C	1990	ISO Committee
C99	1999	Standardization Committee

Advantages/Features of C Language

- C is the widely used language.

It provides many **features** that are given below.

1. **Simple**
2. **Machine Independent or Portable**
3. **Mid-level programming language**
4. **structured programming language**
5. **Rich Library**
6. **Dynamic Memory Management**
7. **Fast Speed**
8. **Pointers**
9. **Recursion**
10. **Extensible**



Drawbacks of C Language

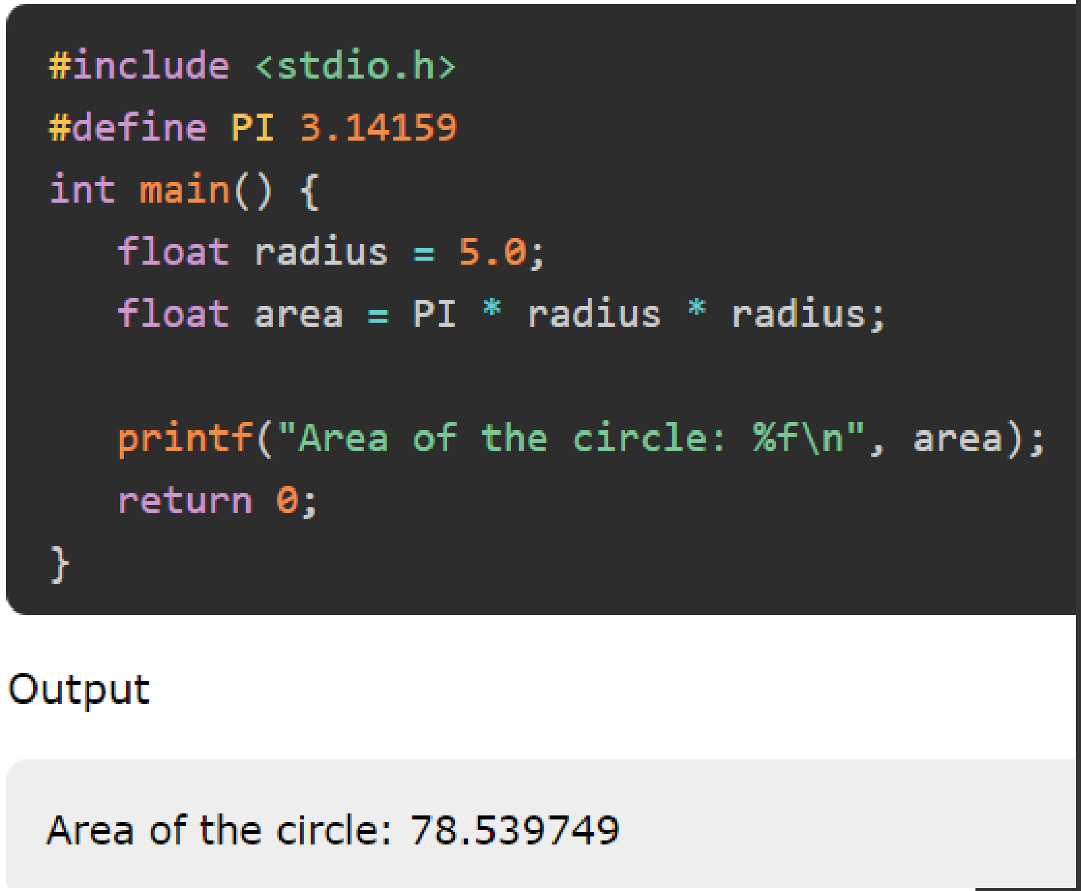
- The following are the disadvantages/drawbacks of C language –
- **Manual Memory Management** – C languages need manual memory management, where a developer has to take care of allocating and deallocating memory explicitly.
- **No Object–Oriented Feature** – Nowadays, most of the programming languages support the OOPs features. But C language does not support it.
- **No Garbage Collection** – C language does not support the concept of Garbage collection. A developer needs to allocate and deallocate memory manually and this can be error-prone and lead to memory leaks or inefficient memory usage.
- **No Exception Handling** – C language does not provide any library for handling exceptions. A developer needs to write code to handle all types of expectations.

Applications of C Language

- **System Programming** – C language is used to develop system software which are close to hardware such as operating systems, firmware, language translators, etc.
- **Embedded Systems** – C language is used in embedded system programming for a wide range of devices such as microcontrollers, industrial controllers, etc.
- **Compiler and Interpreters** – C language is very common to develop language compilers and interpreters.
- **Database Systems** – Since C language is efficient and fast for low-level memory manipulation. It is used for developing DBMS and RDBMS engines.
- **Networking Software** – C language is used to develop networking software such as protocols, routers, and network utilities.
- **Game Development** – C language is widely used for developing games, gaming applications, and game engines.
- **Scientific and Mathematical Applications** – Applications such as simulations, numerical analysis, and other scientific computations are usually developed in C language.
- **Text Editor and IDEs** – C language is used for developing text editors and integrated development environments such as Vim and Emacs.

