As per Revised Syllabus of University of Mumbai,
Choice Based Credit & Grading System 2019

MU

First Year Degree Course In Engineering
COMMON FOR ALL BRANCHES (Sem - II)

'C' PROGRAMMING



SURESHAN NAIR

www.pragationline.com
www.facebook.com/niralibooks
@nirali.prakashan

NIRALI PRAKASHAN
ADVANCEMENT OF KNOWLEDGE

A TEXT BOOK OF

C PROGRAMMING

FOR

SEMESTER – II

F.E. DEGREE COURSES IN ENGINEERING
(COMMON FOR ALL BRANCHES)
(CODE – FEC205)

As Per Revised Syllabus of UNIVERSITY OF MUMBAI, MUMBAI
Choice Based Credit and Grading System w.e.f. Academic Year 2019-2020

Sureshan Nair

Master in Computers from National Institute of Technology. Calicut, with more than 25 years of teaching experience in teaching Engineering Degree students of various engineering colleges affiliated to Mumbai University. Worked several years for all top engineering classes in Mumbai as their leading teacher for computer subjects.

NIRALI PRAKASHAN
ADVANCEMENT OF KNOWLEDGE

N6010

C PROGRAMMING **ISBN 978-93-89825-59-6**

First Edition : January 2020

© : Author

The text of this publication, or any part thereof, should not be reproduced or transmitted in any form or stored in any computer storage system or device for distribution including photocopy, recording, taping or information retrieval system or reproduced on any disc, tape, perforated media or other information storage device etc., without the written permission of Authors with whom the rights are reserved. Breach of this condition is liable for legal action.

Every effort has been made to avoid errors or omissions in this publication. In spite of this, errors may have crept in. Any mistake, error or discrepancy so noted and shall be brought to our notice shall be taken care of in the next edition. It is notified that neither the publisher nor the authors or seller shall be responsible for any damage or loss of action to any one, of any kind, in any manner, therefrom.

Published By :
NIRALI PRAKASHAN
Abhyudaya Pragati, 1312, Shivaji Nagar,
Off J.M. Road, PUNE – 411005
Tel - (020) 25512336/37/39, Fax - (020) 25511379
Email : niralipune@pragationline.com

POLYPLATE

Printed By :
YOGIRAJ PRINTERS AND BINDERS
Works: Sr. No. 10\1,Ghule Industrial Estate,
Tal-Haveli, Dist-Pune 411041.
Tel - (020) 24690371
Mobile - 9850046517, 9404233041

➢ DISTRIBUTION CENTRES

PUNE

Nirali Prakashan : 119, Budhwar Peth, Jogeshwari Mandir Lane, Pune 411002, Maharashtra
(For orders within Pune) Tel : (020) 2445 2044, Mobile : 9657703145
Email : niralilocal@pragationline.com

Nirali Prakashan : S. No. 28/27, Dhayari, Near Asian College Pune 411041
(For orders outside Pune) Tel : (020) 24690204 Fax : (020) 24690316; Mobile : 9657703143
Email : bookorder@pragationline.com

MUMBAI

Nirali Prakashan : 385, S.V.P. Road, Rasdhara Co-op. Hsg. Society Ltd.,
Girgaum, Mumbai 400004, Maharashtra; Mobile : 9320129587
Tel : (022) 2385 6339 / 2386 9976, Fax : (022) 2386 9976
Email : niralimumbai@pragationline.com

➢ DISTRIBUTION BRANCHES

JALGAON

Nirali Prakashan : 34, V. V. Golani Market, Navi Peth, Jalgaon 425001, Maharashtra,
Tel : (0257) 222 0395, Mob : 94234 91860; Email : niralijalgaon@pragationline.com

KOLHAPUR

Nirali Prakashan : New Mahadvar Road, Kedar Plaza, 1st Floor Opp. IDBI Bank, Kolhapur 416 012
Maharashtra. Mob : 9850046155; Email : niralikolhapur@pragationline.com

NAGPUR

Nirali Prakashan : Above Maratha Mandir, Shop No. 3, First Floor,
Rani Jhanshi Square, Sitabuldi, Nagpur 440012, Maharashtra
Tel : (0712) 254 7129; Email : niralinagpur@pragationline.com

