Coimbatore-35
An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A+' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF AIML

OBLEM SOLVING AND C PROGRAMMING

I YEAR - I SEM

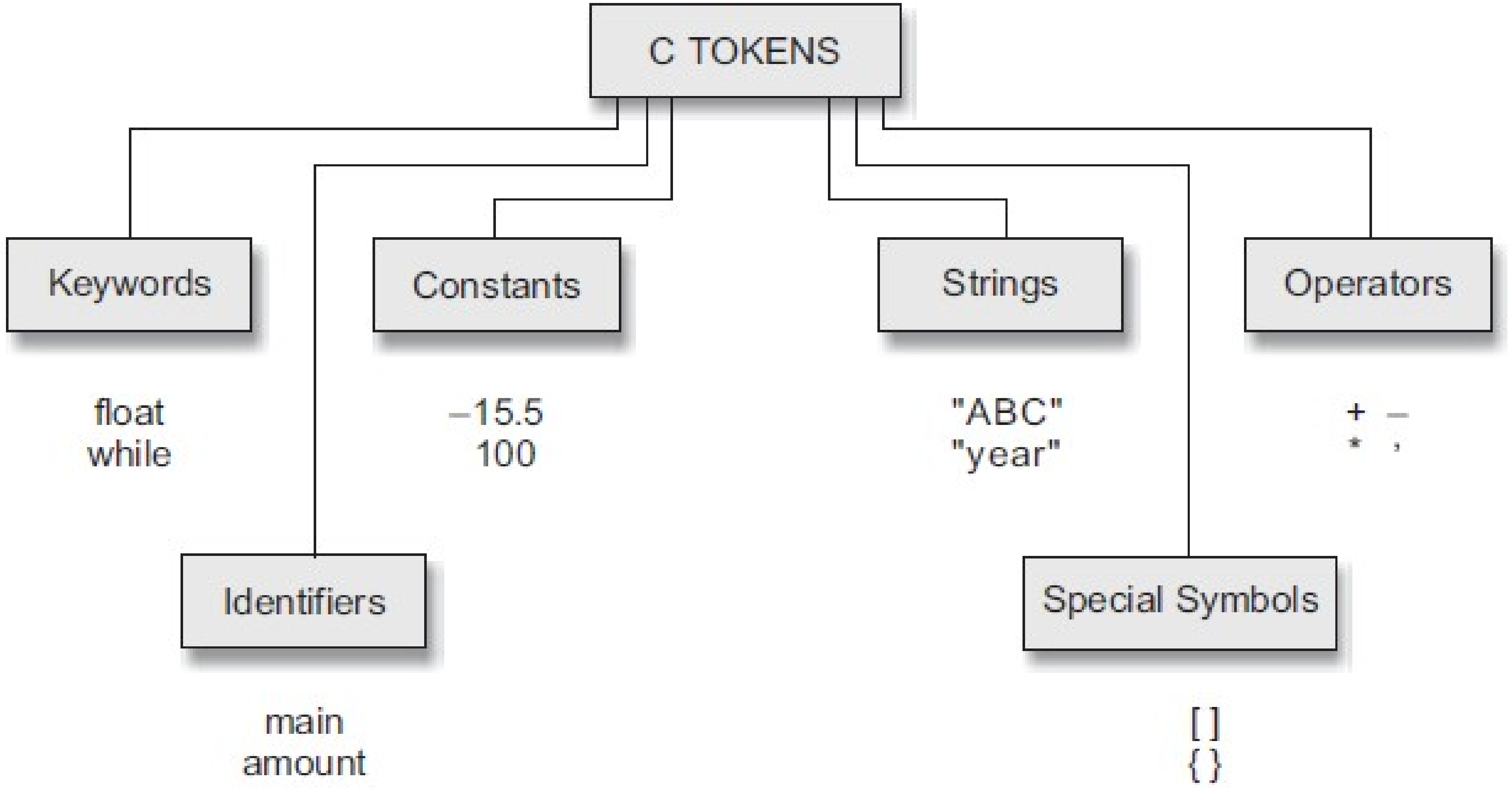
UNIT 2 – C Programming Basics

3 –Keywords, Identifiers, Constants & Delimeters

nming language is designed to help process certain kinds of d
characters and strings and to provide useful output known as infor
of processing of data is accomplished by executing a seq
s called a program.
ructions are formed using certain symbols and words according t
syntax rules (or grammar).
gram instruction must confirm precisely to the syntax rules of the l
other language, C has its own vocabulary and grammar.
cters in C are grouped into the following categories:
etters
igits
pecial characters
hite spaces