General Overview of a Simple C Program's Structure:

- **Header Files:**
- The ***#include directives*** at the beginning of the program are used to include ***header files. Header files*** provide function ***prototypes*** and ***definitions*** that allow the C compiler to understand the functions used in the program.
- **Main Function:**
- Every ***C program*** starts with the ***main function***. It is the program's entry point, and execution starts from here. The ***main function*** has a ***return type*** of ***int***, indicating that it should return an integer value to the operating system upon completion.
- **Variable Declarations:**
- Before using any variables, you should declare them with their ***data types***. This section is typically placed after the ***main function's*** curly opening brace.



```c
#include <stdio.h>
#define PI 3.14159
int main() {
    float radius = 5.0;
    float area = PI * radius * radius;

    printf("Area of the circle: %f\n", area);
    return 0;
}
```

Output

```
Area of the circle: 78.539749
```

General Overview of a Simple C Program's Structure:

- **Statements and Expressions:**
- This section contains the ***actual instructions*** and ***logic*** of the program. C programs are composed of statements that perform ***actions*** and ***expressions*** that compute values.
- **Comments:**
- ***Comments*** are used to provide ***human-readable*** explanations within the code. They are not executed and do not affect the program's functionality. In C, comments are denoted by ***//*** for ***single-line comments*** and ***/* */*** for ***multi-line comments***.
- **Return Statement:**
- Use the ***return statement*** to terminate a function and return a value to the caller function. A ***return statement*** with a value of ***0*** typically indicates a successful execution in the ***main function***, whereas a ***non-zero value*** indicates an error or unexpected termination.

```
// C program to demonstrate syntax of
arithmetic operators
#include <stdio.h>

int main()
{
	int a = 10, b = 4, res;
	res = a + b; // addition
	printf("a + b is %d\n", res);

	return 0;
}
```

General Overview of a Simple C Program's Structure:

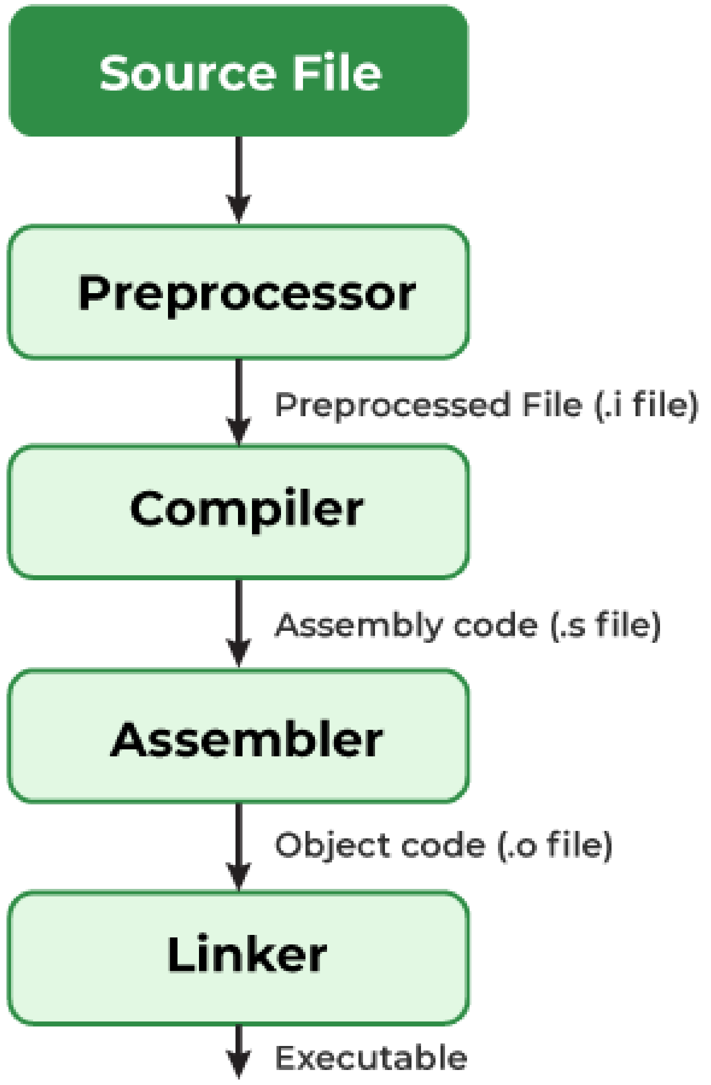
- **Functions:**
- C programs can include ***user-defined*** functions and ***blocks*** of code that perform specific tasks. Functions help modularize the code and make it more organized and manageable.
- **Standard Input/Output:**
- C has ***library functions*** for reading user ***input (scanf)*** and printing output to the console ***(printf)***. These functions are found in C programs and are part of the standard I/O library (***stdio.h*** header file). It is essential to include these fundamental features correctly while writing a simple C program to ensure optimal functionality and readability.

```
// C Program to illustrate the use of user-defined function
#include <stdio.h>
// Function prototype
int sum(int, int);

// Function definition
int sum(int x, int y){
        int sum;
        sum = x + y;
        return x + y;
}
int main(){
        int x = 10, y = 11;

        // Function call
        int result = sum(x, y);
        printf("Sum of %d and %d = %d ", x, y, result);
        return 0;
}
```

Compilation process in C



- **What is a compilation?**
- The compilation is a process of converting the source code into object code. It is done with the help of the compiler. The compiler checks the source code for the syntactical or structural errors, and if the source code is error-free, then it generates the object code.
- The c compilation process converts the source code taken as input into the object code or machine code. The compilation process can be divided into four steps, i.e., Pre-processing, Compiling, Assembling, and Linking.
- The preprocessor takes the source code as an input, and it removes all the comments from the source code. The preprocessor takes the preprocessor directive and interprets it. For example, if **<stdio.h>,** the directive is available in the program, then the preprocessor interprets the directive and replace this directive with the content of the **'stdio.h'** file.
- The following are the phases through which our program passes before being transformed into an executable form:

Compilation process in C

- **Preprocessor**
- The source code is the code which is written in a text editor and the source code file is given an extension ".c". This source code is first passed to the preprocessor, and then the preprocessor expands this code. After expanding the code, the expanded code is passed to the compiler.
- **Compiler**
- The code which is expanded by the preprocessor is passed to the compiler. The compiler converts this code into assembly code. Or we can say that the C compiler converts the pre-processed code into assembly code.
- **Assembler**
- The assembly code is converted into object code by using an assembler. The name of the object file generated by the assembler is the same as the source file. The extension of the object file in DOS is '.obj,' and in UNIX, the extension is 'o'. If the name of the source file is **'hello.c',** then the name of the object file would be 'hello.obj'.
- **Linker**
- Mainly, all the programs written in C use library functions. These library functions are pre-compiled, and the object code of these library files is stored with '.lib' (or '.a') extension. The main working of the linker is to combine the object code of library files with the object code of our program.