DELHI

Nirali Prakashan : 4593/15, Basement, Agarwal Lane, Ansari Road, Daryaganj
Near Times of India Building, New Delhi 110002 Mob : 08505972553
Email : niralidelhi@pragationline.com

BENGALURU

Nirali Prakashan : Maitri Ground Floor, Jaya Apartments, No. 99, 6th Cross, 6th Main,
Malleswaram, Bengaluru 560003, Karnataka; Mob : 9449043034
Email: niralibangalore@pragationline.com

Other Branches : Hyderabad, Chennai

niralipune@pragationline.com | www.pragationline.com

Also find us on www.facebook.com/niralibooks

Note : Every possible effort has been made to avoid errors or omissions in this book. In spite this, errors may have crept in. Any type of error or mistake so noted, and shall be brought to our notice, shall be taken care of in the next edition. It is notified that neither the publisher, nor the author or book seller shall be responsible for any damage or loss of action to any one of any kind, in any manner, therefrom. The reader must cross check all the facts and contents with original Government notification or publications.

PREFACE

As a teacher teaching engineering students of Mumbai University for more than twenty five years, I am really happy to present the first edition of C programming to the students of first year engineering of Mumbai University. This book is prepared according to the new syllabus (2019) of University of Mumbai; which comes into effect from the academic year 2019-2020.

The syllabus has been meticulously followed throughout the book with emphasis on minute details. The topics have been structured in order to develop the subject in a logical sequence. The coverage of this book will definitely help students to understand and grasp the subject with ease.

For each section numerous programs are included using the simple to complex approach. Almost all past exam questions are solved which will definitely help a student to understand programming properly and score well. I would like to request students to keep this text till they complete their engineering course as this basic book may help them to revise programming concepts for their campus selection process.

At the end, there are some questions and answers given which are frequently asked in interviews based on C language.

Any constructive suggestions for enhancing the utility of this book and for removing errors that might have gone unnoticed will be gratefully acknowledged.

Author

Sureshan Nair

ACKNOWLEDGEMENT

I would like to express my deepest gratitude to all who have encouraged me to come out with a text book on C programming.

I would like to thank all my Students, Mr. Rajendra Bade, my parents, my wife Sujatha and daughters Sanju and Sandra for their support and encouragement. I would like to thank all at Nirali Prakashan especially Mr. Shashikant Patel for giving me this opportunity. I am also thankful to Mr. Paras Rambhia and Ms. Riddhi Vira for their excellent typesetting work.

DEDICATED

TO

ALL MY STUDENTS

SYLLABUS

Module 1 — 05 Hours

Introduction

- Introduction to components of a Computer System
- Introduction to Algorithm and Flowchart

Fundamentals of C Programming

- Keywords, Identifiers, Constants and Variables
- Data types in C
- Operators in C
- Basic Input and Output Operations
- Expressions and Precedence of Operators
- In-built Functions

Module 2 — 07 Hours

Control Structures

- Introduction to Control Structures

Branching and Looping Structures

- If statement, If-else statement, Nested if-else, else-if Ladder
- Switch statement
- For loop, While loop, Do while loop
- Break and Continue

Module 3 — 04 Hours

Functions

- Introduction to functions
- Function prototype, Function definition, Accessing a function and parameter passing
- Recursion

Module 4 — 04 Hours

Arrays and Strings

- Introduction to Arrays
- Declaration and initialization of one dimensional and two-dimensional arrays
- Definition and Initialization of String
- String functions

Module 5 — 04 Hours

Structure and Union

- Concept of Structure and Union
- Declaration and Initialization of structure and union
- Nested Structures
- Array of Structures
- Passing structure to functions

Module 6 — 04 Hours

Pointers

- Fundamentals of pointers
- Declaration, initialization and dereferencing of pointers
- Operations on Pointers
- Concept of dynamic memory allocation

TEACHING AND EXAMINATION SCHEME

Course code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Tutorial	Practical	Total
FEC205	C Programming	2	-	-	2	-	-	2

Course code	Course Name	Examination Scheme							
		Theory					Term work	Practical/ Oral	Total
		Internal Assessment			End Sem. Exam.	Exam. Duration (in Hrs.)			
		Test 1	Test 2	Average					
FEC205	C Programming	15	15	15	60	2	-	-	75

ASSESSMENT

Internal Assessment Test

Assessment consists of two class tests of 15 marks each. The first class test is to be conducted when approx. 40% syllabus is completed and second class test when additional 35% syllabus is completed. Duration of each test shall be one hour.