Letters	**Digits**
Uppercase A.....Z	All decimal digits 0 9
Lowercase a.....z	
Special Characters	
, comma	& ampersand
. period	^ caret
; semicolon	* asterisk
: colon	– minus sign
? question mark	+ plus sign
‘ apostrophe	< opening angle bracket
“ quotation mark	(or less than sign)
! exclamation mark	> closing angle bracket
| vertical bar	(or greater than sign)
/ slash	(left parenthesis
\ backslash	) right parenthesis
~ tilde	[left bracket
_ under score	] right bracket
$ dollar sign	{ left brace
% percent sign	} right brace
	# number sign
White Spaces	
Blank space	
Horizontal tab	
Carriage return	
New line	
Form feed	

ge of text, individual words and punctuation marks are called toke
in a C program the smallest individual units are known as C toker
types of tokens as shown in Fig.
ns are written using these tokens and the syntax of the language.



C tokens and examples

C word is classified as either a “keyword” or an “identifier”.
words have fixed meanings and these <u>meanings cannot be change</u>
words must be written in lowercase.

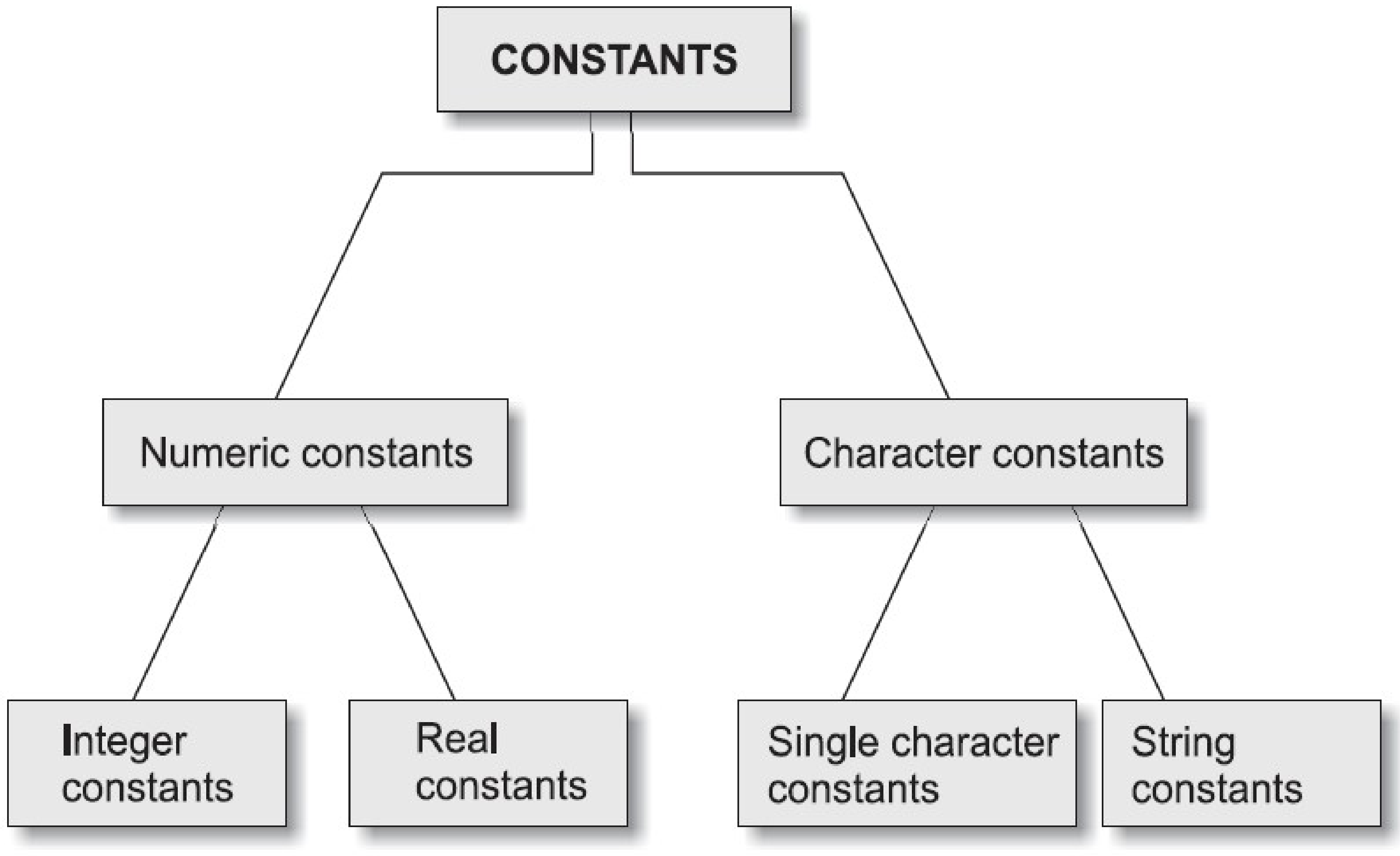
ANSI C Keyword

	double	**int**
	else	long
	enum	register
	extern	return
	float	short
ue	for	signed
	goto	sizeof
	if	static

ers refer to the names of variables, functions and arrays.
are user-defined names.
ppercase and lowercase letters are permitted.
derscore character is also permitted in identifiers.
<u>or Identifiers:</u>
t character must be an alphabet (or underscore).
st consist of only letters, digits or underscore.
y first 31 characters are significant.
not use a keyword.
st not contain white space & special symbols.
<u>id Identifiers:</u>
NAME, SUB, TOT_MARKS, _TEMP, Y2K
<u>alid Identifiers:</u>
rn, $stay, 1Record, STD NAME.

in C refer to fixed values that <u>do not change</u> during the ex

several types of constants



Basic types of C constants

onstant refers to a sequence of digits.

ree types of integers, namely:

Integer

eger and

imal Integer.

ntegers:

f a set of digits, 0 through 9, preceded by an optional – or + sign.

nples of decimal integer constants are: 123 – 321 0 654321 +78

spaces, commas, and non-digit characters are not permitted betw