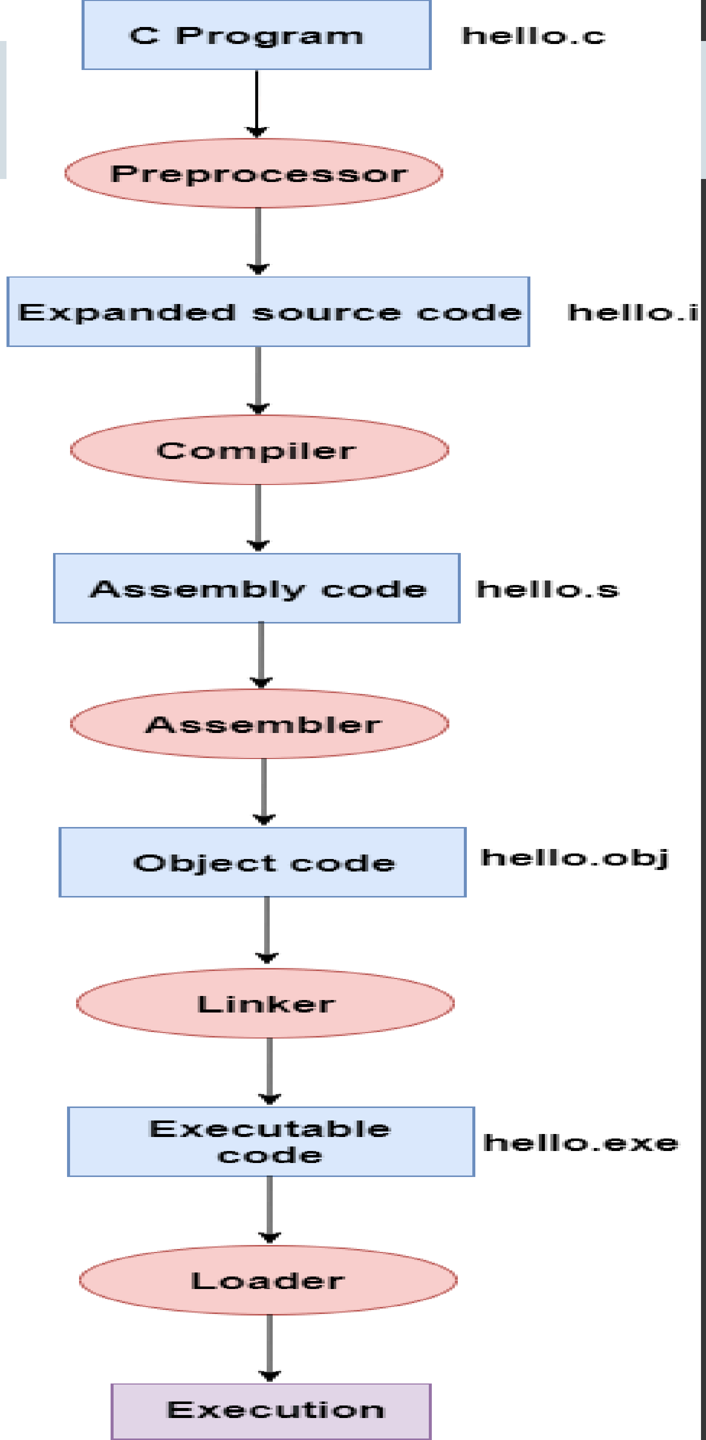
Compilation process in C

Let's understand through an example.

```
#include <stdio.h>
int main()
{
   printf("Hello javaTpoint");
   return 0;
}
```