End Semester Theory Examination

1. Question paper will comprise of total 06 questions, each carrying 15 marks.
2. Total 04 questions need to be solved.
3. Question No : 01 will be compulsory and based on entire syllabus wherein sub-questions of 2 to 5 marks will be asked.
4. Remaining questions will be mixed in nature. (e.g. Suppose Q.2 has part (a) from module 3 then part (b) will be from any module other than module 3)
5. In question paper weightage of each module will be proportional to number of respective lecture hrs as mentioned in the syllabus.

CONTENTS

Chapter 1 : Introduction to C	1.1 – 1.42
Chapter 2 : Control Statements	2.1 – 2.82
Chapter 3 : Functions	3.1 – 3.26
Chapter 4 : Arrays	4.1 – 4.56
Chapter 5 : String Handling	5.1 – 5.18
Chapter 6 : Structures and Unions	6.1 – 6.10
Chapter 7 : Pointers and Dynamic Variables	7.1 – 7.4
Appendix A : Some More Solved Problems	A1 – A8
Appendix B : Frequently asked questions in interviews on C language	B1 – B8

CHAPTER 1

INTRODUCTION TO C

We know that, a computer is a problem solving machine. To solve a problem using a computer we need to prepare a set of instructions. Such a set of instructions is known as a program. To prepare a program we have to use a language which computer can understand. Computer can understand only one language known as machine language (binary language).

In this language each instruction is a sequence of zeros and ones. Learning and remembering such sequences is really difficult. So they developed some languages known as the high level programming languages (HLL), which are English like languages. For example, C, C++, Java, Pascal, Basic, etc. One can easily learn a language like C. But a program written in a language like C can not be executed directly since computer recognizes only the machine language.

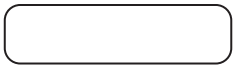
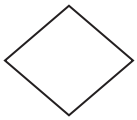
So after creating a program in a language like C that should be translated into the machine language of the computer. This can be done by a program known as a compiler. For each type of machine and operating system (OS) there should be a unique compiler. There are many C language compilers. Turbo C is a popular C language compiler in India.

1.1 ALGORITHM AND FLOWCHARTS

Normally we initally create a set of instructions for solving a problem in English. Such a set of instructions is known as an algorithm. Or in other words an algorithm is a step by step procedure for solving a problem. **To create an algorithm, we need to analyse the problem and decide the inputs, outputs and the necessary processing required to get the expected output.**

A diagramatic representation of an algorithm is known as a flow chart.

Some shapes that are used in flowcharts.

	Start/Stop
	Input/Output
	Process (Computations)
	Decision box (Condition box)
	Specify the direction
	Connector

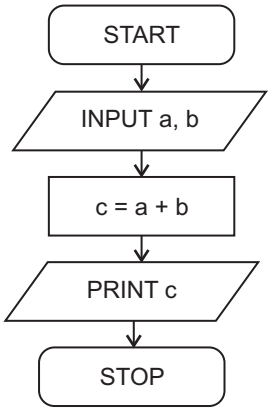
Create an algorithm and a flowchart for the following problems :

1. To find sum of two numbers.

   Algorithm

   1. START
   2. INPUT a, b
   3. LET C = a + b
   4. PRINT C
   5. STOP

Flowchart :

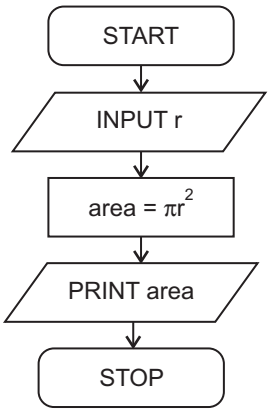


2. To find area of a circle using its radius.

   Algorithm

   1. START
   2. INPUT r
   3. LET area = πr^2
   4. PRINT area
   5. STOP

Flowchart :

