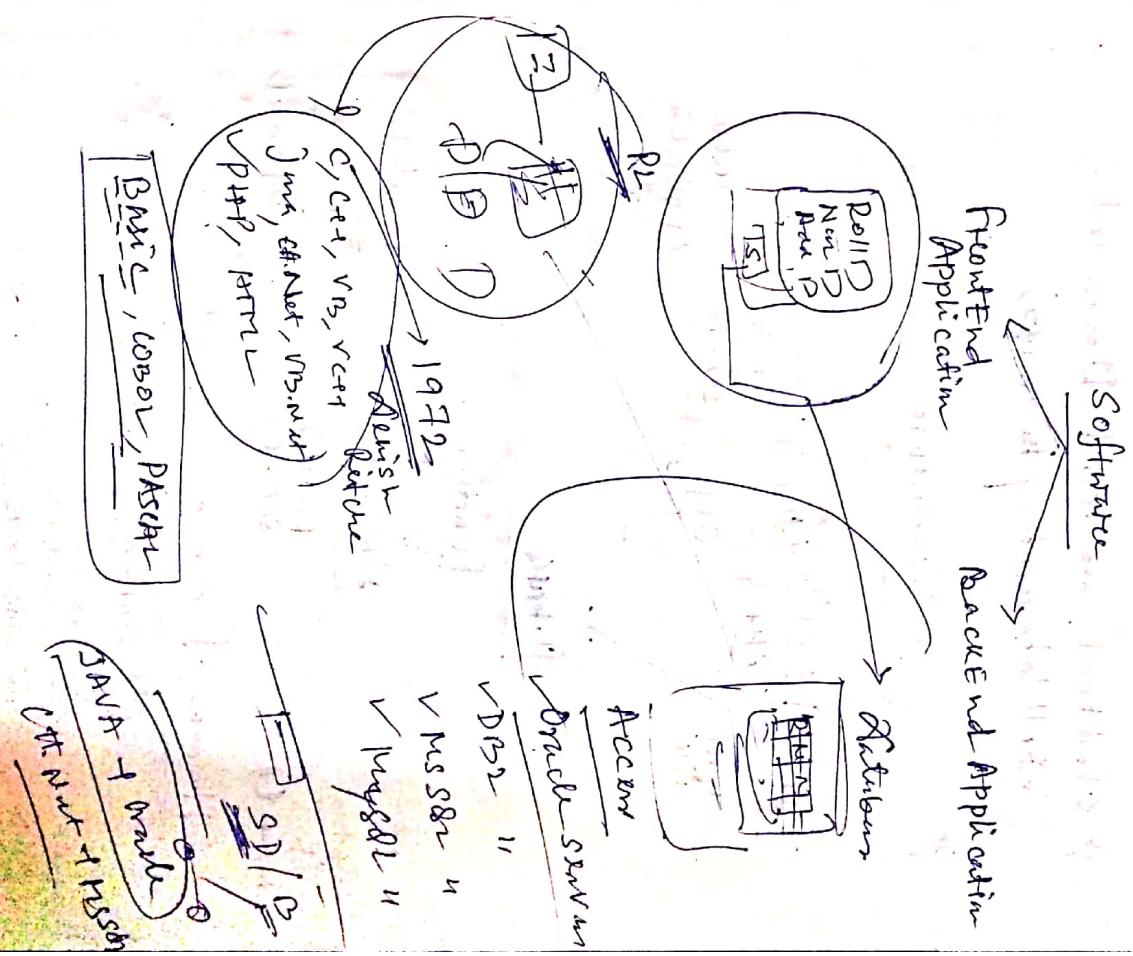


## Programming Language

- ✓ user language contain with set of instruction.
- ✓ we can build software.



# C

C is a low level programming language.

It is a console based application.

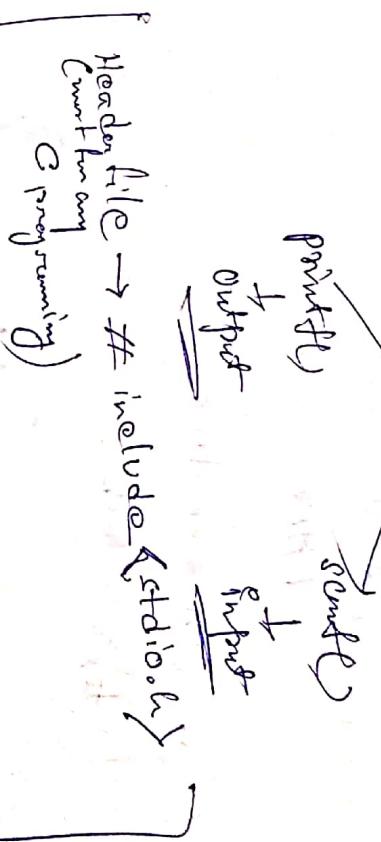
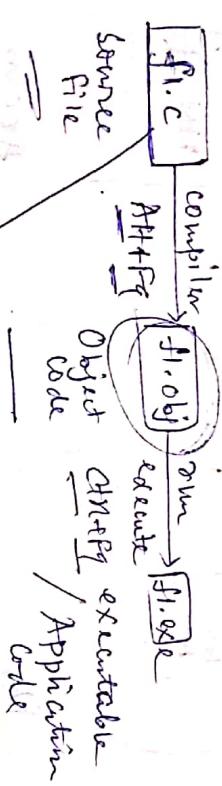
C is a function oriented, non procedural programming language.

## programming language

Created invented by Dennis Ritchie

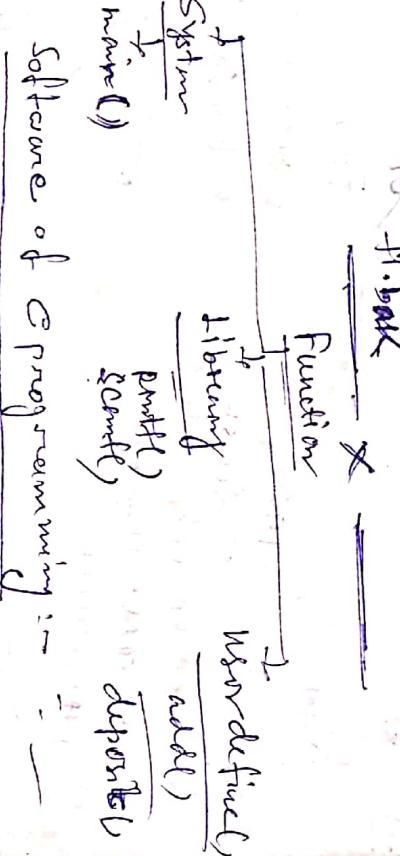
in Bell laboratory on 1972

## C Architecture



Library file / Header file

stdio.h (standard input / output Header file)



## Software of C programming :-

- (1) Turbo C
- (2) Min C
- (3) Borland C
- (4) Code ware

Alt-F9 → Compile  
j → Complete line in C

Ctrl+F5 → Run

Name : —  
Add : —

PHW : —

①

variable  
variable  
Name  
Name  
datatype

standard datatype  
(primitive)

- ① int → 2 byte → %d
- ② float → 4 byte → %f
- ③ char → 1 byte → %c

String

Collection of  
characters

void main()  
{  
 char c;  
 printf("%c",

201	for	for
-	=	=
-	-	-
-	-	-

int w, x, y, z → w = 10  
float a, b → a = 10.5  
char g → g = 'A'

operator

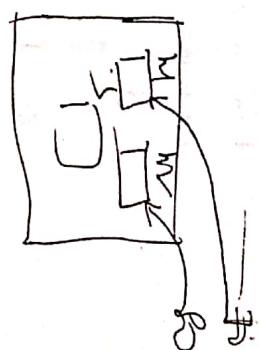
s = n + n

Proximity  
variable declaration after else if()

Comment → // adddition Comment.

addition of two predefine nos.

