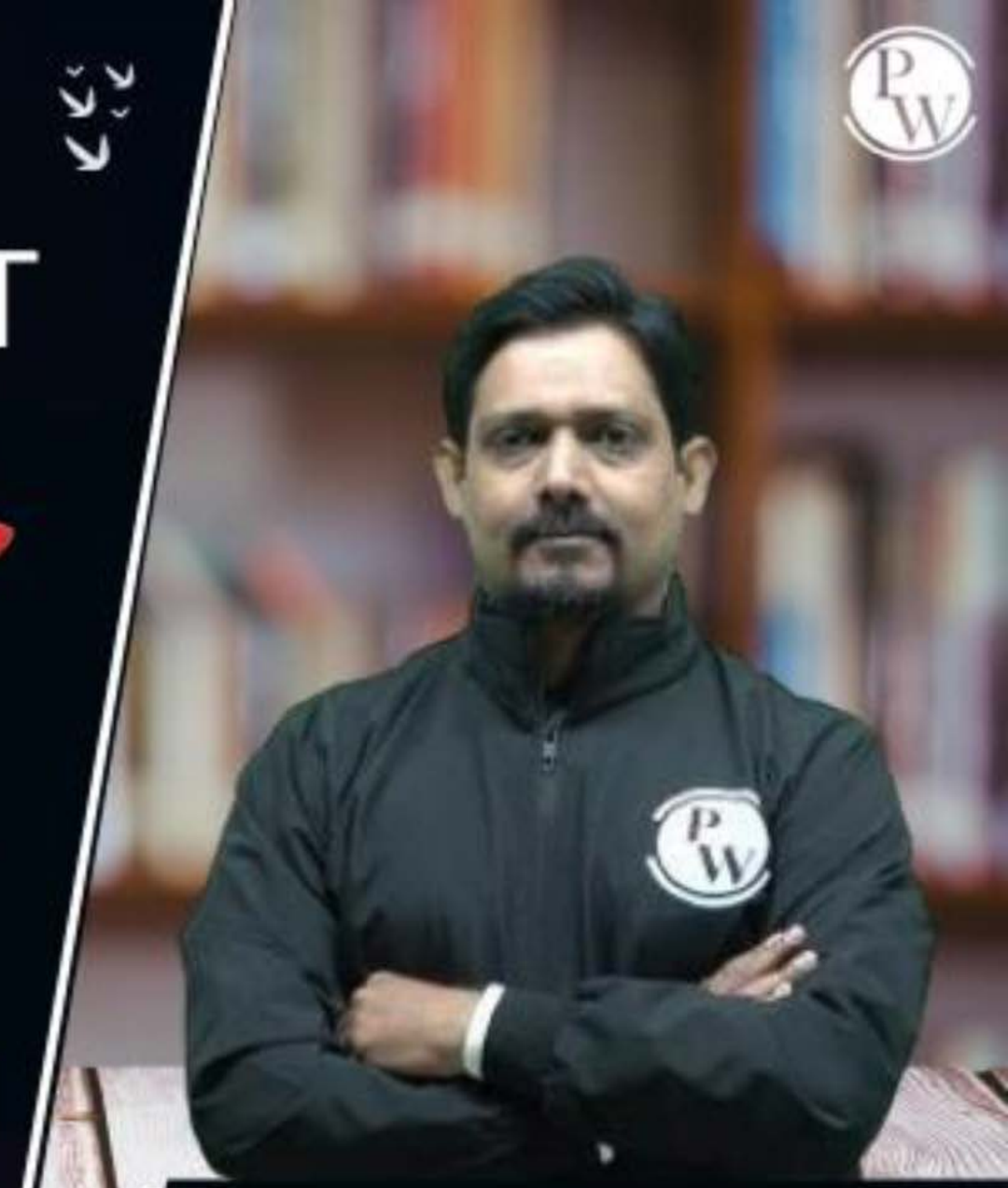
Computer Science & IT

C programming

Data Types & Operator

Lecture No. 01



By- Abhishek Sir

Topics to be Covered

- Topic — Tokens
- Topic — keywords, identifier, constant
- Topic — String, operator, Special character
- Topic — int char, float
- Topic