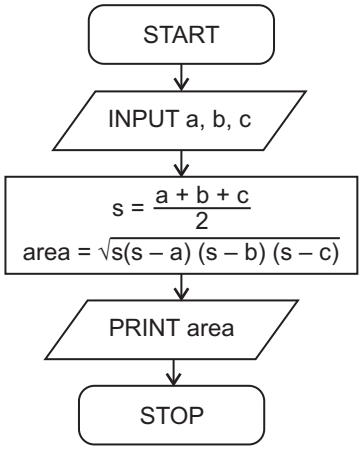


3. To find the area of a triangle using its sides.

   Algorithm

   1. START
   2. INPUT a, b, c
   3. LET $s = \frac{a + b + c}{2}$
   4. Let area = $\sqrt{s\,(s - a)\,(s - b)\,(s - c)}$
   5. PRINT area
   6. STOP

Flowchart :

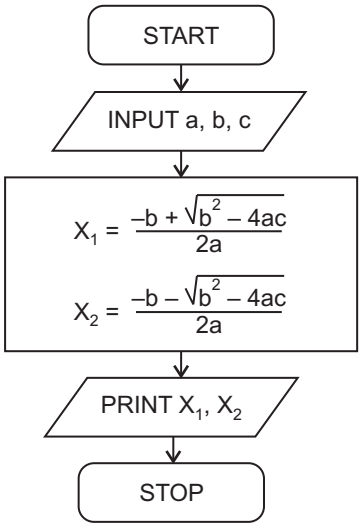


4. To find the roots of a given quadratic equation.

   Algorithm

   1. START
   2. INPUT a, b, c
   3. LET $x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$
   4. LET $x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$
   5. PRINT x_1 and x_2
   6. STOP

Flowchart :

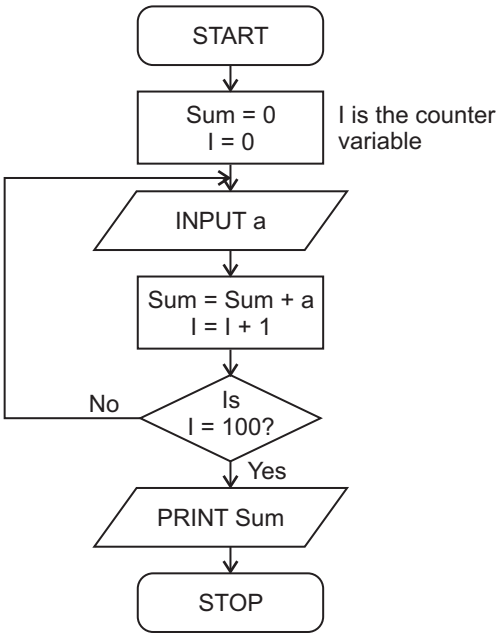


5. To find sum of 100 given numbers.

   Algorithm

   1. START
   2. SUM = 0
   3. REPEAT step 4 and 5, 100 times
   4. Input = a
   5. SUM = SUM + a
   6. PRINT SUM
   7. STOP

Flowchart :

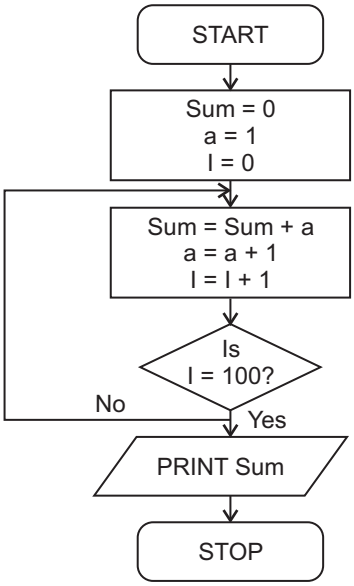


6. To find sum of first 100 natural no's.

   Algorithm

   1. START
   2. SUM = 0, a = 1
   3. REPEAT step 4 and 5, 100 times
   4. SUM = SUM + a
   5. a = a + 1
   6. PRINT SUM
   7. STOP

Flowchart :