o space is apparent  
operator & variable  
none



```
// Comment
#include <stdio.h>
#include <conio.h>

void main()
{
    int n1, n2, s;
    clrscr();
    n1 = 60;
    n2 = 70;
    s = n1 + n2;
    printf("The sum of %d and %d\n"
           "is %d", n1, n2, s);
    getch();
}

int main()
{
    int n1, n2, s;
    clrscr();
    printf("Enter the 1st no ");
    scanf("%d", &n1);
    printf("Enter the 2nd no ");
    scanf("%d", &n2);
    s = n1 + n2;
    printf("The sum of %d and %d is %d", n1, n2, s);
    getch();
}
```

bseal  $\rightarrow$  10000

comme  $\rightarrow$  zero

pf  $\rightarrow$  8m

gesd = bseal + comme - pf

#include <stdio.h>

#include <conio.h>

void main()

}

int m1, m2, m3;

float avg;

clrscr();

printf("Your Bengali : ");

scanf("%d", &m1);

printf("Your English : ");

scanf("%d", &m2);

printf("Your Maths : ");

scanf("%d", &m3);

avg = (float)m1 + m2 + m3 / 3;

m1 + m2 + m3 / 3

float avg  
avg = (float)m1 + m2 + m3 / 3;

printf("Your total marks : %f", avg);

$$\begin{aligned} \text{Percentage} &\rightarrow \\ \bar{P} &= \frac{m_1 + m_2 + m_3}{3} \times 100 \\ m_1 &= \dots \\ m_2 &= \dots \\ m_3 &= \dots \\ m_4 &= \dots \\ m_5 &= \dots \\ m_6 &= \dots \\ S &= m_1 + m_2 + m_3 + m_4 + m_5 + m_6 \end{aligned}$$

$$\begin{aligned} \text{int } m_1, m_2, m_3, S \\ S = m_1 + m_2 + m_3 \end{aligned}$$

float avg  
avg = (m1 + m2 + m3) / 3

return 0;

int m1, m2, m3, S  
S = m1 + m2 + m3  
avg = (m1 + m2 + m3) / 3  
return 0;

5

getch();

Programming Logic

## Technique

### Constructor

Constructur

17 Checking

if, else-if, else

2) looping  
a) while b) do-while c) for

3) Branching

$(a = \gamma u) \neq 0$

(au) < 100

Maximum Number

```
#include <stdio.h>
#include <conio.h>
int main()
{
    clrscr();
    printf("Enter the 1st no. ");
    scanf("%d", &n1);
    printf("Enter the 2nd no. ");
    scanf("%d", &n2);
    if (n1 > n2)
```

Print d - C<sup>o</sup>'l. d' is the minimum "m")



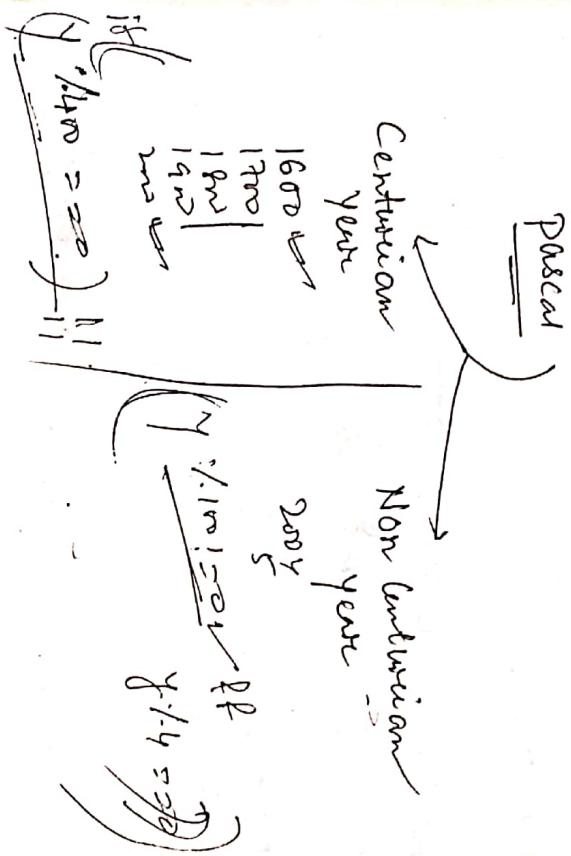
Year → leap year

2004 →  
→ 8  
12

1600 ✓  
1700 ✓  
1800 ✓  
1900 ✓  
2000 ✓

#include < stdio.h>  
#include <conio.h>  
void main();  
{  
 int y;

clrscr();  
{  
 printf("Enter year: ");  
 scanf("%d", &y);  
 {  
 if ((C4y % 400 == 0) || (y % 100 == 0 && y % 4 == 0))  
 {  
 printf("The year %d is leapyear.", y);  
 }  
 else  
 {  
 printf("The year %d is not leapyear.", y);  
 }  
 }  
}



### Logical operator

- 1) Logical AND →  $\wedge$
- 2) " OR →  $\vee$
- 3) " NOT →  $\neg$

else if

$>= 75 \rightarrow A$

$>= 60 \rightarrow B$

$>= 40 \rightarrow C$

$< 40 \rightarrow D$

$m > 50$

$m < 50$

$m > 40$

$m < 40$

$m > 20$

$m < 20$

$m > 10$

$m < 10$

$m > 0$

$m < 0$

$m > -10$

$m < -10$

$m > -20$

$m < -20$

$m > -30$

$m < -30$

$m > -40$

$m < -40$

void main()

Grade of marks

{  
int m;

printf("Enter your marks : ");

scanf("%d", &m);

if (m>=50)

else if (m>=40)

printf("Your marks are → A");

else if (m>=20)

printf("Your marks are → B");

else if (m>=10)

printf("Your marks are → C");

else if (m>=0)

printf("Your marks are → D");

else if (m<=40)

printf("Your marks are → E");

else if (m<=20)

printf("Your marks are → F");

else if (m<=10)

printf("Your marks are → G");

else if (m<=0)

printf("Your marks are → H");

三  
卷之三

$$P_f = 100 \text{ mbar} \quad \text{at } 12^{\circ}\text{C}$$

—  $\sin \theta = \frac{1}{\sqrt{2}}$

— 50

Geal → 12000 - pf

ପାଇଁ କିମ୍ବା ୧୦୦୦ ଟଙ୍କା ହେଲାମୁଣ୍ଡିଲା

bard = 2

(narrator):

$$P_f = \frac{1}{m}$$

```
else if(sod == snow)
```

१२८३

1  
leg

Pf 2500

Grad 2 b son - pf

—

```
Printf("%c", S.charAt(0));
```

$$f_n = \cos(n)$$

۳

else

Scanned by CamScanner

# Data Type

Type	Length	Range
unsigned char	8 bits	0 to $\rightarrow 255$
char	8 bits	-128 to $\rightarrow 127$
enum	16 bits	-32,768 to $\rightarrow 32,767$
unsigned int	16 bits	0 to 65,535
short int	16 bits	-32,768 to 32,767
int	32 bits	-32,768 to 32,767
unsigned long	32 bits	0 to 4,294,967,295
long	64 bits	$\pm 2^{63} - 1$
float	32 bits	$3.4 \times (10^{-38} - 3.4)$
double	64 bits	$1.7 \times (10^{-308} - 1.7)$
long double	128 bits	$(1.7 \times 10^{-308})$

Nested if  
if inside if is called Nested if.

if ( $my=30$ )

{ if ( $my=80$ )

{ p — letter

else

— Normal printed

}

else

if ( $my=25$ )

p — passed by construction

else

p — failed

3.0e+0 (30000.4932  
to 1.1e+0 (10000.4932)

y  
x  
h1, h2, h3

Largest between three numbers

int a1, a2, a3;

else

printf("%d",

scanf("%d",

"%d",

"%d",

"%d",

"%d",

if (a1 >

else

"%d",

else

printf("%d",

"%d",

Printf("In the numbers  
%d and %d  
%d is largest)

WTF → gives set of code executing many times

NTABLBE.C (with delay function)

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

```
#include <conio.h>
```

```
#include <dos.h>
```

```
void main()
```

```
{
```

```
int s, n, i;
```

```
Closeon();
```

```
printf("Enter the start number : ");
```

```
scanf("%d", &n);
```

```
int =
```

```
i = j;
```

```
while (i <= 10)
```

```
{
```

```
delay(1000);
```

```
s = nx;
```

```
printf("%d\n", s);
```

```
i = i + 1;
```

```
getch();
```

```
}
```

