

# **Problem Solving With C**

**UE15CS151**

**By:**

**Amit V Pujari**

Asst. Professor

Department of Computer Science And  
Engineering

PES University

Mail : amith.v.pujari@gmail.com

## **Topics Covered:**

- **Data Types**
- **Format Specifiers**
- **Structure of a C Program**
- **Phases of C program execution**
- **Break Statement**
- **Continue statement**
- **Exit Statement**

**This Notes can be used only for reference and kindly do not solely depend on it.  
Only those topics which need more explanation are included here. Please Note  
“The prescribed Text book has to be referred for the examination”**

## Data Types:

A data type is a classification identifying one of the various types of data such as real, integer etc, that determines the possible values for that type. Data type also specifies the operations that can be done on values of that type, the meaning of the data and the ways values of that type can be stored.

Main Divisions under the Data types are:

- **Numeric**

- Integer - int

- Qualifiers – signed , unsigned, long and short

- Real Numbers – float, double

- Qualifier – long

- **Non Numeric Types : Char**

- Sometimes Char is intern considered to be of an integer type as every character has an equivalent ASCII value which is of integer type .

- **Structured Types – Arrays, Structures etc.**

- **Pointer Types**

- **File Types**

## Format Specifiers in c:

- **Integer Types**

- %d - represents Integer values
- %o – represents Octal Values
- %x / %X - Hexadecimal values
- %i - decimal ,octal and hexadecimal values
- %u – Unsigned Integer values
- %h – Short integer values

- **Real Types**

- %f /%g - Floating point values
- %lf - double values
- %e / %E– Exponential representation

- **Character Type**

- %c – Character Type
- %s - String [ array of characters]

## Structure of a C Program :

Documentation Section
Pre-processor Directives
Global Declaration
<pre>main() {     Declaration section;     Executable Section; }</pre>
User Defined Functions

- **Documentation Section :**

This section consists of the Comment statements that brief about the Program

Example:

// this program takes 2 numbers and displays the sum of the numbers.

- Single line comment : // "String"
- Multi line comment : /\* "String 1"  
"String 2"  
"String 3" \*/

- **Pre-processor Directives :**

- Pre-processing is the first step which takes place before the compilation. when you compile a program we are actually calling the pre-processor i.e. the compile will now invoke the pre-processed program.
- It takes the source code that you have generated and produce a program. Now the program produced by the pre-processor will not have this "#" symbol. It actually produces the pre-processed code.
- Pre-processor copies the processed code of stdio.h header file contains declaration of the functions.

- **Global Declarations:**

This segment is outside main() and here we declare global variable and function prototypes .

- **main function :**

Execution a C Program by Default starts from the main function. It acts as a default starting point to a function execution. But this default execution can be changed.

- **Pair of Curly Braces :**

These Brackets define the scope. Scope here means the boundary for the usage of particular variables and executable statements.

All functions must have these curly braces. A function must start and end with curly brackets

- **Declaration Section:**

The variables that are used within the main should be declared at the beginning of the main function. But the Higher GCC versions allows variable declarations anywhere in the program.

- **Executable Section:**

This section includes the instructions or statements that are executable like control statements, i/o statements etc. All these statements must end with Semicolon[";"].

- **User Defined Functions:**

C language gives privilege to the user to create their own functions and execute them. Such functions are termed as user defined functions and this section is completely reserved for such functions.

## **Phases of Program Execution:**

- Editing Phase
- Pre compilation
- Compilation
- Assembling
- Linking
- Loading

## break , continue and exit statements:

### example 1 :

```
int i,a[3] = {1,2,3};
for(i=0;i<3;i++)
    printf("%d \t",&a[i]);
printf("The end");
```

**output:** 1      2      3      The end

### example 2 : With Break

```
int i,a[3] = {1,2,3};
for(i=0;i<3;i++)
{
    if(i==1)
    {
        break;
    }
    printf("%d \t",&a[i]);
}
printf("The end");
```

**output:** 1      The end      //The control totally comes out of the looping statement(for) when the "break" statement is executed.

### example 3 : With Continue

```
int i,a[3] = {1,2,3};
for(i=0;i<3;i++)
{
    if(i==1)
    {
        continue;
    }
    printf("%d \t",&a[i]);
}
printf("The end");
```

**output:** 1      3      The end      //The control skips the next staemnts of the loof and gets back to the updaton of the loop.

**example 4 :** With Exit

```
int i,a[3] = {1,2,3};
for(i=0;i<3;i++)
{
    if(i==2)
    {
        exit(0);
    }
    printf("%d \t",&a[i]);
}
printf("The end");
```

**output:** 1      //The whole main function terminates when the exit statement is Encountered