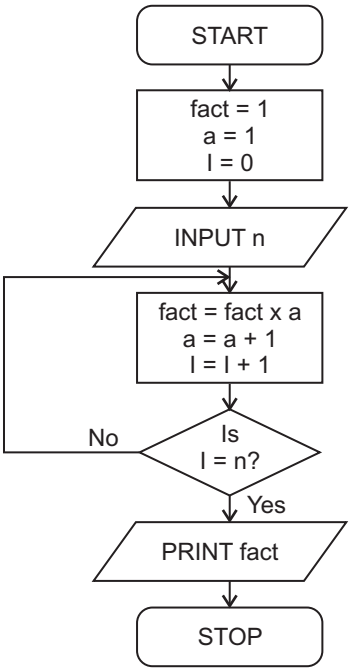


7. To find the factorial of a given number.

   Algorithm

   1. START
   2. fact = 1, a = 1
   3. INPUT n
   4. REPEAT steps 5 and 6, n times
   5. fact = fact $\times$ a
   6. a = a + 1
   7. PRINT fact
   8. STOP

Flowchart :

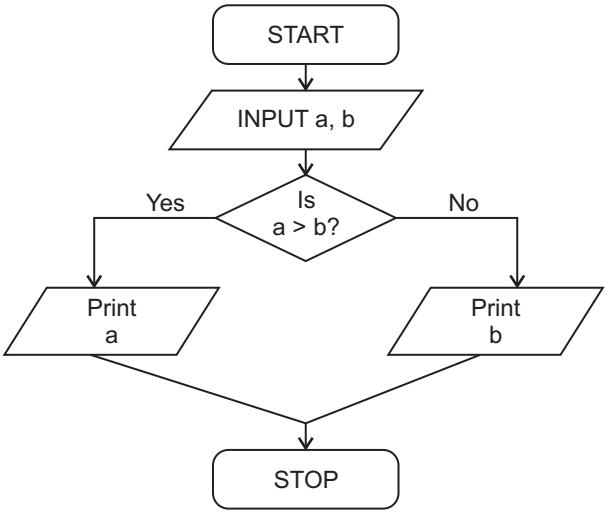


8. To find largest of two numbers.

   Algorithm

   1. START
   2. INPUT a, b
   3. IF a > b THEN PRINT a ELSE PRINT b
   4. STOP

Flowchart :

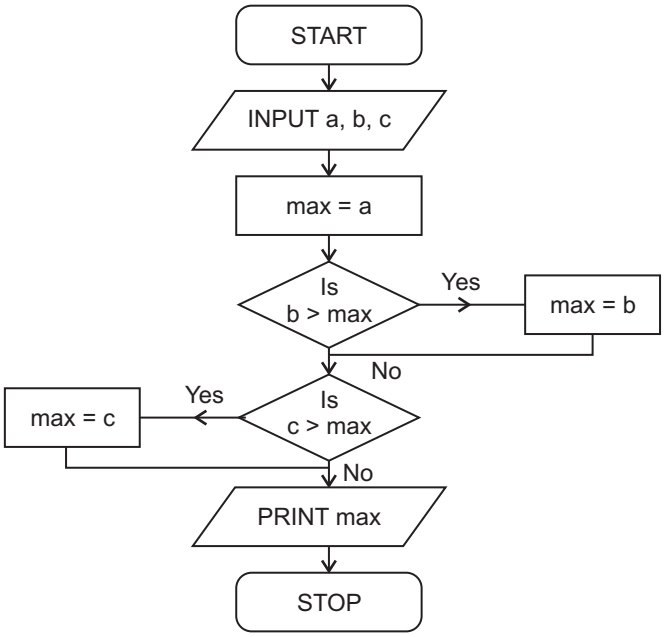


9. Draw a flowchart to find largest of 3 nos.

   Algorithm

   1. START
   2. INPUT a, b, c
   3. max = a
   4. If b > max then max = b
   5. If c > max then max = c
   6. Print max
   7. STOP

Flowchart :



Structure of a C program :

Consider the following program for finding sum of two numbers.

```
#include <stdio.h> → header file inclusion
void main ()
{
  int a, b, c;                          → variable declaration
  printf("Enter 2 no:\n");
  scanf ("%d%d", &a, &b);               → statements
  c=a+b;
  printf("%d", c);
}
```

Note :

C is a case sensitive language. That means, upper and lower case English letters are treated differently. Normally a C language program is always in lower case form.

stdio.h

This is a header file that contains declarations of some input and output operations (functions). There are many header files which are part of a C language compiler. Each header file contains declarations of some functions. If we use a function declared in a header file in our program then that file should be included in our program using #include. In the above program we use scanf for input and printf for output, both are declared in stdio.h. Some header files are given below:

ctype.h
math.h
string.h
stdio.h
alloc.h

Keywords and Identifiers :

A program can be treated as a collection of words and some punctuation marks such as comma, semicolon etc. Words are classified into two categories.