1

2

3

4

5

6

7

8

9

10

11

12

13

## Application of Do While Loop

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int n;
    clrscr();
    do
    {
        cout << "Enter the number : ";
        fflush(stdin);
        cin > n;
        cout << endl;
        cout << "The square value of ";
        cout << n * n;
        cout << endl;
        cout << "Press q to continue ";
        cout << endl;
        if (ch == 'q')
            break;
        else
            getch();
    } while (ch != 'q');
}
```

for (i=1; i<=10; i++)

}  
Initial Condition  
Conditional increment / decrement  
Expression / decrement Expression

s = 0  
for (n=1; n<=10; n++)  
s = s + n;

s = 0 + 1 + 2 + 3 + ... + 10 = 55

i = 1

for (i=10; i>=1)

i = 10

a / 10  
b = 1  
20 / 10

for (i=10; i>=1)

i = 10

a / 10  
b = 1  
20 / 10

(11)

void main (C),

2

int n, s;  
clrscr();  
s = 0;  
for (n=1, n<=10, n++);  
{  
    s = s + n;

Sum of 1 to 10 = s. (for loop)  
print("The sum of 1 to 10 is %d", s);  
getch();

5.

Somewhere,  $\rightarrow 10$

$$i = \underbrace{1 \times 2 \times 3 \times 4 \times 5 \times 6}_{\text{Summanden}} \quad \text{and result}$$

```
int n, s;
class C {
    s = 0;
    sum(n) {
        if (n == 10) n++;
        if (n >= 10) return s;
        s += n;
        sum(n);
    }
}
long int fact() {
    long int n, f;
    class C c;
    cout << "Enter number: ";
    cin >> n;
    f = fact(n);
    cout << "The factorial value is " << f;
    return f;
}
```

5.

```
cout << "The sum of all even numbers from 1 to 10 is " << sum();
getch();
```

# Number is prime or not

Prime No

```
void main()
{
```

```
    n = 7
    flag = 0
    i = 2
    if (n > 1)
        for (i = 2; i <= sqrt(n); i++)
            if (n % i == 0)
                flag = 1
                break;
        if (flag == 0)
            cout << "The number is prime";
        else
            cout << "The number is not prime";
}
```

```
int n, i, flag;
if (flag == 0)
    cout << "The number is prime";
else
    cout << "The number is not prime";
}
```

```
cout << "Enter the number";
scanf ("%d", &n);
if (n == 1)
    cout << "The number is not prime";
else
    cout << "The number is prime";
}
```

```
if (flag == 0)
    cout << "The number is prime";
else
    cout << "The number is not prime";
}
```

```
if (n == 1)
    cout << "The number is not prime";
else
    cout << "The number is prime";
}
```

```
if (flag == 1)
    break;
}
```

```
for (i = 2; i <= n - 1; i++)
    if (n % i == 0)
        flag = 1
        break;
}
```

```
if (flag == 0)
    cout << "The number is prime";
else
    cout << "The number is not prime";
}
```

```
if (flag == 0)
    cout << "The number is prime";
else
    cout << "The number is not prime";
}
```

The — add value  
to

display first 20 fibonaci  
series.

5  
getchar();

void main()

{

Fibonacci Series

$$\begin{array}{r} 0 \\ 1 \\ 1 \\ 2 \\ 3 \\ 5 \\ 8 \\ 13 \\ 21 \\ 34 \\ 55 \\ \hline 89 \\ 144 \end{array}$$

a = 0  
b = 1

a = 0  
b = 1

clrscr();

printf ("The Fibonacci series is:");

c = 0;  
c = 1;  
c = 0;

for (c = 0; c <= 100)  
    c = a + b  
    printf ("%d", c);  
    a = b;  
    b = c;

C = a + b;  
printf ("%d", C);  
a = b;  
b = c;

3

## Reverse of a Number

void main()

## Reversing

$$n = 153 \quad m = 351 \quad mn = 0$$

$$10 \left| \begin{matrix} 1 & 5 & 3 \\ 1 & 5 & 3 \end{matrix} \right|^{15} \\ = 0 + 3 = 3$$

$$\frac{15}{3} \longrightarrow m_2.$$

$$10 \left| \begin{matrix} 1 & 5 \\ 1 & 0 \end{matrix} \right|^{15} \\ = 10 + 5 = 15$$

$$m_1$$

$$mn = 10 \times 15 + 3 \\ = 30 + 3 = 33$$

$$10 \left| \begin{matrix} 1 & 0 \\ 1 & 5 \end{matrix} \right|^{15} \\ = 10 + 5 = 15$$

$$m_0$$

n, m, mn

for n > 0  
while (n > 0)

$$mn = 10 \times m + n$$

$$m = mn / 10$$

}

n = n / 10

}

n = n / 10

}

$$c = d * d \\ - ① \\ s = s + c \\ 0 + 121 \\ 121 + 121 \\ 242 + 242 \\ 484$$

484

$$80$$

$$35$$

35

printf("The opposite of %d is %d.\n", n, mn);

getchar();

$$mn = d * d / 10;$$

$$mn = 4 * 4 / 10;$$

while (n > 0)

Printf ("Enter the number : ");

scanf ("%d", &n);

if (n == 0)

int n, mn, mn;

else

else

else

else

else

## Reversing

145 → Krishnamurthy No

void main()

{

int n, m, s, r, d;

s = 0;

cout <<

to → no enter →  
sent → Rn  
estimate.

s = 0;

t = n;

while (t > 0)

{

d = n / 10;

c = d \* d \* d;

s = s + c;

} if (s == n)

{

cout << " - " <<

)

s = s + f / 10;

} if (s == n)

2

cout << "

)

1  
1 + 24 + 120 = 145

void main()

{

int n, d, s, t, i;

s = 0;

clrscr();

printf ("Enter →

Scandf ("%.d" &n);

t = n;

while (t > 0)

{

d = t % 10;

f = 1;

for (i = 1; i <= d; i++)

{

j = f \* i;

f = f \* i;

s = s + f;

t = t / 10;

} if (s == n)

Print ("The number is  
Kin") ;

କୃତ୍ସମାନରେ ଏହାରେ ଅଧିକ ଦୂର

```
int main()
{
    int m, s, t;
    cout << "Enter minutes : ";
    cin >> m;
    cout << "Enter seconds : ";
    cin >> s;
    t = m * 60 + s;
    cout << "Total time in seconds is : " << t;
}
```

```
    }  
    j++;  
    if(j < n){  
        cout << " " << arr[j];  
    }  
}
```

$\{ \text{if } C \leq n \}$

32  
31 (51, P1/1) standard

## Pattern Print

*	*	*	*	*
*	*	*	*	*
*	*	*	*	*
*	*	*	*	*
*	*	*	*	*

```
for(r=1; r<=5; r++)
    for(c=1; c<=5; c++)
        cout << "*";
```

```
cout << endl;
```

```
for(r=1; r<=5; r++)
    for(c=5; c>r; c--)
        cout << "*";
```

```
cout << endl;
```

```
for(r=1; r<=5; r++)
    for(c=1; c<=r; c++)
        cout << "*";
```

```
for(r=1; r<=5; r++)
    for(c=1; c<=5-r; c++)
        cout << " ";
```

```
void main()
```

```
{
```

```
int c, r;
```

```
clrscr();
```

```
for(c=1; c<=5; c++)
    
```

```
for (r=1; r<=5; r++)
    