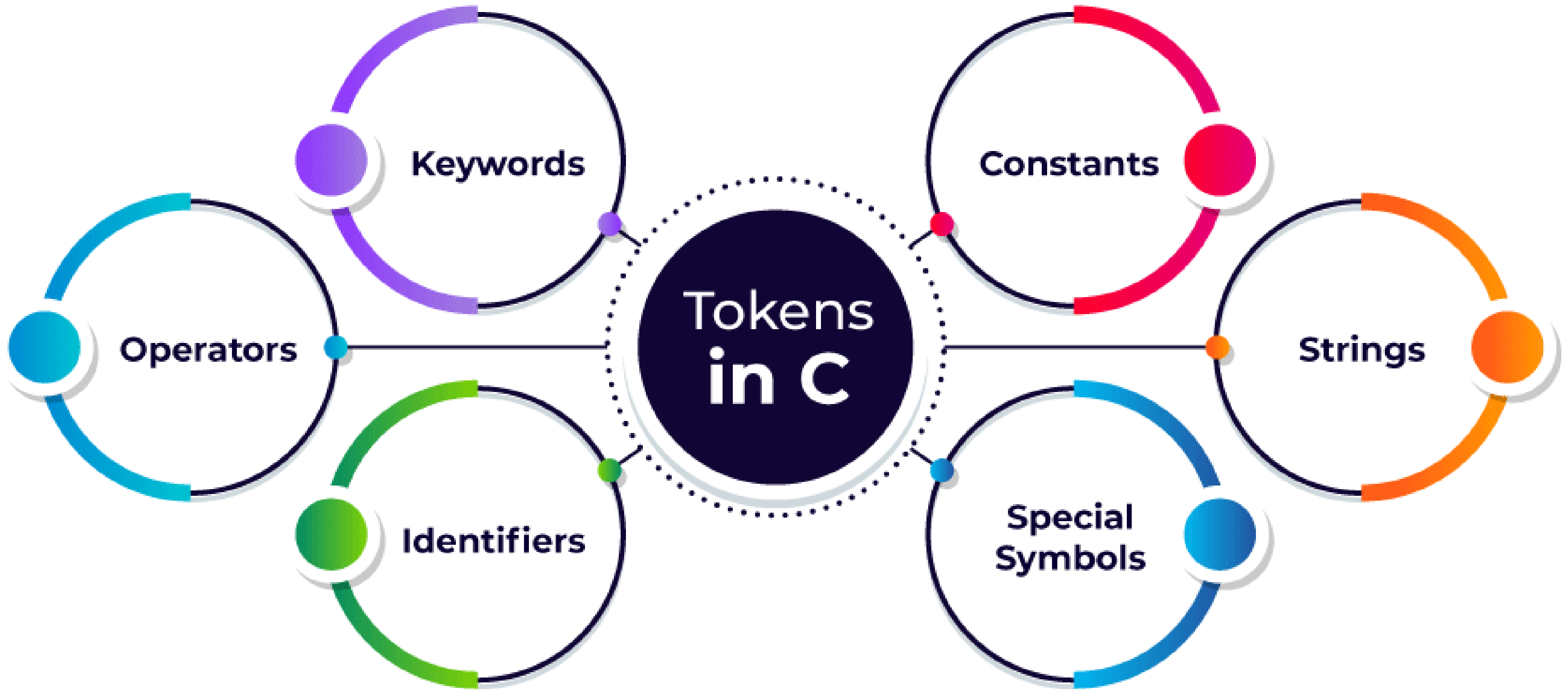
In the above flow diagram, the following steps are taken to execute a program:

- Firstly, the input file, i.e., **hello.c,** is passed to the preprocessor, and the preprocessor converts the source code into expanded source code. The extension of the expanded source code would be **hello.i.**
- The expanded source code is passed to the compiler, and the compiler converts this expanded source code into assembly code. The extension of the assembly code would be **hello.s.**
- This assembly code is then sent to the assembler, which converts the assembly code into object code.
- After the creation of an object code, the linker creates the executable file. The loader will then load the executable file for the execution.



Tokens in C

- A token in C can be defined as the smallest individual element of the C programming language that is meaningful to the compiler. It is the basic component of a C program.
- A token in C is the **meaningful smallest unit** used in a C program.
- **Types of Tokens in C:** The tokens of C language can be classified into six types based on the functions they are used to perform.



Keywords in C

- In C Programming language, there are many rules so to avoid different types of errors. One of such rule is not able to declare variable names with auto, long, etc. This is all because these are keywords. Let us check all keywords in C language.
- **What are Keywords?**
- Keywords are predefined or reserved words that have special meanings to the compiler. These are part of the syntax and cannot be used as identifiers in the program. A list of keywords in C or reserved words in the C programming language are mentioned 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

C Identifiers

Identifiers are unique names that are assigned to variables, structs, functions, and other entities.

Rules to Name an Identifier in C

1. An identifier can include letters (a-z or A-Z), and digits (0-9).
2. An identifier cannot include special characters(/,<,>,*,!,~,#,@,$,%,^,&,*) except the '_' underscore.
3. Spaces are not allowed while naming an identifier.
4. An identifier can only begin with an underscore or letters.
5. We cannot name identifiers the same as keywords because they are reserved words to perform a specific task. For example, printf, scanf, int, char, struct, etc.
6. The identifier must be **unique** in its namespace.
7. C language is case-sensitive so, 'name' and 'NAME' are different identifiers.

✓ Number, num=2, val, VAL=23, number1,number2, a1,a2,

✗ 1num, #val, ^list, $name, ~age, &age

✓ _num1, num1_, customer_list, customer_list1
✗ 1num_, customer-list, 1_num,

✓ customer_list
✗ customer list

✓ INT, PRINTF, CHAR
✗ int = 0, printf=0,

✓ name=0, NAME=0, Name=0, NaMe=0, NAMe=0,namE=0

C Comments

- The **comments in C** are human-readable explanations or notes in the source code of a C program. A comment makes the program easier to read and understand. These are the statements that are not executed by the compiler or an interpreter.
- It is considered to be a good practice to document our code using comments.
- **When and Why to use Comments in C programming?**

1. A person reading a large code will be bemused if comments are not provided about details of the program.
2. C Comments are a way to make a code more readable by providing more descriptions.
3. C Comments can include a description of an algorithm to make code understandable.
4. C Comments can be used to prevent the execution of some parts of the code.

- **Types of comments in C**
- In C there are two types of comments in C language:
- **Single-line comment**
- **Multi-line comment**

C Comments

Single-line Comment in C

A single-line comment in C starts with (**//**) double forward slash. It extends till the end of the line and we don't need to specify its end.

