

16th Aug '19

UNIT - 1

what is prog. lang?

H T L
L F O

Binary → 0, 1

Machine level.

→ closer to machine low.

→ closer to user HLL.

↳ mid - somewhere in b/w.

00110110111 } programming binary

a = b + c } programming

ADD B, C } Assembly.

why we need Assembly?

Problem in converting into binary
Bcz everyone writes different.

• also called
Mother lang.

Pointers

-int a = 10

ALGOL (1960)

BCPL (1967)

B (1970)

C (1972) - Dennis

Ritchie

(AT&T Bell labs.)

K&R C (1978)

Kernighan
& Ritchie

ANSI C (1989)

ANSI/ISO C (1990) Standard

c99 (1999)

History of C

5 mark

① ALGOL (1960)

algorithmic lang.



② BCPL (1967)

Binary combined binary lang.
prog.



③ B (1970)



④ C — Dennis Ritchie (1972).

AT&T Bell labs.

⑤ K&R C (1978)

The C prog lang.

By Brian

Kernighan

Dennis Ritchie

standard
used

⑥

ANSI C (1989)



⑦ ANSI / ISO C (1990)



⑧ c99 (1999)

#

features of c

1. Case sensitive
2. Portable | Platform oriented | independent
3. Inbuilt functions | predefined
4. Procedure | function oriented

eg. Result
→

→ Total
→ Average ↗
→ Percentage .

5. fast & efficient

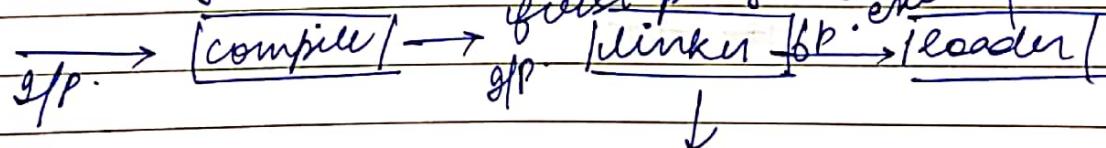
6. Robust (reliable)

Want
let errors pass .

7. structural



first prog . c



[library]
fn.

diff for
every
comp.
obj
etc
prog
const
acc
var
func

First C program.

```
#include <stdio.h> → preprocessor  
int main ()  
{  
    /* My first C Prog */  
    printf ("Hello World");  
    return 0;  
}
```

2 ways.

multi single
/* */ //

Misuse - increase LOC. of actual software.

Q. | ID card
 | Name:
 | Course:
 | Address:
 |

no

```
# include <stdio.h>
void main()
{
```

```
    printf (" ID CARD ");
    printf (" NAME: Kaushiki ");
    printf (" COURSE: BCA ");
    printf (" ADDRESS: B-9/176 ");
```

y

OR.

```
C printf (" IDCARD \n
          Name: Kaushiki \n
          Course : BCA \n
          Address: B-176 ");
```

};

[tab] t] escape sequence.
New line [n]

OR.

~~for~~ pr

22nd
10/
1pm / /

Character Sets

Digits
(0-9)

Alphabets
(a-z
A-Z)

Special symbols

! . ,
; * + - /
@ [] < >
{ } ()

white space
tab
space.
newline

C Token - smallest individual unit.

include < — >

Keywords Identifiers Constants strings operators

Numerics char.
↳ int ↳ single
↳ Real ↳ multiple

<u>Int:</u>	Decimal (10)	0-9	123
	Octal (8)	0-7	0123
	Hexa (16)	0-9 A-F	0x123
	Binary (2)	0-1	

Real:

3.143314

Mantissa E/e Exponent

3.14 e²
3.14 x 10²

(11 of eⁿ, variable)

3.1433



314.33

341.345



341345 x 10²
3.41345 e²

single char const. 'a' "a"

ASCII

ch = ↗ ASCII.

Variables

syntax

{ data type } > { name }

Declaration || int a;

Initialisation

Rules.