```

```
    
```

```
    cout << "*";
```

```
}
```

```
    cout << endl;
```

```
}

void main()
{
    int c, r;
    clrscr();
    for(c=1; c<=5; c++)
        for (r=1; r<=c; r++)
            cout << "*";
    cout << endl;
}
```

printf("%d", n);

getch();

2)

void main  
{

int c, m, n;  
clrscr();

for (m = 1; m <= 5; m++)

① 1  
1 2  
1 2 3  
1 2 3 4  
1 2 3 4 5  
② 1  
1 2  
3 3 3  
4 4 4 4  
5 5 5 5 5  
③ 1  
2 3  
4 5 6  
7 8 9 10  
11 12 13 14 15

printf("%d", n);

④

A  
A-B  
A-B-C  
A-B-C-D  
A-B-C-D-E

wildcard

{

int c, m, n;

clrscr();  
for (c = 1; c <= m; c++)

{

printf("%d", c);

{

⑤

{

wildcard();

3

Printf("%d", c);  
getch();

2

Printf("%d", c);  
getch();

1

int c, m, n;  
clrscr();  
for (c = 1; c <= m; c++)  
{

for (c = 1; c <= n; c++)

{

printf("%d", dis);

dist;

'  
printf("\n");

getch();

3

Ass

American Standard Code for

Information Interchange



ASCII code

0

0 → dot  
1 → dotdot

2 → space

3

4

5

6

7

8

9

10

11

void main()

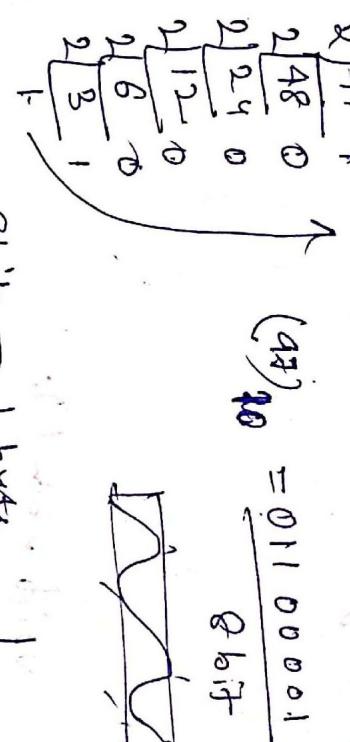
0

i

j

else{  
for(i=0; i<255; i++)

{  
printf("%d-%d\n", i, i)



for(j=0; j<255; j++)  
{  
 p → ("%d.%d.%d.%d") → 65. — A  
 getch();  
 ascinvalue →





`for (c=0; c < n; c++)`

2

printf(" %c", arr[i])

3

`printf("\n");`

4

getchar();

5

switch-case



switch(d)

case 1:

break; ("Study")

case 2:

break; ("C")

- 3:

default: ("Final")

Menu Driven

M, N, R, C, ch

P ("Print") for Addition

3

4

5

char ch

switch(ch)

case 1:

break;

case 2:

case 3:

case 4:

case 5:

case 6:

case 7:

case 8:

case 9:

case 10:

case 11:

case 12:

case 13:

case 14:

case 15:

case 16:

case 17:

case 18:

case 19:

case 20:

case 21:

case 22:

case 23:

case 24:

case 25:

case 26:

case 27:

case 28:

case 29:

case 30:

case 31:

case 32:

case 33:

case 34:

case 35:

case 36:

case 37:

case 38:

case 39:

case 40:

case 41:

case 42:

case 43:

case 44:

case 45:

case 46:

case 47:

case 48:

case 49:

case 50:

case 51:

case 52:

case 53:

case 54:

case 55:

case 56:

case 57:

case 58:

case 59:

case 60:

case 61:

case 62:

case 63:

case 64:

case 65:

case 66:

case 67:

case 68:

case 69:

case 70:

case 71:

case 72:

case 73:

case 74:

case 75:

case 76:

case 77:

case 78:

case 79:

case 80:

case 81:

case 82:

case 83:

case 84:

case 85:

case 86:

case 87:

case 88:

case 89:

case 90:

case 91:

case 92:

case 93:

case 94:

case 95:

case 96:

case 97:

case 98:

case 99:

case 100:

case 101:

case 102:

case 103:

case 104:

case 105:

case 106:

case 107:

case 108:

case 109:

case 110:

case 111:

case 112:

case 113:

case 114:

case 115:

case 116:

case 117:

case 118:

case 119:

case 120:

case 121:

case 122:

case 123:

case 124:

case 125:

case 126:

case 127:

case 128:

case 129:

case 130:

case 131:

case 132:

case 133:

case 134:

case 135:

case 136:

case 137:

case 138:

case 139:

case 140:

case 141:

case 142:

case 143:

case 144:

case 145:

case 146:

case 147:

case 148:

case 149:

case 150:

case 151:

case 152:

case 153:

case 154:

case 155:

case 156:

case 157:

case 158:

case 159:

case 160:

case 161:

case 162:

case 163:

case 164:

case 165:

case 166:

case 167:

case 168:

case 169:

case 170:

case 171:

case 172:

case 173:

case 174:

case 175:

case 176:

case 177:

case 178:

case 179:

case 180:

case 181:

case 182:

case 183:

case 184:

case 185:

case 186:

case 187:

case 188:

case 189:

case 190:

case 191:

case 192:

case 193:

case 194:

case 195:

case 196:

case 197:

case 198:

case 199:

case 200:

case 201:

case 202:

case 203:

case 204:

case 205:

case 206:

case 207:

case 208:

case 209:

case 210:

case 211:

case 212:

case 213:

case 214:

case 215:

case 216:

case 217:

case 218:

case 219:

case 220:

case 221:

case 222:

case 223:

case 224:

case 225:

case 226:

case 227:

case 228:

case 229:

case 230:

case 231:

case 232:

case 233:

case 234:

case 235:

case 236:

case 237:

case 238:

case 239:

case 240:

case 241:

case 242:

case 243:

case 244:

printf("Enter your first number");

scanf("%d", &n1);

printf("Enter your second number");

scanf("%d", &n2);

if (n1 < n2)

{

Case 1:

printf("Addition = ");

printf("Subtraction = ");

printf("Multiplication = ");

printf("Division = ");

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

switch(a)

{

Case 1:

n = n1 + n2;

printf("-----");

break;

Case 2:

n = n1 - n2;

printf("-----");

Case 3:

n = n1 \* n2;

printf("-----");

Case 4:

n = n1 / n2;

printf("-----");

getch();

exit(0);

default :

printf("It's a wrong input");

3  
getch();

3

2

while (1) → infinity loop

#include <iostream.h> → exit.

header file.

$$th = \pi$$

$$pr = g$$



220V  
240V  
240V

More like grade 2 way  $\rightarrow$  Taken

Case 1: print('Takun'  $\rightarrow$  2nd)  
percent;

Case 2: print('Ling van Ies...is, am')

int "Ies, am"

float & any:

classen();

print('Takun works');

sent (  $\rightarrow$  th )

print('a certain value');

send (  $\rightarrow$  mth );

while ( 1 )

{

classen();

percent(

print(  $\rightarrow$  total );

print(  $\rightarrow$  total2 );

print(  $\rightarrow$  avg );

print(  $\rightarrow$  grade );

}

john( );

def float(

print('Takun');

1.8 = M + Pn;  
M = (plant) S/Pj  
S = switch(n);  
Pj = 1000

1.8 = M + Pn;

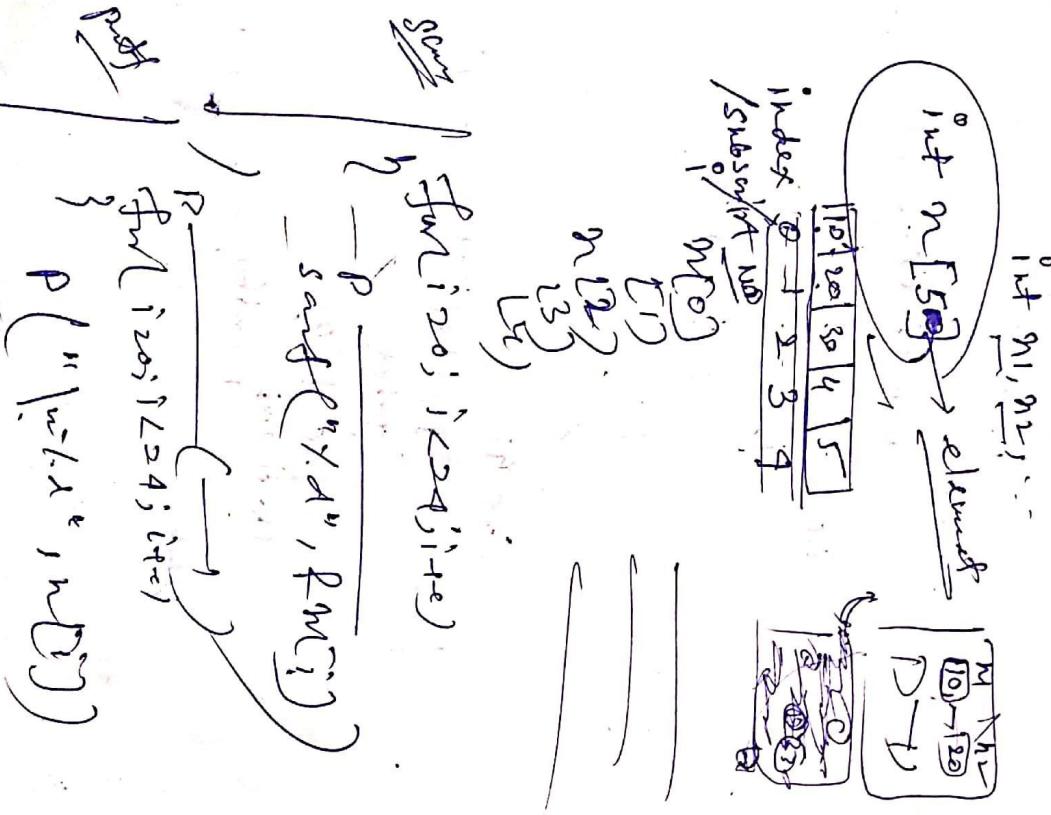
M = (plant) S/Pj

S = switch(n);

Pj = 1000

## Array

- ✓ It is a collection of elements with a single variable name but having similar datatype.
- ✓ It stores data in a contiguous format.



Number is stored in min

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