Syntax of Single Line C Comment

```
// This is a single line comment
```

2. Multi-line Comment in C

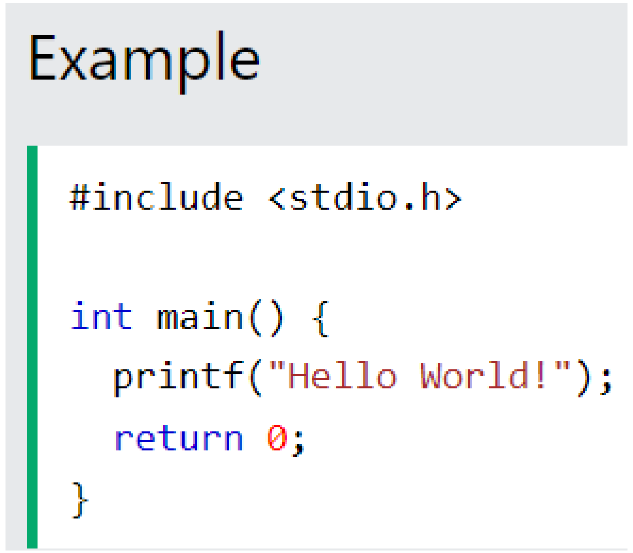
The Multi-line comment in C starts with a forward slash and asterisk (**/***) and ends with an asterisk and forward slash (***/**). Any text between **/*** and ***/** is treated as a comment and is ignored by the compiler. It can apply comments to multiple lines in the program.

Syntax of Multi-Line C Comment

```
/*Comment starts
continues
continues
.
.
.
Comment ends*/
```

C Output (Print Text)

Output (Print Text)
To output values or print text in C, you can use the printf() function:



Example

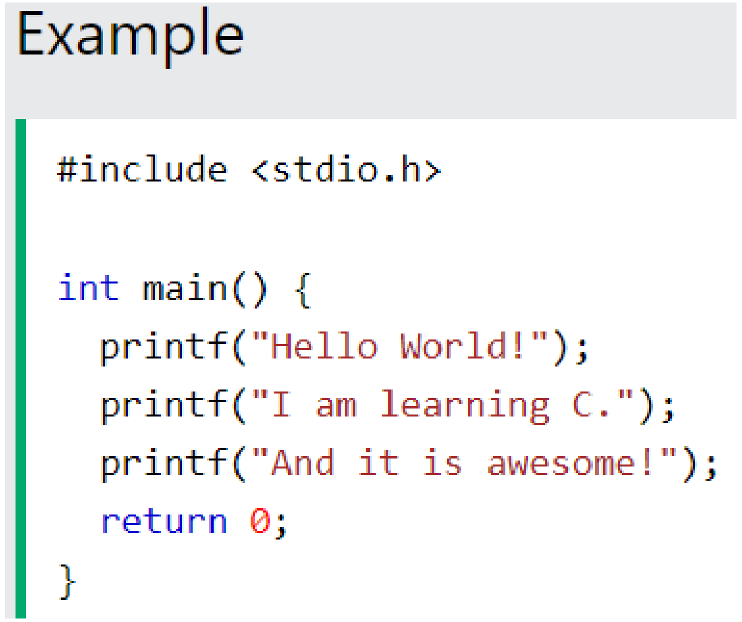
```c
#include <stdio.h>

int main() {
  printf("Hello World!");
  return 0;
}
```

Double Quotes
When you are working with text, it must be wrapped inside double quotations marks "".

If you forget the double quotes, an error occurs.

You can use as many printf() functions as you want. However, note that it does not insert a new line at the end of the output:



Example

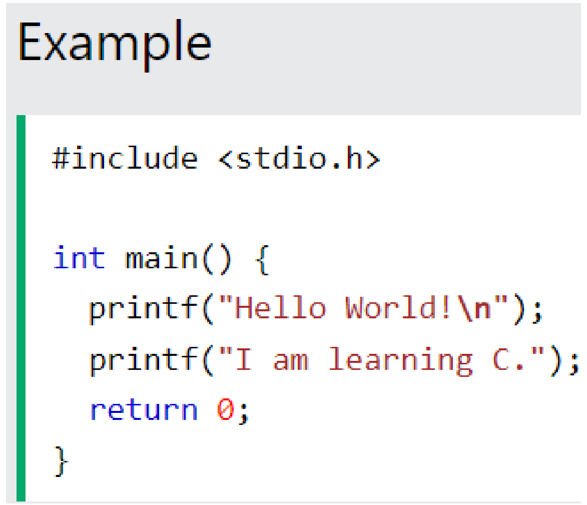
```c
#include <stdio.h>

int main() {
  printf("Hello World!");
  printf("I am learning C.");
  printf("And it is awesome!");
  return 0;
}
```

Output:
Hello World!I am learning C.And it is awesome!

C New Lines

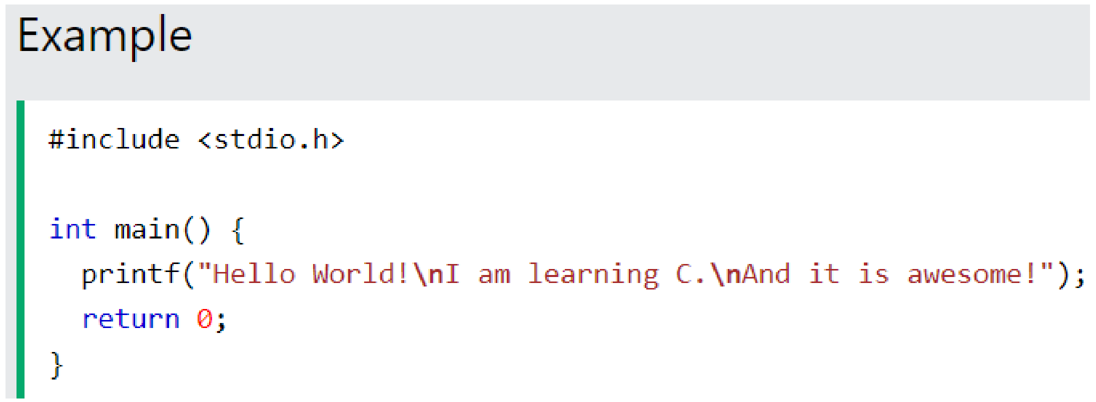
New Lines
To insert a new line, you can use the \n character:


Example

```
#include <stdio.h>

int main() {
  printf("Hello World!\n");
  printf("I am learning C.");
  return 0;
}
```

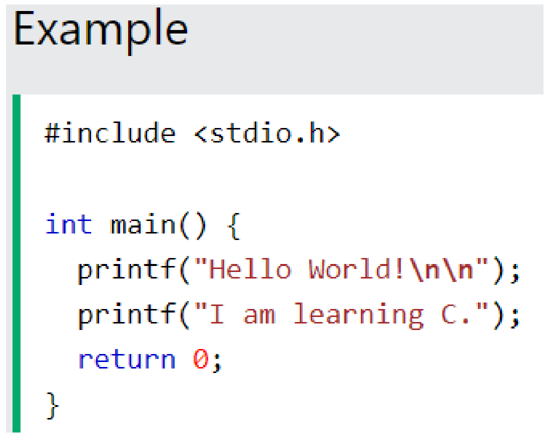
You can also output multiple lines with a single printf() function. However, this could make the code harder to read:


Example

```
#include <stdio.h>

int main() {
  printf("Hello World!\nI am learning C.\nAnd it is awesome!");
  return 0;
}
```

Tip: Two \n characters after each other will create a blank line:


Example

```
#include <stdio.h>

int main() {
  printf("Hello World!\n\n");
  printf("I am learning C.");
  return 0;
}
```

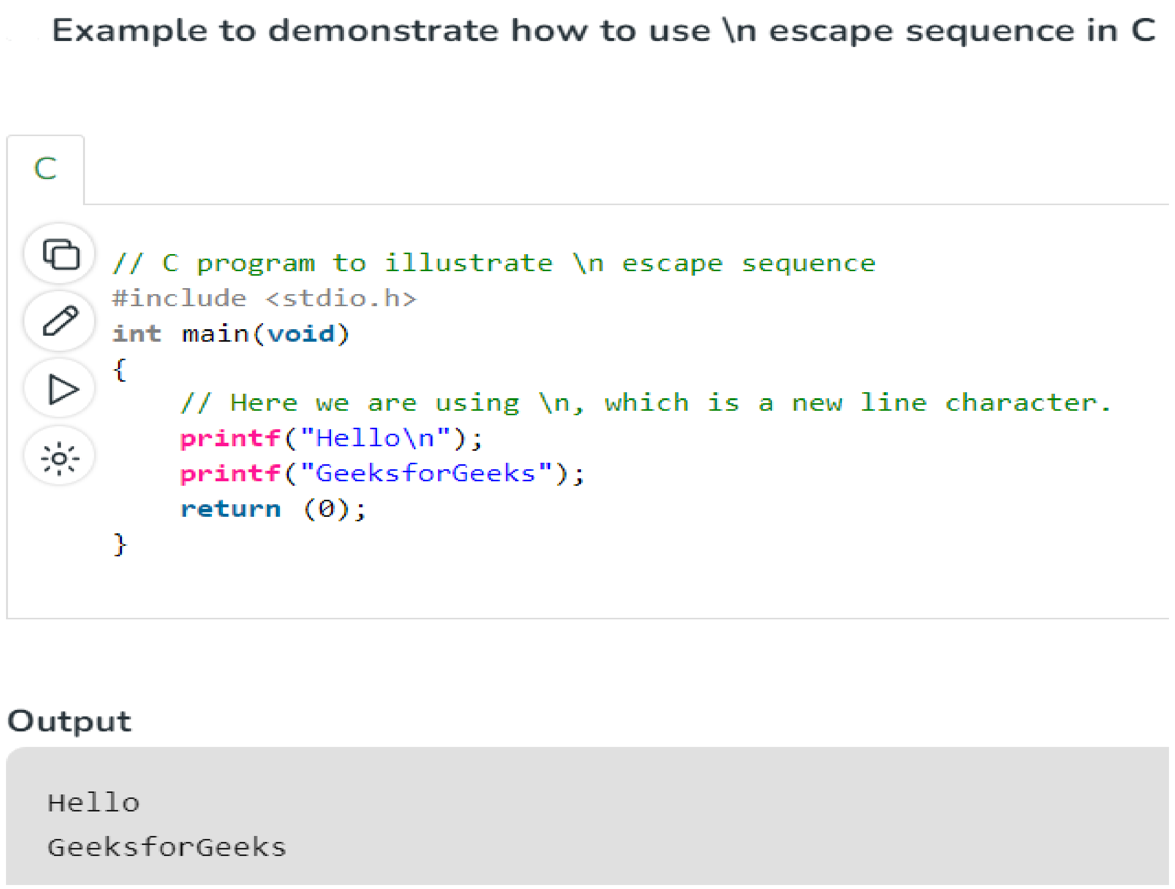
Escape Sequence in C

- The escape sequence in C is the characters or the sequence of characters that can be used inside the string literal. The purpose of the escape sequence is to represent the characters that cannot be used normally using the keyboard. Some escape sequence characters are the part of ASCII charset but some are not.

Escape Sequence	Name	Description
\n	New Line	It moves the cursor to the start of the next line.
\t	Horizontal Tab	It inserts some whitespace to the left of the cursor and moves the cursor accordingly.
\\	Backslash	Use to insert backslash character.
\’	Single Quote	It is used to display a single quotation mark.
\”	Double Quote	It is used to display double quotation marks.
\?	Question Mark	It is used to display a question mark.

An escape sequence contains a backslash (\) symbol followed by one of the escape sequence characters or an octal or hexadecimal number.