le, 15 750 20,000 $1000 are illegal numbers.

ger:

f any combination of digits from the set 0 through 7, with a leadin

nples of octal integer are: 037 0 0435 0551

mal Integer:

e of digits preceded by 0x or 0X is considered as hexadecimal inte

also include alphabets A through F or a through f.

A through F represent the numbers 10 through 15.

are the examples of valid hex integers: 0X2 0x9F 0Xbcd 0x

se octal and hexadecimal numbers in programming.

bers are inadequate to represent quantities that vary continuously, eights, temperatures, prices, and so on.

ities are represented by numbers containing fractional parts like 17

ers are called real (or floating point) constants.

mples of real constants are: 0.0083 –0.75 435.36 +247.0

ers are shown in decimal notation, having a whole number followe e fractional part.

e to omit digits before the decimal point, or digits after the decimal

.95 –.71 +.5 are all valid real numbers.

er may also be expressed in exponential (or scientific) notation.
e, the value 215.65 may be written as 2.1565e2 in exponential nota
ultiply by 102.
form is: **mantissa e exponent**
a is either a real number expressed in decimal notation or an integ
nt is an integer number with an optional plus or minus sign.
separating the mantissa and the exponent can be written in either l

ponent causes the decimal point to "float", this notation is said to
oating point form.
f legal floating-point constants are: 0.65e4   12e-2   1.5e+5   3.18E
vhite space is not allowed.
notation is useful for representing numbers that are either very lar
e.
e, 7500000000 may be written as 7.5E9 or 75E8.

racter constant (or simply character constant) contains a single cha
r of <u>single quote </u>marks.

character constants are: ‘5’ ‘X’ ‘;’ ‘ ’

e character constant ‘5’ is not the same as the number 5.

stant is a blank space.

onstants have integer values known as ASCII values mapped to eac
e, the statement printf(“%d”, ‘a’); would print the number 97, the A

e statement printf(“%c”, ‘97’); would output the letter ‘a’.

es for all characters are given below.

character constant represents an integer value, it is also possible to
perations on character constants.