```
void main();
```

```
int n[5], i; max, min,
```

```
clrscr();
```

```
for (i = 0; i < 5; i++)
  {
    /* code */
  }
}
```

printf ("Enter numbers")

scanf ("%d", &n[i]);

max = min = n[0];

for (i = 1; i < 5; i++)
 {
 /\* code \*/
 }
}

if (n[i] > max)
 max = n[i];

if (n[i] < min)
 min = n[i];

scanf

fun(i, 20);

- sang("A", fm[i])

fun(i, 24); fm[i]

fun(i, 24); fm[i]

put

fm[i] = fm[i] + fm[i]

Printf ("num = %d",  
prime & num - 1),  
getch();

Algorithms no one  
C program

5	71	68	7	82
0	1	2	3	4

sn ↪ ≠

reid main()

int n[s], d; odd, even;

odd = 0;

even = 0;

even();

else even();

for (i=0; i < n; i++)

if (n[i] % 2 == 0)

    odd++;

    even++;

    scanf("%d", &n[i]);

    if (n[i] % 2 != 0)

        odd++;

        even++;

    printf("%d", odd);

    printf("\n");

    if (odd == 0)

        printf("No even numbers\n");

    else if (odd == n)

        printf("All numbers are odd\n");

    else printf("Even numbers are %d", even);

    else printf("Odd numbers are %d", odd);

    else printf("Even numbers are %d and odd numbers are %d", even, odd);

    else printf("Even numbers are %d and odd numbers are %d", odd, even);

    else printf("Even numbers are %d and odd numbers are %d", even, odd);

    else printf("Even numbers are %d and odd numbers are %d", odd, even);

    else printf("Even numbers are %d and odd numbers are %d", even, odd);

No is available in inputed  
no

# include <stdio.h>

# include <conio.h>

void main()

int n[s], d, i; f;

f = 0;

clrscr();

for (i=0; i < n; i++)

    if (n[i] % 2 == 0)

        f++;

    else

        f++;

    printf("\nEnter no : ");

    scanf("%d", &n[i]);

    if (f == 0)

        printf("No even no are odd\n");

    else if (f == n)

        printf("All numbers are odd\n");

    else if (f == 1)

        printf("One even number is present\n");

    else printf("Even numbers are %d", f);

    else printf("Odd numbers are %d", f);

Sum of 2 nos in array

$$f(g) = 1$$

R. Prentiss ("Availabili")

int min([s], max([s]), n) {

Clemens 11

for  $C_i = 0$ ,  $1 \leq i \leq n-1$

Printed ("Not available")

gehen).

“Enter —————— ” ( “Q u i c k , D i d ” ) .

for (i=0; i<n; i++)

Point f ("Euler")

rain (the sun shines) is

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

$n_3[i] = n_1[i] + n_2[i]$ ,

141

<u>7</u>	<u>5</u>	<u>80</u>	<u>7</u>	<u>50</u>
<u>0</u>	<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>
<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>	<u>5</u>
<u>6</u>	<u>7</u>	<u>8</u>	<u>9</u>	<u>0</u>
<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>	<u>5</u>

1

3

John Paul

Answer  $\Rightarrow$  which one is bigger

$\text{for}(i=0; i < 4; i++)$

$\text{int } n[5]; \text{int } s[5];$   
elements:

$\text{for}(i=0; i < 4; i++)$

$\text{print}(n[i] \rightarrow)$   
 $\text{scanf}()$

$\text{for}(i=0; i < 4; i++)$

$\text{print}(c[i] \rightarrow)$

$\text{scanf}()$

$\text{for}(i=0; i < 4; i++)$

$\text{for}(i=0; i < 4; i++)$

$\text{for}(i=0; i < 4; i++)$

$\text{else}$   
 $n[i] = m[2]$

$\text{printf("y.d.", n[2])};$

$\text{else}$

Count				
for( $r=0; r < 2; r++$ ) $n[3][r]$				
for( $c=0; c < 4; c++$ ) $n[3][c]$				
$r=0$	$c=0$	1	2	3
$r=1$	$c=0$	1	2	3

$\text{for}(r=0; r < 2; r++)$   
 $\text{for}(c=0; c < 4; c++)$

$P \rightarrow$   
 $\text{("y.d.", } \& n[3][c])$

$\text{G.D}$   
 $\text{G.D}$

$\text{display}$

Character is Digger

for (n=0; n<=2; n++)

cout << arr[1];

int arr[5] = {1, 2, 3, 4, 5}, i;

for (i=0; i<5; i++)

for (n=0; n<=2; n++)

for (c=0; c<=2; c++)  
print (c)

seen (%d)\n"

for (n=0; n<=2; n++)

m[i] = m[3+i]

for (c=0; c<=2; c++)

if (m[n] <= 100)

arr[i] = arr[3+i]

arr[3+i] = arr[i]

char name[20];  
cout << "Enter Name" << endl;  
cin >> name;

"\0" → (\0)  
String terminator  
space → \0  
→ Enter

char g[5]

if

name

for (i=0; i<4; i++)

g[i] = name[i]

cout << g;

String → Collection  
of characters

String

for

cout

3

g[i];

cout

endl;

cout

endl;

cout

endl;

3

3

3

void main () {

}

char name[20];

class();

printf("Enter name");

ffscanf("%s", name);

C scanf("%s", name);

gets(name);

printf("Your name is %s",

name);

getch();

5



for(i=0; i<n) { }

↓  
Initial Conditional  
Expression Expression.

{  
Expression Expression  
}

Sum of 1 to n

int loop

#include < stdio.h>  
#include < conio.h>

void main()

}  
for(i=10; i>0) {  
 a = 10;  
 b = 10;  
 a = a - 1;  
 b = b + 1;  
 cout << a << b << endl;  
}

a = 10;  
b = 10;  
a = a - 1;  
b = b + 1;

cout << a << b << endl;

int main()  
{  
 int i;  
 for(i=1; i<=n; i++)  
 cout << i << endl;  
}

a = 10;  
b = 10;  
a = a - 1;  
b = b + 1;

cout << a << b << endl;

getch();

s = 0;  
for(n=1; n<=10; n++)  
{  
 s = s+n;

}  
cout << s;

## Factorial at Anywhere

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

```
long int n, f
```

```
clrscr();
```

```
printf("Enter NO ");
```

```
scanf("%d", &n);
```

```
for (i=1; i<=n; i++)
{}
```

```
f = f * i;
```

$\frac{n!}{\text{Prime No}}$

```
for (i=2; i<=n-1; i++)
{}
```

```
if (n % i == 0)
```

```
{ flag = 1
```

~~break;~~

```
} if (flag == 0)
```

```
{ printf(" "); }
```

```
} else
{ for (m=1; m<=10; m++)
{}
```

```
if (n % 2 == 0)
```

```
{ s = s + m;
```

```
}
```

Print ("The sum of even no in n is %d");

3

Prime No

→ No one prime or not

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