- keyword x
- alphabet or -
- — only spe chas.
- max length 31
- Case sensitive

Data Types in C

syntax :

(Datatype) Identifier = some value.

uses:-

- 1). type of value
- 2). return type of fn.
- 3). type of arguments for functions.

→ Types of datatype

- optional
- 1). **Primary (fundaments)** - int, char, float [void]
 - 2). **User defined** - fns., structures, unions, typedef, enumerations.
 - 3). **Derived** - Pointers, arrays, fns.
- ~~unit 3~~

Primitive

Numeric

int float
int double

(1 word)

character

char

void

(gives work of
in powers of 2)

long

short

unsigned

signed

int

Modifier -

int = (-2^8) to $(2^8 - 1)$

short

long

short int

int

long int

Any no. is 50
(~~max~~ ~~10~~, 0);

signed & unsigned

↓
default.

variable will not
take -ve.

~~signed into $= -32$~~

~~110~~

M&B.

signed bit

8 bit no.

if

-32

then MSB
takes up the
sign.

unsigned
data lists.

-2^8 to 2^8

→ no. of decimal
places.

→ float - 32 bit → 5 bits precision

double - 64 bits → 14 bits

→ char. → signed (-2^7 to $2^7 - 1$)

→ unsigned (-2^8 to $2^8 - 1$).

→ char → 1 byte.

- Q. WAP to calculate sum, average & % in class 12. (5 mark subject)
- Q. WAP to divide 2 integers & display result in int & float

#include < stdio.h >

void main()

{

// Program to calculate
sum, average & % in 12

int a = 80, 75, 90, 92, 77,

float a, b, c, d, e, f, g;

a = 77; b = 79; c = 80; d = 89;

e = 76;

f = (a + b + c + d + e) / 5;

printf (" Average Marks : ")

printf f;

printf ("")

Data Types

Primary

User defined

1. `typedef`
2. `Enumerations`

`typedef` - Type definition

Syntax:

`typedef int identifier;`

e.g. `typedef int basino`

\hookrightarrow `basino a;`

Enumerations

Syntax:

`enum < identifier > { v1, v2, v3 ... vn };`

e.g. `enum a { a };`

~~enum subject { SIT, TC, Phy, Maths, C };~~

~~subject fav;~~

~~subject rate;~~

~~fav = 0; // then it is 0.~~

```
if ( fav = 'C' )
    printf ("Good");
else printf ("Go");
```

Output & Input

1). `printf ("Hello");`

~~printf ("sum");~~

2.) `printf ("format specifier", variable list)`
a, b, c

3.) `printf ("%d", int var)` %d → int

%f → float

%c → char.

%e → if you want exponent

eg. Marks = 50

`printf ("Marks = %d", sum);`
Marks %d

4). `FIT = 60 C = 60 Maths = 50.`

`printf ("FIT = %d C = %d
Maths = %d", fit, c, maths);`

5) `printf ("sum = %d", a+b);`

6.) `scanf ("format specifier", & a);`

↓
address of
operator

eg. ~~scanf ("%d", &a);~~

~~scanf ("%d %d %f", &a, &b, &c);~~

```
int ra;  
printf("Enter your roll no.");  
scanf("%d", &a);  
printf("Roll no = %d", a);
```

#define
the 2
#define
the 2

Symolic constt

always
UPPER CASE

→ #define ABC 10 ↳ #define PI 3.14

→ const variable

```
const int p = 3.14;
```

Arrays

```
#include <stdio.h>
```

```
int main()
```

C

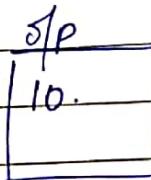
```
int x = 5;
```

```
x = 10;
```

```
printf("%d", x);
```

```
return 0;
```

y.



→ Array is a set of similar values.

↳ set of similar datatype variables.

Syntax: int percentage [60];

for now, with
direct value array

-11-

→ 1. int array [5] = {1, 2, 3, 4, 5}
 | subscript

2. int array [5] =

2. int array [5] = {1, 2, 3, 4}.