Slide

Topic : Token in C language

Slide

precedence table

Last operator

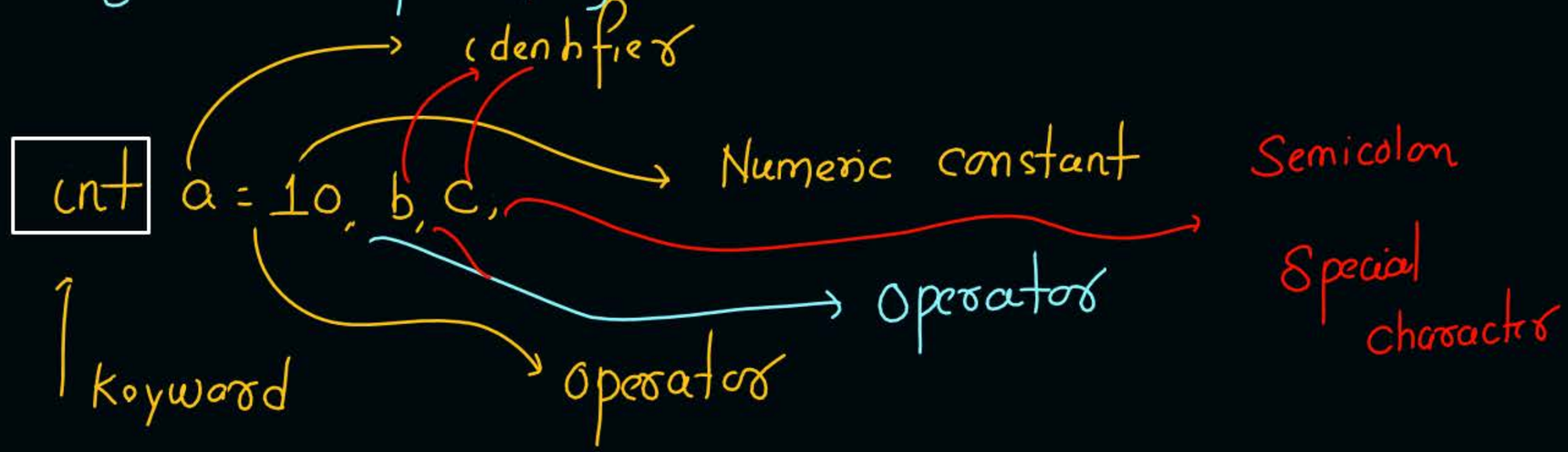
,

precedence Lowest

```
int a = 10, b, c;
```

← Declarative statement

Smallest logical unit of c program is called token



Topic : Reserve keyswords

Some words are reserved and can't be used as

Name of identifier.

Slide

Topic : Reserve keyswords

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while

Topic : Identifier

Name of variable or funchon is called identifier

Rule : It must start with underscore or alphabet followed by underscore/(a...z/A...z) alphabet/digit (0..9)

```
int Int;
```

C-Language is a case Sensitive Language

Slide

59

32 + 16 + 8 + 2 + 1

0 1 1 1 0 1 1

29

16 + 8 + 4 + 1

0 1 1 1 0 1

−24 (2's complement)

MSB - 0 +ve

1 - ve

−32 + 8

1 0 1 0 0 0

24 - 16 + 8

0 1 1 0 0 0

1 0 0 1 1 1

1 1

1 0 1 0 0 0

$1\,0\,1\,0\,0\,0$

$1\,0\,1\,0$

$3\;2\;1\;0$

$-1\times2^{3}+1\times2^{1}$

$-8+2=-6$

2's complement MSB weight is Negative

$1\,0\,1\,0$ (-6)

-10 - 2's complement

$0\ 1010$ $(+10)$ Add a sign

$1\ 0101$ 1's complement

$\begin{array}{r} {}^{1}\,1 \\ \hline 1\ 0\ 1\ 1\ 0 \end{array}$ Add 1

$4\ 3\ 2\ 1\ 0$

$-1\times2^{4} + 0\times2^{3} + 1\times2^{2} + 1\times2^{1} + 0\times2^{0}$

$-16 + 4 + 2 = -10$

Sign Extension

1010 — -6 4 bit

1 1 0 1 0
4 3 2 1 0

5 bit [extend the Sign]

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

$-16 + 8 + 2 = -6$

6 bit - 6

1 1 1 0 1 0

Topic : Constant

decimal No system int a = 10, → decimal constant

Binay constant : int a = 0b1010, (zero followed by alphabet b/B)
0B1010,

(0,1,2,3,4,5,6,7) Octal constant : Starts with zero int a = 012,
10

$1 \times 8^1 + 2 \times 8^0$
$8 + 2 = 10$

Slide

Topic : Constant

Hexadecimal , format specifier %x

0, 1, 2, 3, 4, 5, 6, 7, 8, 9 a, b, c, d, e, f

int a = 0xa;

0 — zero followed by alphabet x/X

$10 \times 16^0 = 10$

Slide

Topic : Constant

```
#include <stdio.h>
int main(){
    int a = 15;
    int b = 0b101;
    printf("%d", a+b);
    return 0;
}
```