```
void main()
```

```
{
```

```
int n, flag;
```

```
flag = 0;
```

```
clrscr();
```

```
printf("Enter NO");
```

```
scanf("%d", &n);
```

```
if (n == 1)
```

```
{
```

```
flag = 1
```

```
for (i = 2; i <= n - 1; i++)
```

```
{
```

```
if (n % i == 0)
```

```
{
```

```
flag = 1
```

```
break;
```

```
}
```

```
if (flag == 0)
```

```
{
```

```
printf("The number is prime");
```

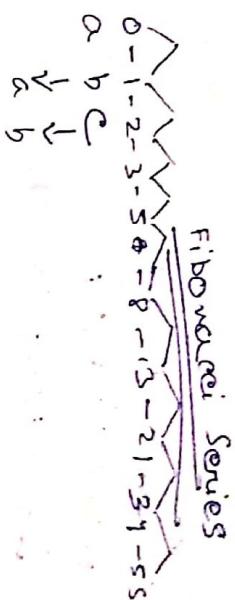
```
} else
```

```
{
```

```
printf("The number is not
```

```
prime");
```

```
getch();
```



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

```
void main()
```

```
{
```

```
C = A + B;
```

```
A = B;
```

```
B = C;
```

```
for (i = 0; i < 10; i++)
```

```
{
```

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

```
if (i < 9)
```

```
{
```

```
printf("\n");
```

```
}
```

```
else
```

```
{
```

```
printf("\n");
```

```
}
```

```
printf("The Fibonacci series is ");
```

```
getch();
```

point (", "));  
print (" ", " );  
cout << numbers << endl;

Output:  
1  
2  
3  
4  
5  
6  
7  
8  
9

i

c++ arr  
point (", " );  
cout << b << endl;  
b = c;

Output:  
1  
2  
3  
4  
5  
6  
7  
8  
9

j

Reverse Of a Number .

n = 153 , rev = 0

10 153  
53
15  
n = rev \* 10 + r  
= 0 \* 10 + 3  
= 0 + 3 = 3

①

char number  
name  
int i

i = strlen (num) ;  
rev = 0 ;  
for (i = i - 1 ; i >= 0 ; i--) {

②  
i > char num[i] ;  
if (num[i] < '0' || num[i] > '9') {  
cout << "Error" << endl ;  
exit (1) ;

ii> char str[ ] ;

str  
strcpy (str, str1)

③

123456789  
str  
char file[ ] ;  
fopen (file, "w") ;  
fwrite (str, sizeof (str), 1, file) ;

123456789  
str  
char file[ ] ;  
fopen (file, "w") ;  
fwrite (str, sizeof (str), 1, file) ;

char buffer  
int i = 0 ;  
for (i = 0 ; i < strlen (str) ; i++) {  
buffer[i] = str[i] ;  
}

char str[ ] ;  
strcpy (str, buffer) ;

char str1[ ] ;  
strcpy (str1, str) ;  
cout << str1 << endl ;

char buffer

int i = 0 ;  
(a = b)

123456789

stump

ANSI "ASCII" = 7bit = 128

if( $\text{strcmp}(\text{str}_1, \text{str}_2) == 0$ )

```
for(i:=0; nome[i] = '0'; i++)
```

```
def __init__(self, name, age):  
    self.name = name  
    self.age = age
```

at  $\rho = 5 \text{ cm}^3/\text{g}$

```
for(i=0; name[i]!='\0'; i++)
```

67 mes45

char n[20], m[20];

10

卷之三

10

void work();

No of characters

```

char n[20];
int i, num; name[20];
class();
cout << "Enter the characters of your name:- ";
for (i=0; n[i] != '\0'; i++)
{
    if (n[i] >='A' & n[i] <='Z')
        num += n[i] - 'A';
    else if (n[i] >='a' & n[i] <='z')
        num += n[i] - 'a';
}
cout << "The sum of all characters is :- " << num;

```

مکانیزم این دستورات را در مقاله (۲۰۰۱) مذکور شده است.

3  
punkt ( „ — „ ) :  
jetzt

≡ chare name [20]

1. S. name

(Page 10)

i      SH      Concentration Table

6	1	2	3	4	5	6
SH	K	SH	K	SH	K	SH
0	1	2	3	4	5	6

for (i=0; i<6; i++)  
    for (j=0; j<6; j++)

        f(i,j) = 0;

    g[i][j] = f[i][j];

    k = 1;

    k++;

    f[i][j] = k;      k++;

    g[i][j] = f[i][j];

    for (j=0; j<6; j++)

        g[i][j] = f[i][j];

    for (i=0; i<6; i++)

        int k = 1;

        k++;

        f[i][j] = k;

        g[i][j] = f[i][j];

    }

}

for (i=0; i<6; i++)

    for (j=0; j<6; j++)

        f[i][j] = 0;

int i, j, k;

main() {

    for (i=0; i<6; i++)

        for (j=0; j<6; j++)

            f[i][j] = 1;

        g[i][j] = f[i][j];

    g[i][j] = f[i][j];

    for (i=0; i<6; i++)

        for (j=0; j<6; j++)

            f[i][j] = 0;

            g[i][j] = f[i][j];

    }

}

for (i=0; i<6; i++)

    for (j=0; j<6; j++)

        f[i][j] = 0;

}

for (i=0; i<6; i++)

    for (j=0; j<6; j++)

        f[i][j] = 0;

## Function

Function having no arguments  
Type and no arguments ..

### Example

Programm declarations contain set of programming instructions or specific code.

This is used to reuse and/or code by writing once.

Easy to learn and easy to develop.

### Rules

Function declaration (prototypes)

Function definition (body)

Calling - Once

Later

Return type / Name of function / Arguments

Print ("Hello World")

return ("String")

void → no return void return

int →

float →

char →

void main ()

void disp(); // declaration.

void main()

present

disp(); // Function calling.

return();

;

Prototype

void disp()

Print ("Hello World")

;

Function with arguments

1) call by value / pass by value

void disp (int);

void main();

int i;

char c; // Declaration of variable

Print ("Enter the value");

scanf ("%d", &i); // Argument

displ();

getch();

3

1) definition

void disp(int t) // Parameters

int add (int t1, int t2)

2

printf ("\\n The result is %d", t);

Function having arguments

new & old hope.

(1) call by value / from my value

int add (int , int) // doee

1) environment

void main()

{

int n1,n2, s;

scanf ("%d %d",

&n1,&n2);

'Sout ("%d", s);

printf ("%d", s);

s = add(n1,n2);

printf ("\\n the sum value is %d", s);

getch();

1) environment

int add (int t1,int t2) // doee

int today

New  
Prog  
see how to do

tot = tot + tnum  
300  
290  
325  
945  
150

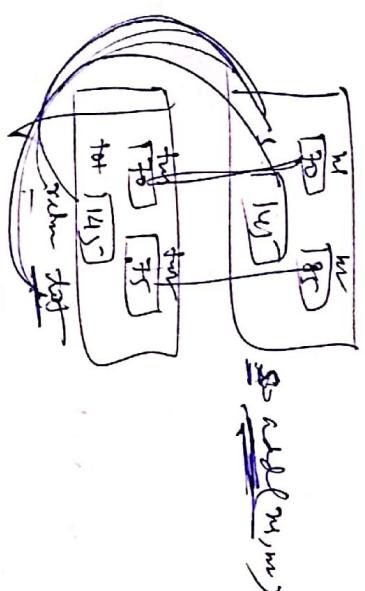
300  
290  
325  
945  
150

3

Call by value / pass by value

Calling Environment  
disp(int) → actual argument/parameter

Called Environment  
disp(int, int)  
→ formal arguments/parameter



2

2

### Addition (array) . .

#include <sidio.h>  
#include <conio.h>