Keywords : Words that are defined in C language for some specific purposes are known as keywords. There are 32 keywords in standard C language. Keywords should be used in lower case form. Some keywords are given below.

void	char	while
int	short	do
float	long	… etc.
double	for	

Identifiers : Words that are used as variable names, function names and constant names are known as identifiers.

For example:

In the above program,
a is an identifier.
b and c are also identifiers.
scanf and printf are also identifiers.

Rules to form an identifier in C:

For an identifier we can use
(a) English letters (A–Z, a–z)
(b) Digits (0–9)
(c) Underscore (_)

The starting character of an identifier should be an English letter or underscore. After the first character we can use any number of English letters, digits and underscore. But only first 31 or 32 characters will be considered. Keywords should not be used as identifiers.

For Example:

a	sum_prod
a1	A

Q. Check whether following words are valid identifiers or not? Give reasons.
sum (valid)
1a (not valid,since it starts with a digit)

a b (not valid,since there is a blank space)

a ? b (not valid,since there is a ? which is not permitted)

a_b (valid)

int (not valid since int is a keyword)

Int (valid)

g.c.d. (not valid,since .is not permitted)

Binary number system

A binary number is a sequence of zeros and ones. In computer's memory everything is stored in binary form.

Q. Convert following numbers into binary form

14

2	14	0
2	7	1
2	3	1
2	1	

$(14)_{10} = (1110)_2$

2. 17

2	17	1
2	8	0
2	4	0
2	2	0
	1	

$(17)_{10} = (10001)_2$

A single 0 or 1 in a binary number is known as a bit. A sequence of 8 bits is known as a byte.

For e.g. 00001010.

Q. Convert into binary.

1. 12.65

2	12	0
2	6	0
2	3	1
	1	

$(12)_{10} = (1100)_2$

Converting .65

$.65 \times 2 = 1.30$,1 is the first bit after decimal point

$.30 \times 2 = 0.60$,0 is the second bit after decimal point

$.20 \times 2 = 0.40$,0 is the third bit after decimal point

$(.65)_{10} = (.101)_2$

So, $(12.65)_{10} = (1100.101)_2$

Q. Convert following binary no's into decimal form.

10111

$1 \times 2^0 + 1 \times 2^1 + 1 \times 2^2 + 0 \times 2^3 + 1 \times 2^4$

$1 + 2 + 4 + 0 + 16$

$= (23)_{10}$

2. 1100.11

$0 \times 2^0 + 0 \times 2^1 + 1 \times 2^2 + 1 \times 2^3$

$+ 1 \times 2^{-1} + 1 \times 2^{-2}$

$= 4 + 8 + ½ + 1/4$

$= (12.75)_{10}$

Note :

A number is stored in computer's memory in a fixed number of bytes. In this form if the starting bit (left most) is 0 then the number is positive and if it is 1 then the number is negative. A negative number will be 2's complement form of the corresponding positive number. To find 2's complement form we have to scan the binary form of the number from right to left. Till we get first 1 we keep the bits as it is. Thereafter replace each 0 by 1 and 1 by 0. Another method to find 2's complement is: add 1 to 1's complement form of the given number.

For example:

6 in 2 bytes form

6 = 0000 0000 0000 0110

Q. Find binary form of –6 in 2 bytes

6 = 0000 0000 0000 0110

Finding 2's complement

$\therefore$ –6 = 1111 1111 1111 1010

(here we use the scanning from right to left method)

Note :

$1 = 1 = 2^1 - 1$

$11 = 3 = 2^2 - 1$

$111 = 7 = 2^3 - 1$

$1111 = 15 = 2^4 - 1$

$1 = 1 = 2^0$

$10 = 2 = 2^1$

$100 = 4 = 2^2$

$1000 = 8 = 2^3$

Data types in C :

We know that all variables used in a program should be declared in the beginning. During compilation, for each declared variable a memory location will be allocated. Data types are used for declaring variables. Data type decides the size of memory allocated for a variable. In c data types can be classified into 4 categories.