Escape Sequence in C



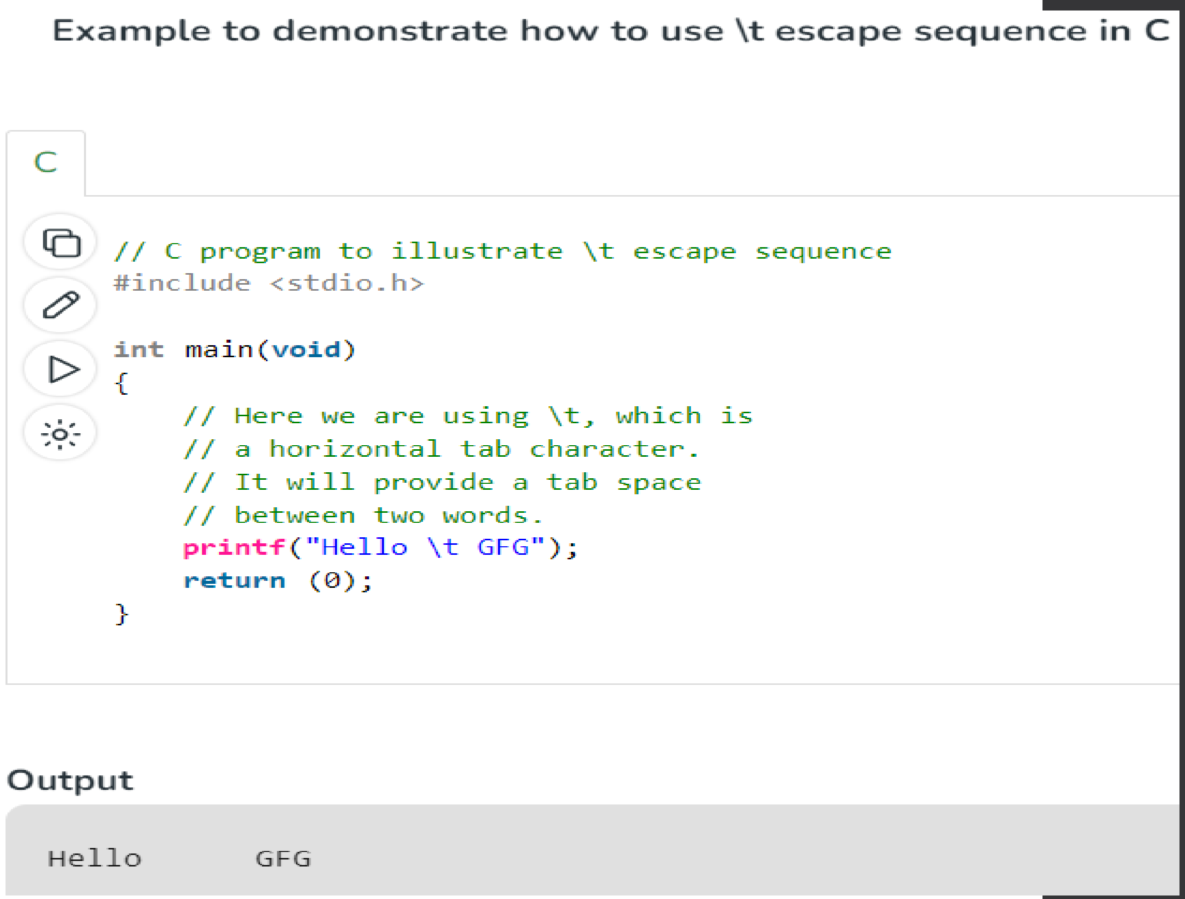
Example to demonstrate how to use \n escape sequence in C

C

```
// C program to illustrate \n escape sequence
#include <stdio.h>
int main(void)
{
    // Here we are using \n, which is a new line character.
    printf("Hello\n");
    printf("GeeksforGeeks");
    return (0);
}
```

Output

```
Hello
GeeksforGeeks
```



Example to demonstrate how to use \t escape sequence in C

C

```
// C program to illustrate \t escape sequence
#include <stdio.h>

int main(void)
{
    // Here we are using \t, which is
    // a horizontal tab character.
    // It will provide a tab space
    // between two words.
    printf("Hello \t GFG");
    return (0);
}
```

Output

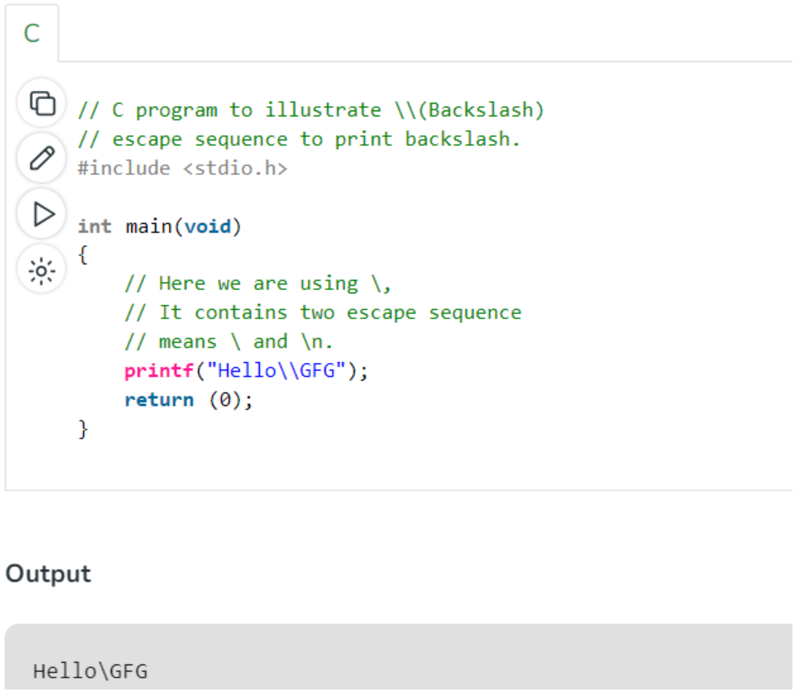
```
Hello      GFG
```

What is \n exactly?