0b101

Binary constant

A. 94
B. 272
C. 20
D. 80

101

$1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 5$

Slide

Topic : Constant

```
#include <stdio.h>
int main(){
    int a = 15;
    int b = 0101;
    printf("%d", a+b);
    return 0;
}
```

begin with 0

octal constant

(65)

65+15 = (80)

0 1 0 1
2 1 0

$1 \times 8^2 + 1 \times 8^0$

$64 + 1 = 65$

A. 94
B. 272
C. 20
D. ~~80~~

Slide

Topic : Constant

```
#include <stdio.h>
int main(){
    int a = 15;
   int b = 0x101;
   printf("%d", a+b);
    return 0;
}
```

0x101

Hexadecimal

$1\times16^2+1\times16^0$

$256+1=257$

A. 94
B. 272
C. 20
D. 80

$0x\,1\,0\,1$
$\quad 2\;1\;0$

Hexadecimal

$1\times16^2+1\times16^0$

$256+1=257$

257
15
272

Slide

Topic : Constant

```
#include <stdio.h>
int main(){
    int a;
    a = 29+0b1001+056+0xa
     printf("%d", a);
    return 0;
}
```

A. 94
B. 272
C. 20
D. 80

0b – Binary
0 (zero) octal
0x (Hexadecimal)

Slide

Topic : Constant

```
#include <stdio.h>
int main(){
    int a;
    a = 29+0b1001+056+0xa
     printf("%d", a);
    return 0;
}
```

29 - decimal

9
46
210
94

1001 Binary

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

$8+1=9$

056 octal

10

$5 \times 8^1 + 6 \times 8^0$

$40+6=46$

Slide

Topic : Constant

```
#include <stdio.h>
int main(){
    int a;
    a = 60+0b1111+014+0x10;
     printf("%d", a);
    return 0;
}
```

A. 94
B. 272
C. 103
D. 80

Slide

Topic : Constant

```
#include <stdio.h>
int main(){
    int a;
     a = 30/0b1111+014+0x10;
      printf("%d", a);
    return 0;
}
```

30/15 + 12 + 16

2 + 12 + 16 = 30

A. 94
B. 272
C. 103
D. 30

1111 = $2^{4}-1$ = 15

15

111 .. 1

n bits

unsigned $2^{n}-1$ =

014

$1\times8^{1}+4\times8^{0}=12$

0x10 = 1×16^{1} = 16

(10)

Signed = n bits (2's complement)

11111...1 = -1

n bits

Slide

0x ← Hexadecimal

← 2 digit

10, 11, 12, 13, 14, 15

Base 16 → 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, f

0xa1

10

0xa

1

$10 \times 16^1 + 1 \times 16^0$

160 + 1 = 161

$10 \times 16^0 = 10$

Topic : Constant

```
#include <stdio.h>
int main(){
   int a;
    a = 0101/011+0110+0x110;
    printf("%d", a);
   return 0;
}
```

A. 94
B. 351
C. 189
D. 80

Slide

Topic : Constant

```
#include <stdio.h>
int main(){
    int a;
     a = 0b1111+014+0x101/010;
     printf("%d", a);
    return 0;
}
```

A. 94

B. 59

C. 103

D. 30

Slide

Topic : Constant / Literals

Integer Literal: 42

Float Literal: 3.140000

Character Literal: A

Double Literal: 2.718280

String Literal: Hello

Slide

Topic : String

String literal

String

```
printf("%d", a)
```

string token

```
char *char = "string";
```

```
char ch[] = "string";
```

string token

Topic : Operator

Token

Data type & operators

++, --	Increment/ Decrement Operator
+, -, *, /, %	Arithmetic Operator
<, <=, >, >=, ==, !=	Relational Operator
&&, ||, !	Logical Operator
&, |, <<, >>, ~, ^	Bit wise Operator
=, +=, -=, /=, %=, &=, |=	Assignment operator
?:	Conditional Operaor

Control. Condihonal. if, else
-f Iterahve - loop

Short circuit code

Slide

Anything is special symbol : ( )

[ ]

;

Data type

1. Amount of Memory required
2. kind of type of value
3. Interpretation of data

1010

Topic : Special Symbol

Brackets [] Used to declare and access arrays.

Parentheses () Enclose function parameters and control expressions.

Braces {} Define blocks of code for functions, loops, etc.

Comma , Separates variables, parameters, or elements in a list.

Colon : Used in switch statements for labels.

Topic : Special Symbol

Semicolon ; Terminates statements.

Asterisk * Used for pointers and multiplication.

Pre-processor # Begins pre-processor directives like #include.

Period . Accesses members of a structure.