1. Basic data types (primary data types)
2. User defined data types
3. Derived data types
4. Empty set.

In this section we cover only the basic data types (or primary data types)

Basic data types : There are 4 basic data types in C; which are given below :

int
float
double
char (all are keywords)

int : stands for integer

e.g. :15
–25 etc.
Memory :2 bytes
Range : -2^{15} to $2^{15} - 1$
(–32768 to +32767)

float : stands for floating point number, single precision real no.

Float numbers can be represented in two different ways :

1. Decimal form
   e.g. : 10.56
   138.3754
   –234.15 etc.

2. Exponential form
   23.56 e+1
   Or means 23.56×10^{1}
   23.56 E+1
   2.356 e+2
   Or means 2.356×10^{2}
   2.356 E+2
   235.6 e–1
   Or means 235.6×10^{-1}
   235.6 E–1

Memory : 4 bytes
Range : 3.4e–38 to 3.4e+38
Precision : 6 digits after decimal point (single precision)

double : stands for double precision real number
Double numbers are also be represented in two different ways
(1) Decimal form, (2) Exponential form. Same as float.
Memory : 8 bytes
Range : 1.7e–308 to 1.7e+308
Precision : 14 digits after decimal point.

char : stands for character

E.g. : 'A' 'B' 'C' '+' '?' '7'
In computer's memory, each character is represented by a number known as ASCII code (American Standard Code for Information Interchange). In this coding scheme the code of A is 65, code of B is 66 code of a is 97, code of b is 98. Starting code is 0 and the final code is 255.
Memory : 1 byte
Range : 256 characters.

Variable declaration :

Syntax :

data type list of variables;
(separated by ,)

```
e.g. : int a;
         float b, c, d;
       char e;
       double f;
```

Variables can be initialized at the time of declaration.

e.g. : int a=5, b;
i.e., initial value of a=5.
If a variable is not initialzed then it may contain some unknown value known as a garbage value.

Qualifiers :

Qualifiers are keywords that can be used to qualify the basic data types to form new data types. There are 4 qualifiers, which are given below :

signed
unsigned
short all are keywords
long

Data types using qualifiers :

1. signed char
   Memory : 1 byte
   Range : –128 to +127
2. unsigned char
   Memory : 1 byte
   Range : 0 to 255
3. signed int
   Memory : 2 bytes
   Range : –32768 to 32767
   (Some as int)
4. unsigned int
   Memory : 2 bytes
   Range : 0 to 65535
   (0 to $2^{16} - 1$)
5. short int
   Memory : 1 byte
   Range : –128 to +127
   (-2^7 to $2^7 - 1$)
6. long int
   Memory : 4 bytes
   Range : -2^{31} to $2^{31} - 1$
7. long double
   Memory : 10 bytes
   Range : 3.4e–4932 to 1.1e+4932
   Precision : more than 19 digits after decimal point

8. unsigned short int
   Memory : 1 byte
   Range : 0 to 255
   (0 to $2^8 - 1$)
9. unsigned long int
   Memory : 4 bytes
   Range : 0 to $2^{32} - 1$

Variable declaration :

E.g. : unsigned int a;

OR

unsigned a;

Note : After a qualifier, if basic type is not mentioned, then int will be assumed.

short int a;

OR

short a;

long double b;

Operators in C :

C has a very rich set of operators. These operators are classified into 8 categories as follows :

1. Arithmetic operators :
   +, –, *, / and % (most operator)
2. Relational operators :
   < (less than) > (greater than)
   <= (less than or equal to)
   >= (greater than or equal to)
   == (equal to) != (not equal to)
3. Logical operators :
   && (and)
   || (or)
   ! (not)
4. Assignment operators :
   =
   +=, –=, *=, /=, %= etc.
5. Increment and decrement operators :
   ++
   ––