```
int odd (int [ ]);
```

```
void main();
```

int n[5];

Elusen

for  $\{ \cdot \} = \{ \}$

Print (Entered premium)

5. 100

$$S = \text{add}(\omega);$$

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

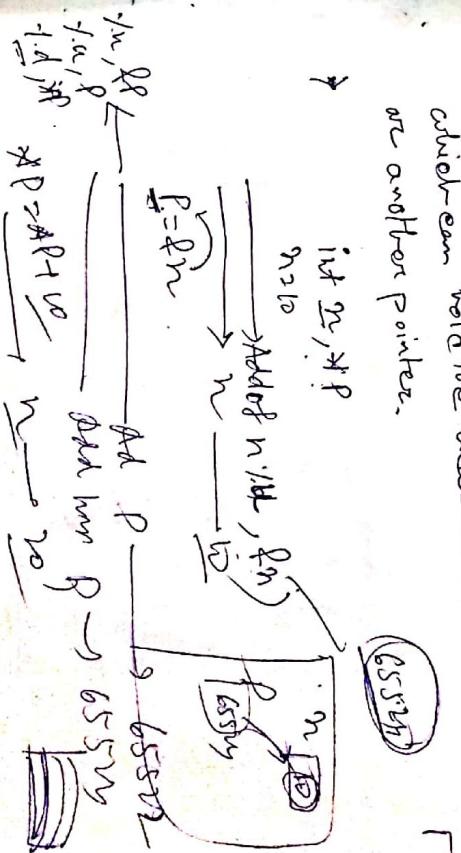
int add (int a, int b);

$\sum_{i=0}^n c_i x^i = 0$  for  $(c_0, c_1, \dots, c_n) \in \mathbb{R}^{n+1}$

Mar number

- Pointer is a special kind of variable which can hold the address of another variable or another pointer.

## Pointers



Scanned by CamScanner

TOP

## Recursive function

If a function call by itself  
Each value

void main();

2

char name[20], \*p;

char c;

printf("Enter the name:");

scanf("%s", name);

printf("The characters of your name

are");

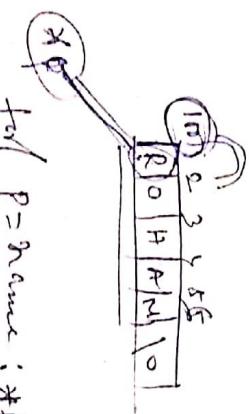
for(p=name; \*p != '\0'; p++)

{

printf("\n%c", \*p);

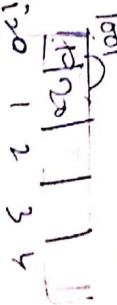
}

for(p=name; \*p != '\0'; p++)



\*fact(p);  
if(\*p == '\0')  
else  
return \*fact(\*p+1);

1  
— \*c, \*p



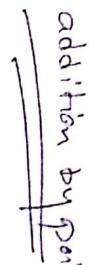
\*p

for(p=name; \*p != '\0'; p=p+1)

$$1001 + 0 \times 2 = 1001 \\ 1001 + 1 \times 2 = 1003$$

Character of name  
by pointer.

## Addition by pointer.



```
int i, n[5];  
i
```

$n = 0$

clsnem();

```
for (i=0; i<5; i++)
```

{

```
cout << "Enter the number";
```

```
cin >> n[i];
```

}

```
n[0] = 5;
```

```
for (i=0; i<5; i++)
```

{

```
n = n + (i+1);
```

```
cout << n[i];
```

```
getch();
```

```
void swap(int n1, int n2);
```

}

-

$t = n_1$   
 $n_2$   
 $n_1$

Difference between call by value and call by reference

Swap 1.C

```
void swap(int, int);  
void main()
```

{

```
int n1, n2;
```

clsnem();

printf("Enter first num - ");

scanf("%d", &n1);

printf("Enter second num - ");

scanf("%d", &n2);

printf("Before Swap n1=%d, n2=%d",

swap(n1, n2);

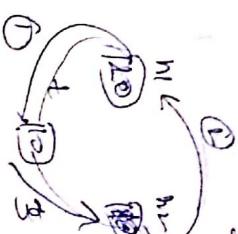
printf("After Swap n1=%d, n2=%d",

getch();

void swap(int n1, int n2);

}

Swapping



Scanned by CamScanner

Partwise swap with dynamic memory

print("Normal swap n1=%d and n2=%d")  
> return  
data2;

5

### Swap2.C

void swap (int &t1, int &t2);  
void main();

int myway;