Topic : Data Types

In C, a data type specifies the

- kind of value a variable can hold,
- compiler allocates memory and
- interprets data

Topic : Primitive Data Types

- int
- character
- float

Topic : Integer Data Type

int → Memory — (System dependent) 4Byte

integer value it will take

Interpretation : 2's complement 1 bit reserved for sign

+ve unsigned

-ve 2's complement of +ve No

Slide

Topic : Modifiers signed and unsigned

4 bit unsigned

Min ⟶ 0

unsigned

%u

Max – 1111

$2^4 - 1 = 15$

10

1010 ↙ Binary

only +ve interpretation

Range $0 - 2^n - 1$

Singed

%d / %i

2's complement

$n = 4$

$-2^{4-1} = -2^3 = -8$

to

$2^{n-1} - 1$

$2^{4-1} - 1 = 8 - 1 = 7$

Range 4 bit

Range -2^{n-1} to $2^{n-1} - 1$

Slide

int = 4B

Short = 2B

long = 8B

long long : 8B Maximum size of int is 8Byte

Topic : int, short, long

```
#include<stdio.h>
int main() {
        int a=10;
        short a1;
        short int a3;
        long a2;
        long int a4;
        unsigned int a5;
        unsigned short int a6;
        unsigned short a7;
        unsigned int a8;
        return 0;
```

%ld

unsigned long int, %lu

Topic : int, short, long

```
#include<stdio.h>
int main() {
   int a=10;
   printf("%lu ", sizeof(unsigned));
   printf("%lu ", sizeof(short));
   printf("%lu ", sizeof(long));
   printf("%lu ", sizeof(long long));
   return 0;
}
```

4
2
8
8Byte

Slide

Character

Hello

01001000 01100101 01101100 01101100 01101111

8bits

Slide

char ch. ⟶ Memory Required :- 1 Byte 8 bits

value :- integer

Interpretation integer interpretation character

ASCII table

American standard code

for information interchange

ASCII value

0...9	48...57
A...Z	65...90
a...z	97...122

```
char ch = 'a';
```

printf("%c", ch), ⟶ a (without quote)

printf("%d", ch), ⟶ ASCII value

97.

```
char ch = 'a' + 2;
```

printf("%c", ch); ⟶ (without quote)

printf("%d" ch); 99 print.

Slide

ASCII Table

0	NUL	16	DLE	32		48	0	64	@	80	P	96	`	112	p
1	SOH	17	DC1	33	!	49	1	65	A	81	Q	97	a	113	q
2	STX	18	DC2	34	"	50	2	66	B	82	R	98	b	114	r
3	ETX	19	DC3	35	#	51	3	67	C	83	S	99	c	115	s
4	EOT	20	DC4	36	$	52	4	68	D	84	T	100	d	116	t
5	ENQ	21	NAK	37	%	53	5	69	E	85	U	101	e	117	u
6	ACK	22	SYN	38	&	54	6	70	F	86	V	102	f	118	v
7	BEL	23	ETB	39	'	55	7	71	G	87	W	103	g	119	w
8	BS	24	CAN	40	(	56	8	72	H	88	X	104	h	120	x
9	HT	25	EM	41	)	57	9	73	I	89	Y	105	i	121	y
10	LF	26	SUB	42	*	58	:	74	J	90	Z	106	j	122	z
11	VT	27	ESC	43	+	59	;	75	K	91	[	107	k	123	{
12	FF	28	FS	44	,	60	<	76	L	92	\	108	l	124	|
13	CR	29	GS	45	-	61	=	77	M	93	]	109	m	125	}
14	SO	30	RS	46	.	62	>	78	N	94	^	110	n	126	~
15	SI	31	US	47	/	63	?	79	O	95	_	111	o	127	DEL

Space

Slide

float data type : real No

Memory requirement : float : 4 Byte - Single precision %f

double : 8 Byte - Double precision %f or %lf

long double : 12 Byte - %Lf

Value real No

Interpretation : IEEE-754

Float Data Type

Representation of Non-Integer Values:

Precision:

Approximate Representation:

Memory Efficiency:

Float Data Type

Float

Double:

Long double

Float Data Type

```
int main() {

   float pi = 22.0/7;
   double pi1 = 22.0/7;
   long double pi2= 22.0/7;

   printf("%.4f ", pi);
   printf("%.10lf ", pi1);
   printf("%.8f ", pi1);
   printf("%.12Lf ", pi2);
   return 0;
}
```

```
int main(){
float pi = 22.0/7;

printf ("%f", pi%2);

return 0,

}
```

Error

% operator does not apply on floating point

Slide

2 mins Summary

Topic Tokens

Topic constant, identifier, keywords, operators, string

Topic Binary, octal, Hexadecimal
0b 0 0x

Topic integer, char, float

Topic

Slide

t.me/Abhishekshormapw

THANK - YOU

Slide