ASCII Table

Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char
0	0	0		32	20	40	[space]	64	40	100	@	96	60	140	`
1	1	1		33	21	41	!	65	41	101	A	97	61	141	a
2	2	2		34	22	42	"	66	42	102	B	98	62	142	b
3	3	3		35	23	43	#	67	43	103	C	99	63	143	c
4	4	4		36	24	44	$	68	44	104	D	100	64	144	d
5	5	5		37	25	45	%	69	45	105	E	101	65	145	e
6	6	6		38	26	46	&	70	46	106	F	102	66	146	f
7	7	7		39	27	47	'	71	47	107	G	103	67	147	g
8	8	10		40	28	50	(	72	48	110	H	104	68	150	h
9	9	11		41	29	51	)	73	49	111	I	105	69	151	i
10	A	12		42	2A	52	*	74	4A	112	J	106	6A	152	j
11	B	13		43	2B	53	+	75	4B	113	K	107	6B	153	k
12	C	14		44	2C	54	,	76	4C	114	L	108	6C	154	l
13	D	15		45	2D	55	-	77	4D	115	M	109	6D	155	m
14	E	16		46	2E	56	.	78	4E	116	N	110	6E	156	n
15	F	17		47	2F	57	/	79	4F	117	O	111	6F	157	o
16	10	20		48	30	60	0	80	50	120	P	112	70	160	p
17	11	21		49	31	61	1	81	51	121	Q	113	71	161	q
18	12	22		50	32	62	2	82	52	122	R	114	72	162	r
19	13	23		51	33	63	3	83	53	123	S	115	73	163	s
20	14	24		52	34	64	4	84	54	124	T	116	74	164	t
21	15	25		53	35	65	5	85	55	125	U	117	75	165	u
22	16	26		54	36	66	6	86	56	126	V	118	76	166	v
23	17	27		55	37	67	7	87	57	127	W	119	77	167	w
24	18	30		56	38	70	8	88	58	130	X	120	78	170	x
25	19	31		57	39	71	9	89	59	131	Y	121	79	171	y
26	1A	32		58	3A	72	:	90	5A	132	Z	122	7A	172	z
27	1B	33		59	3B	73	;	91	5B	133	[	123	7B	173	{
28	1C	34		60	3C	74	<	92	5C	134	\	124	7C	174	|
29	1D	35		61	3D	75	=	93	5D	135	]	125	7D	175	}
30	1E	36		62	3E	76	>	94	5E	136	^	126	7E	176	~
31	1F	37		63	3F	77	?	95	5F	137	_	127	7F	177	

stant is a sequence of characters enclosed in double quotes.

ers may be letters, numbers, special characters and blank space.