3. int array [5];

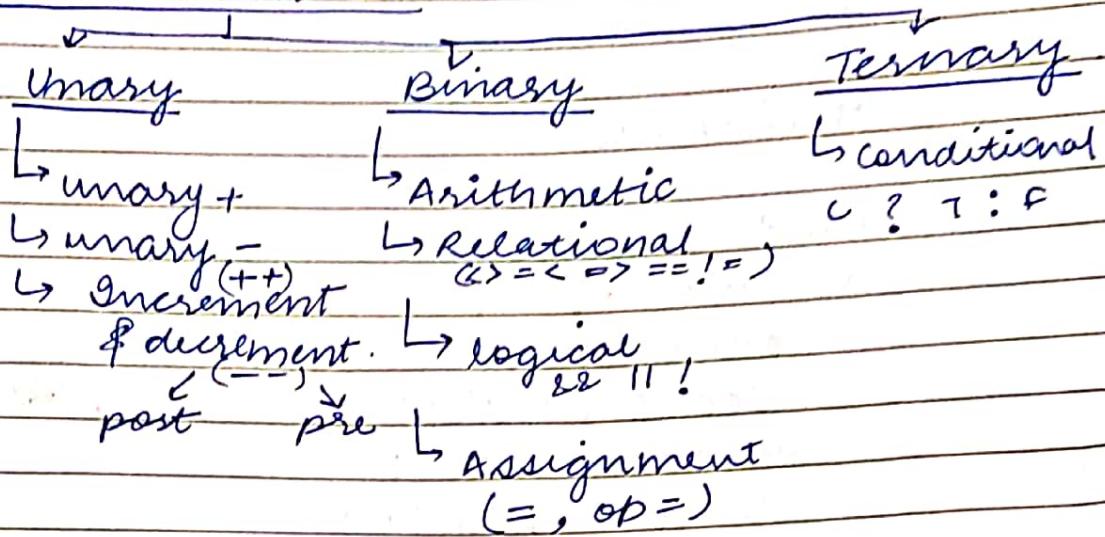
4. int array []; X

→ printf ("%d", array [2]);

~~Q.~~ Q. a, b a=5 b=10. O/P | a=10 |
 | ar[3]; |
 | ar[0]=5 |
(1) T=a
 | ar[1]=-10 |
 | ar[2]=ar[0] |
 | ar[0]=ar[1] |
 | ar[1]=ar[2] |
 | b=5 |

#

Operators in C



① Arithmetic + - * / % → only int

int, float, double, char

- Integer arithmetic
- Real " eq. $3.4 / 2.5 + 2.4 * 5$
- Mixed Mode - " eq. $3 / 2 + 2 * -1$
- Int arith - eg. $3 / 2 + 2 * 5 - 1$
- Real arith - eg. $3.4 / 2.5 + 2.5 * 5.0 - 1.0$
- Mixed Mode - eg. int i = 3 / 3.5
(Calculation in bigger DT)

* Q. Meaning Relational are inversion of each other.

Q. WAP to calculate months & days from no of days given by the user.

#include < stdio.h >

void main()

{ int n, d, m;

printf ("Enter any no: ");

scanf ("%d", &n);

m = n / 30;

d = n % 30;

printf ("no of months: %d", m);

printf ("no of days : %d", d);

y.

(Q)

Relational <, <=, >=, ==, !=, >

a < b b > a

→ Answer of every relational op is either 1 or 0.

 ↓ ↓
 T F

→ Also compare arithmetic op.

eg. $1+35 > 36+2$

$1+36 > 2+100$

Unary NOT !

inverts the value

! (a < b)

 T
 F = 0

(a >= b)

0.

a = 10

b = 20

* Q. Qf expr logical & relational.

- / -

(3)

logical

U& AND

II OR

! NOT.

(i) AND (&).

C1	&	C2	
T		T	1
T		F	0
F		T	0
F		F	0.

(ii) OR (||)

C1	C2	C1 C2
T	T	1
F	T	1
T	F	1
F	F	0.