The newline character (\n) is called an escape sequence, and it forces the cursor to change its position to the beginning of the next line on the screen. This results in a new line.

Escape Sequence in C

Example to demonstrate how to use \\ escape sequence in C



C

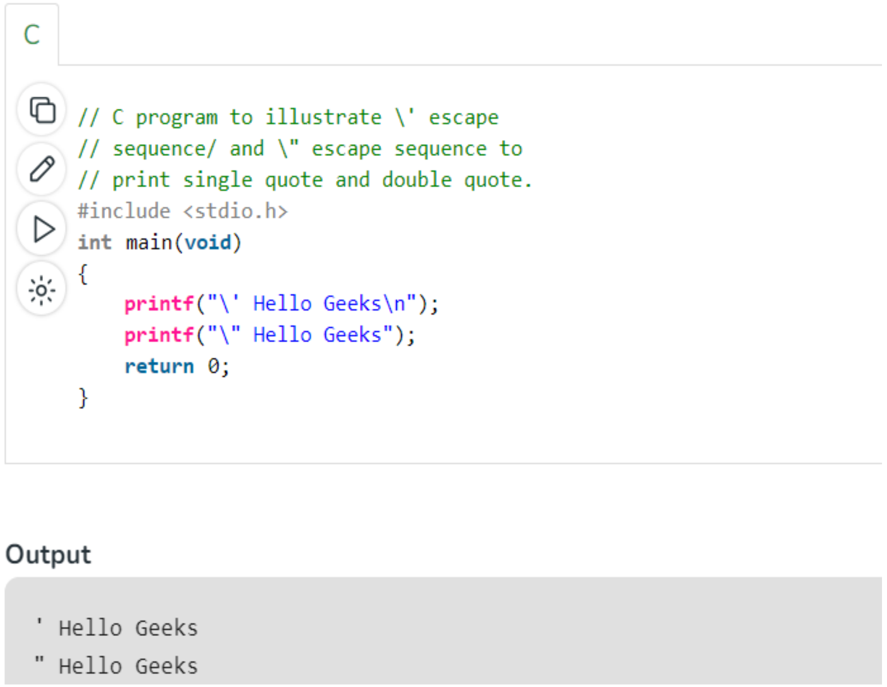
```c
// C program to illustrate \\(Backslash)
// escape sequence to print backslash.
#include <stdio.h>

int main(void)
{
    // Here we are using \,
    // It contains two escape sequence
    // means \ and \n.
    printf("Hello\\GFG");
    return (0);
}
```

Output

```
Hello\GFG
```

Example to demonstrate how to use \' and \" escape sequence in C



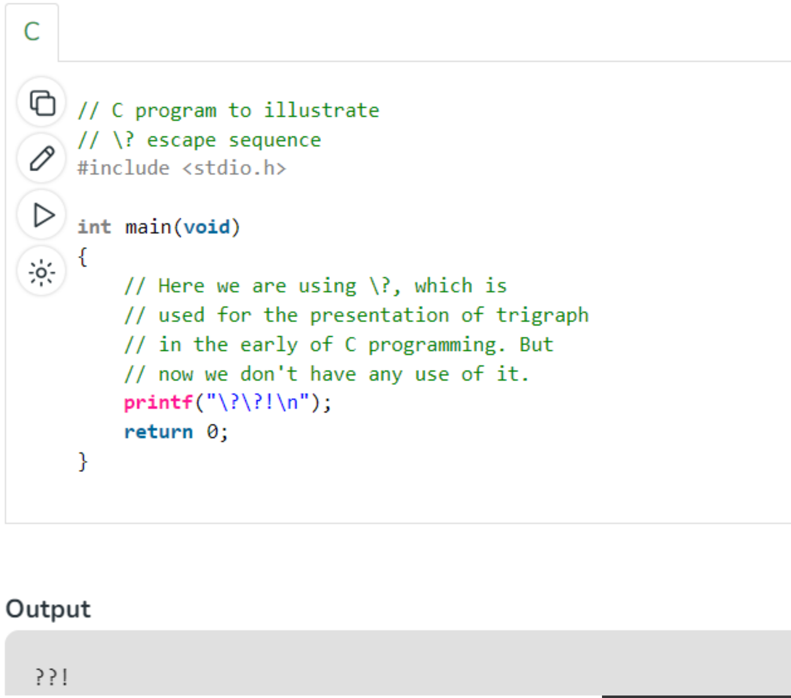
C

```c
// C program to illustrate \' escape
// sequence/ and \" escape sequence to
// print single quote and double quote.
#include <stdio.h>
int main(void)
{
    printf("\' Hello Geeks\n");
    printf("\" Hello Geeks");
    return 0;
}
```

Output

```
' Hello Geeks
" Hello Geeks
```

Example to demonstrate how to use \? escape sequence in C



C

```c
// C program to illustrate
// \? escape sequence
#include <stdio.h>

int main(void)
{
    // Here we are using \?, which is
    // used for the presentation of trigraph
    // in the early of C programming. But
    // now we don't have any use of it.
    printf("\?\?!\n");
    return 0;
}
```

Output

```
??!
```

Video Resources To Follow(click on the link):-

- C programming Bangla Tutorial 5.12 : First C program
- C programming Bangla Tutorial 5.13 : Comments and Escape sequence
- C programming Bangla Tutorial 5.15 : Keyword, Variable, data type (part-1)
- C programming Bangla Tutorial 5.17 : More on data types