



## Overview of C

- C is developed by Dennis Ritchie
- C is a structured programming language
- C supports functions that enables easy maintainability of code, by breaking large file into smaller modules
- Comments in C provides easy readability
- C is a powerful language

## Program structure

#include inserts another file. “.h” files are called “header” files. They contain stuff needed to interface to libraries and code in other “.c” files.

This is a comment. The compiler ignores this.

```
#include <stdio.h>
/* The simplest C Program */
void main()
{
    printf("Hello world\n");
}
```

The main() function is always where your program starts running.

Blocks of code (“lexical scopes”) are marked by { ... }

Print out a message. ‘\n’ means “new line”.



## Header files

- The files that are specified in the include section is called as header file
- These are precompiled files that has some functions defined in them
- We can call those functions in our program by supplying parameters
- Header file is given an extension .h
- C Source file is given an extension .c



## Main function

- This is the entry point of a program
- When a file is executed, the start point is the main function
- From main function the flow goes as per the programmers choice.
- There may or may not be other functions written by user in a program
- Main function is compulsory for any c program



## Running a C Program

- Type a program
- Save it
- Compile the program – This will generate an exe file (executable)
- Run the program (Actually the exe created out of compilation will run and not the .c file)
- In different compiler we have different option for compiling and running. We give only the concepts.

- Single line comment
  - `//` (double slash)
  - Termination of comment is by pressing enter key

- Multi line comment

```
/*....
```

```
.....*/
```

This can span over to multiple lines

- Primitive/Primary data types
  - int, float, char
- Secondary data types
  - Arrays come under this category
  - Arrays can contain collection of int or float or char or double data
- User defined data types
  - Structures and enum fall under this category.

- Variables are data that will keep on changing
- Declaration  
`<<Data type>> <<variable name>>;`  
`int a;`
- Definition  
`<<varname>>=<<value>>;`  
`a=10;`
- Usage  
`<<varname>>`  
`a=a+1;   //increments the value of a by 1`





## Variable names- Rules

- Should not be a reserved word like int etc...
- Should start with a letter or an underscore(\_)
- Can contain letters, numbers or underscore.
- No other special characters are allowed including space
- Variable names are case sensitive
  - A and a are different.



# Input and Output

- Input
  - `scanf("%d",&a);`
  - Gets an integer value from the user and stores it under the name "a"
- Output
  - `printf("%d",a);`
  - Prints the value present in variable a on the screen

- In most software, the statements in the program may need to repeat for many times.
  - e.g., calculate the value of  $n!$ .
  - If  $n = 10000$ , it's not elegant to write the code as  $1*2*3*...*10000$ .
- **Loop** is a control structure that repeats a group of steps in a program.
  - **Loop body** stands for the repeated statements.
- There are three C loop control statements:
  - **while**, **for**, and **do-while**.

- The syntax of `while` statement in C:

```
while (loop repetition condition)
{
    statement 1;
    statement 2;
}
```

- Loop repetition condition** is the condition which controls the loop.
- The **statement** is repeated as long as the loop repetition condition is **true**.
- A loop is called an **infinite loop** if the loop repetition condition is always true.

Eg:

```
a=10;
while(a != 0)
{ printf("%d",a);
  a--;
}
```

Output: 10987654321



## The `for` Statement in C

- The syntax of `for` statement in C:

```
for (initialization expression; loop repetition condition; inc/dec)
{
    statement 1;
    statement 2;
}
```

- The **initialization expression** set the initial value of the loop control variable.
- The **loop repetition condition** test the value of the loop control variable.
- The **update expression** update the loop control variable.
- Eg: Program to print Hello 10 times

```
for(i=0;i<10;i++)
{
    printf("Hello");
}
```



# Increment and Decrement Operators

- The statements of increment and decrement are commonly used in the loop.
- The increment (i.e., `++`) or decrement (i.e., `--`) operators are the frequently used operators which take only one operand.
- The increment/decrement operators increase or decrease the value of the single operand.
  - e.g., for (int i = 0; i < 100; `i++`){ ... }
  - The variable i increase one after each iteration.

- The syntax of do-while statement in C:

```
do
{
    statement 1;
    statement 2;
}
while (loop repetition condition);
```

- The *statement* is first executed.
- If the **loop repetition condition** is true, the *statement* is repeated.
- Otherwise, the loop is exited.

- Eg: i=10;

```
do
{
    printf("%d",i);
    i--;
}while(i!=0);
```

Output:

10987654321

- What are the different loops in C? Differentiate between While and Do-while loop.

StudentSuvidha.com