(iii) NOT (!)

$$1 \leftrightarrow 0$$

$$0 \leftrightarrow 1.$$

if ($a < b$) & ($b < c$)

printf ("a is largest");

(4) Encapsulation & Decapsulation

a + +

b - -

→ Post Incr.

$$a = 10 \quad b = a \text{ (++)}$$

$$\textcircled{1} \quad b = a$$

$$\textcircled{2} \quad a \text{ (++)}$$

$$b = 10$$

$$a = 11.$$

Post Dec.

$$a = 10$$

$$b = a - -$$

$$\textcircled{1} \quad b = a$$

$$\textcircled{2} \quad \cancel{a} = a - -$$

$$b = 10$$

$$a \rightarrow 9.$$

→ Pre Incr.

$$b = ++a$$

$$b = 11 \quad a = 11$$

Pre Dec.

$$b = a$$

$$a = 9.$$

$$y = 10$$

$$x = y + + \quad + \quad y + + \quad ++y \\ = 10 + 11 \cancel{+} + 13 \\ = 34.$$

$$\boxed{y = 12}$$

Bitwise op.

\wedge , \vee , $\wedge\wedge$, $\wedge\wedge\wedge$, \ll , \gg
 AND OR XOR left shift right shift

B1	B2	$B1 \& B2$	$B1 B2$
0	0	0	1 0
0	1	0	1
1	0	0	1
1	1	1	1

(exclusive or)

$B1 \wedge B2$

odd ↓ even ↓ 0	0
	1
	1
	0

eg. shift this by this

$$a = \underline{\underline{0}}1001 \\ 10010$$

$$b = 00110010$$

left shift take $\rightarrow \times 2$

int $a = 5$

size of (a) = 32 bits

int $b = a \ll 2$

eg. $20 \gg 2$

10100

01010

00101

$\div 2$

Special op.

&, *, .,), →, size of.
 address of pointers comma. Know size of any DT.

Expression Evaluation

top priority * / %.
 low pri. + -

$$\begin{array}{l} 3/2 + 3 * 3 - 4 \\ \text{FP. } \begin{array}{l} \textcircled{1} 1 + 3 * 3 - 4 \\ \textcircled{2} 1 + 9 - 4 \end{array} \end{array}$$

$$\begin{array}{l} \text{LP } \begin{array}{l} \textcircled{1} 10 - 4 \\ \textcircled{2} 6. \end{array} \end{array}$$

Datatype conversions

→ implicit ($S \rightarrow B$)
 → explicit ("both")

int → float
 $\frac{3}{3} \rightarrow 3.0$
 ↑ loss of data

eg. int i, x; $x = e/i + i * f - d$
 long e;
 float f;
 double d;

long long → float
 long → float - double
 float - double
 ↓
 double.

exit with
 float
 unsigned long int
 long int
 unsigned int
 int
 short / char

Explicit conversionType Casting

(data type) variable name;

 $x = (\text{float}) a; a=3;$

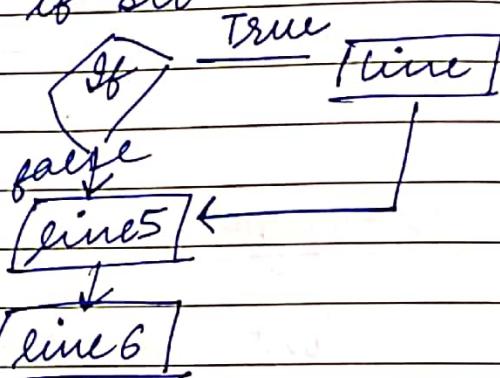
3.00

float
(int)t;
(double)f;

Decision Making & Branchingif

```
int main()
{
    S 1
    cond
    {
        2
        {
            3
            {
                4
            }
        }
    }
}
```

if stt



True
 if (cond)

else
 false
 {
 }
 }

if...else
 Q. Take a no from user & print -ve or +ve

```
int n;
printf ("Enter No.");
scanf ("%d", &n);
if (n < 0)
    printf ("Negative");
else
    printf ("positive");
return 0;
```

Q. quotient float a, b;

Print ("Enter 2 nos");

scanf ("%f %f", &a, &b);

~~if (if (c == 10))~~

~~n = a/b;~~

Print ("Quotient * . f")

Nested
if...else

with
calculator.