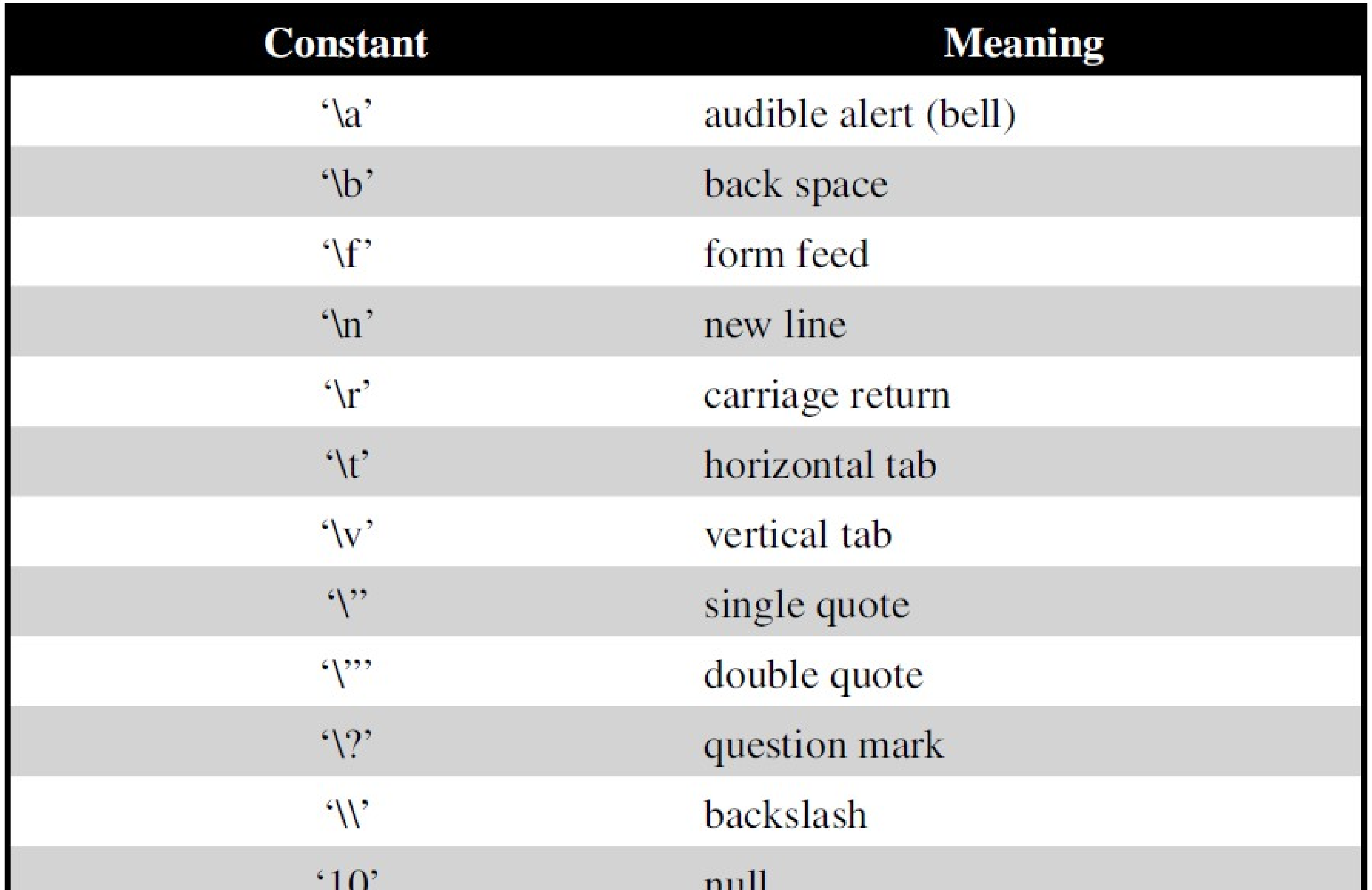
e: “Hello!” “1987” “WELL DONE” “?...!” “5+3” “X”

hat a character constant (e.g., ‘X’) is not equivalent to the single c
g., “X”).

ngle character string constant does not have an equivalent integer
nstant has an integer value.

rings are often used in programs to build meaningful programs.

ome special <u>backslash </u>character constants that are used in output
, the symbol ‘\n’ stands for newline character.
h backslash character constants is given in Table.
ch one of them represents one character, although they consist of t
cters combinations are known as <u>escape sequences.</u>



Constant	Meaning
‘\a’	audible alert (bell)
‘\b’	back space
‘\f’	form feed
‘\n’	new line
‘\r’	carriage return
‘\t’	horizontal tab
‘\v’	vertical tab
‘\’’	single quote
‘\”’	double quote
‘\?’	question mark
‘\\’	backslash
‘\0’	null

e symbols which has some syntactic meaning and has got significa
ot specify any operations.
delimiters list is show below.

L	NAME	MEANING
	Hash	Pre processor directive
	Comma	Variable delimiter used to separa
	Colon	Label delimeters
	Semi colon	Statement delimeters
	Parenthesis	Used in expressions or in functio
	Curly braces	Used for blocking c structure
	Square braces	Used along with arrays