Q. find largest no. of the 3.

$$\text{det} \cdot b^2 - 4ac.$$

d>0 d<0 d=0.

$$d = b * b - 4 * a * c.$$

Q. WAP to take total marks from user
and display grade.

75-100 A 80-99 B+

60-74 B+ 45-59 B <44 C.

Q. Calculate the cost of consumption :

units

0-200

charges

Rs 0.50 / unit

201-400

Rs 100 + 0.65 / unit excess of 200

401-600

Rs 230 + 0.80 / unit -- of 400

601 & above

Rs 390 + Rs 1 / unit even
of 600.

C constructs

conditional / selection

Repetition

If... else ✓

loops

Switch ✓

Conditional op. ✓

Switch

switch (choice) → variable
 S int
 or
 char.

case 1:

break;

case 2:

break;

!

default:

3.

eg. switch (marks)

S

case 10:

case 9:

case 8:

printf ("Excellent");

break;

case 7:

case 6:

printf ("good");

break;

Q. WAP to take no from a user & display corresponding day of week:

switch (Monday)

int n;

scanf ("Enter no %d", &n);

switch (n)

case 1:

printf ("Monday");

break;

case 2: printf ("Tuesday"); break;

;

default: printf ("Enter again");

scanf ("%d", n);

Loops:

1. while:

while (i <= 10)

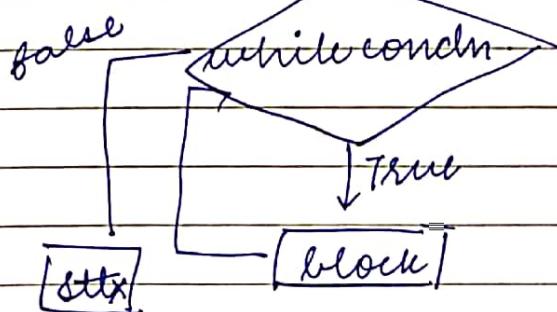
{

loop printf ("value %d", i);

i++;

y

start.



value
value²
value³

Q. do ... while

do

{

 block;

} while () ;

Q. WAP to display first N natural nos when n is entered by user.

printf ("Enter n:");

int n, i=1;

scanf ("%d", &n);

while (int i=+; i<=n; i++)

 printf "%d", i);
 i++

Q. for.

for (initiation; condition; inc./dec)

 block;

 y.

Q. Sum: $1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} \dots + \frac{1}{N}$

Nested loops $m \times n$

outer \rightarrow for (int i ; $i < n$; inc.)

inner \rightarrow for (int j ; $j < m$; inc.)
 {
 }
 y
 y

(2) while ($cond^n$)

while ($cond^n$)
 {
 }
 y.

any permutation
 com. is poss
 & upto whatever
 level

Q. 1

1 2 3

1 2 3 : n.

for (int $i=0$; $i < n$; $i++$)
 {

for (int $j=0$; $j < i$; $j++$)
 printf ("*");
 printf ("%d", j);
 printf ("\n");
 y

Q. Prime:-

for ($i=1$; $i < n$; $i++$)

Q. a - b.

{ if ($n \% i == 0$)

count++;

y

if (count > 1)

printf ("Not prime");

else printf ("Prime");

y

Loop control constts

break & continue

break

continue

for ()

{

if (condⁿ)

{

break;

↓ skip

↓

}

e.g. for (i = a; i < n; i++)

{

if (n % i == 0)

{ printf("Not prime");

break; }

}

e.g. for (i = 1; i <= n; i++)

{

if (i % 7 == 0)

printf(" %d ", i);

else

continue;

}

for ()

{

if ()

!

{ break; }

else

continue;

{

}

Defns
& eq.
of
relative
operators

goto label

goto label;

label:

e.g. label:

```
if( i <= 100)
    printf ("%d", i);
else break;
i++;
goto label;
```

go goto x;
|
|
→ x:
find as well as
backward

over ~~theoretical~~