```
class();
printf("Enter - ");
scanf("%d", &t1);
printf("Ans - ");
scanf("%d", &t2);
printf("Before swap n1=%d n2=%d", t1, t2);
swap (t1, t2);
printf("After swap n1=%d n2=%d", t1, t2);
```

date type together.

Structure  
structure is a user define data type or  
Customised data type which is use  
to combine the different primitive

int n1,n2

int n3

72 - 80

□ □ □

n n

(1) 4 4 - 1 1  
0 1 2 3 4

17 17 80 80

Name 26 by

Student

int sri; → number

char str[20]; →

int sno; →

int marks; →

int t1;

t = &t1;

atn = &t2;

};

void swap (int &t1, int &t2);

int myway;

class();

int t1;

int t2;

int ans;

int sum;

int n1,n2;

int n3;

int n4;

int n5;

int n6;

int n7;

int n8;

int n9;

int n10;

int n11;

int n12;

int n13;

int n14;

int n15;

int n16;

int n17;

int n18;

int n19;

int n20;

int n21;

int n22;

int n23;

int n24;

int n25;

int n26;

int n27;

int n28;

int n29;

int n30;

int n31;

int n32;

int n33;

int n34;

int n35;

int n36;

int n37;

int n38;

int n39;

int n40;

int n41;

int n42;

int n43;

int n44;

int n45;

int n46;

int n47;

int n48;

int n49;

int n50;

int n51;

int n52;

int n53;

int n54;

int n55;

int n56;

int n57;

int n58;

int n59;

int n60;

int n61;

int n62;

int n63;

int n64;

int n65;

int n66;

int n67;

int n68;

int n69;

int n70;

int n71;

int n72;

int n73;

int n74;

int n75;

int n76;

int n77;

int n78;

int n79;

int n80;

int n81;

int n82;

int n83;

int n84;

int n85;

int n86;

int n87;

int n88;

int n89;

int n90;

int n91;

int n92;

int n93;

int n94;

int n95;

int n96;

int n97;

int n98;

int n99;

int n100;

int n101;

int n102;

int n103;

int n104;

int n105;

int n106;

int n107;

int n108;

int n109;

int n110;

int n111;

int n112;

int n113;

int n114;

int n115;

int n116;

int n117;

int n118;

int n119;

int n120;

int n121;

int n122;

int n123;

int n124;

int n125;

int n126;

int n127;

int n128;

int n129;

int n130;

int n131;

int n132;

int n133;

int n134;

int n135;

int n136;

int n137;

int n138;

int n139;

int n140;

int n141;

int n142;

int n143;

int n144;

int n145;

int n146;

int n147;

int n148;

int n149;

int n150;

int n151;

int n152;

int n153;

int n154;

int n155;

int n156;

int n157;

int n158;

int n159;

int n160;

int n161;

int n162;

int n163;

int n164;

int n165;

int n166;

int n167;

int n168;

int n169;

int n170;

int n171;

int n172;

int n173;

int n174;

int n175;

int n176;

int n177;

int n178;

int n179;

int n180;

int n181;

int n182;

int n183;

int n184;

int n185;

int n186;

int n187;

int n188;

int n189;

int n190;

int n191;

int n192;

int n193;

int n194;

int n195;

int n196;

int n197;

int n198;

int n199;

int n200;

int n201;

int n202;

int n203;

int n204;

int n205;

int n206;

int n207;

int n208;

int n209;

int n210;

int n211;

int n212;

int n213;

int n214;

int n215;

int n216;

int n217;

int n218;

int n219;

int n220;

int n221;

int n222;

int n223;

int n224;

int n225;

int n226;

int n227;

int n228;

int n229;

int n230;

int n231;

int n232;

int n233;

int n234;

int n235;

int n236;

int n237;

int n238;

int n239;

int n240;

int n241;

int n242;

int n243;

int n244;

int n245;

int n246;

int n247;

int n248;

int n249;

int n250;

int n251;

int n252;

structure variable

struct student  
{  
 int id;  
 char name[20];  
 float marks;

①

struct student  
{  
 int id;  
 char name[20];  
};

int s1;

int s2;

int s3;

s1

student s1;

student s2;

student

student s3;

student

student s4;

student

student s5;

student

student s6;

student

student s7;

student

student s8;

student

student s9;

student

student s10;

student

student s11;

student

student s12;

student

student s13;

student

student s14;

student

student s15;

student

student s16;

student

student s17;

student

student s18;

student

student s19;

student

student s20;

student

student s21;

student

student s22;

student

student s23;

student

structure array members



struct student  
{  
 int id;  
 char name[20];  
 float marks;

int s1[10];

int s2[10];

int s3[10];

int s4[10];

int s5[10];

int s6[10];

int s7[10];

int s8[10];

int s9[10];

int s10[10];

int s11[10];

int s12[10];

int s13[10];

int s14[10];

int s15[10];

int s16[10];

int s17[10];

int s18[10];

int s19[10];

int s20[10];

street student

not sr  
class student  
does not total



(1)

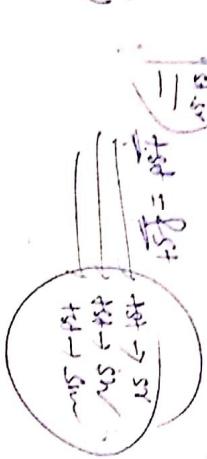
(2)



name  
marks

student

marks = 85  
name → sr  
marks → 85

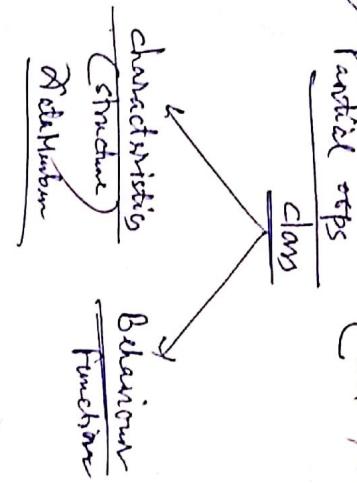


name  
marks

student

marks = 85  
name → sr  
marks → 85

C++  
Object oriented Programming language (System)  
in Partial steps (OOPL / oops)



name  
color

bird

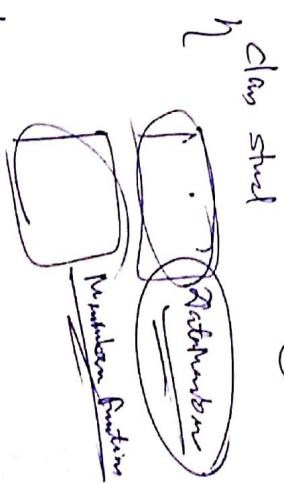
name → Harry  
color → blue

Conceptual

Substance  
R

Instance of an class  
is called

object



student  
R

Name  
Number

Access specification

- 1) private
- 2) public
- 3) Protected

## 4 Features of C++

1) Abstraction

2) Encapsulation

3) Inheritance

4) Polyorphism

## class student

private : // Protection / Data hide  
int sr;

char sub[5];  
int total;

public : // Encapsulation

void main() → iostream.h

void main()

int s[5];  
char sub[5];

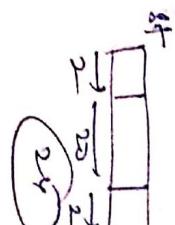
cout << "Enter student no : "

cin >> m

int sum  
operator<<

cout << cin >>

int sum  
operator<<







Polymer Sci

卷之三

Some of his remarks were

from 2 points + time studies and often is dynamic

- Static  $\Rightarrow$  function over loading.
  - function overloading  $\rightarrow$  It is a kind of polymorphism where we can define one or more functions with same name having different signatures.

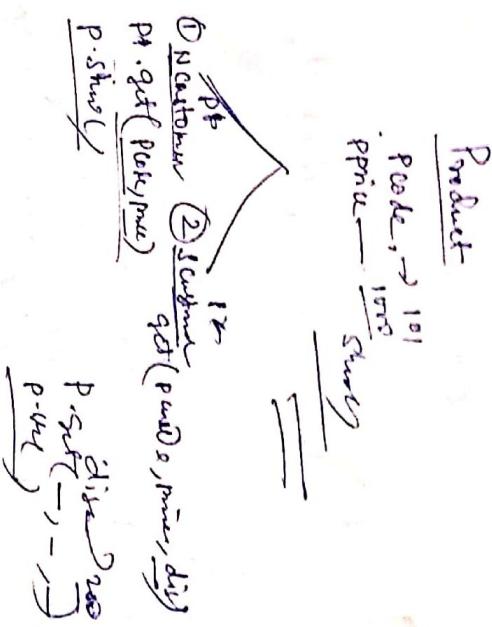
## Function Signature

- (i) Different No's arguments

  - add(10, 20)
  - add(20, 30, 50)
  - Diff wirt types argnt
  - add(10, 20)
  - add(10, 5), 20, 5

3) Different sequence even though

  - add(10, 20, 5)
  - add(10, 5, 20)

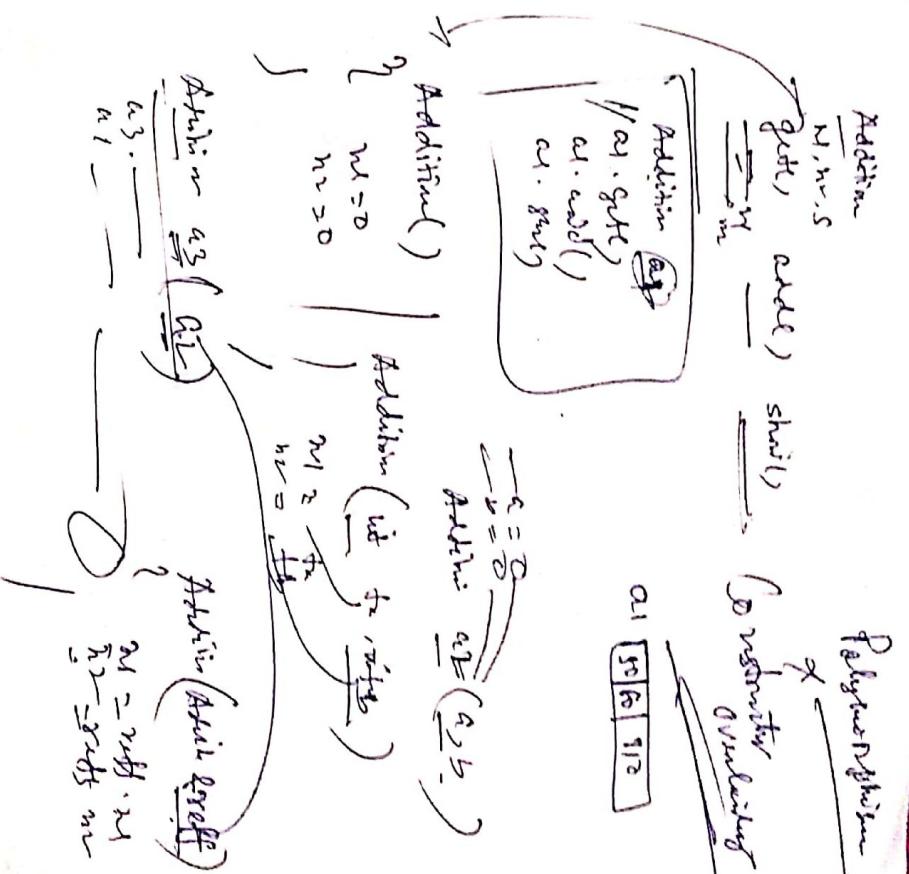


## Constructor

- ↳ It is a special kind of function which is used to allocate memory of an object at a class.
- ↳ There is always a default constructor in our programme and whenever we declare any object during that time, it calls its constructor implicitly.
- ↳ We can also declare constructor in our programme to initialise data member of a class.
- ↳ If the constructor name must be same as the class name.
- ↳ If there is no return type in constructor then constructor should be declared in public scope always.
- ① There are 3 types of constructors:
  - ↳ Default zero level constructor by copy overloaded or 11
  - ↳ Copy constructor

## Addition

## Polymerization



class addition

{  
private : int m1, m2;

public : addition () // zero level constructor

{

  m1 = 10;

  m2 = 20;

}  
addition (int m1, int m2) // parameterized constructor

{  
  m1 = m1;

  m2 = m2;

addition(addition & next) copy constructor

cout << "Enter number";  
cin >> a;  
cout << " " ;  
cout << " " ;

cin >> n1;  
n1 = ref.n1;

n2 = ref.n2;

cin >> b;

addition obj(a, b);

void get()

{  
cout << "Enter number";

cin >> n1;

cout << " " ;

cin >> n2;

}  
void add()

{  
s = n1 + n2;

s.  
void show();

{  
cout << "The value is " << s;

>i  
void main()

{  
int a, b;  
addition obj;

cin >> a;

obj.set(a);

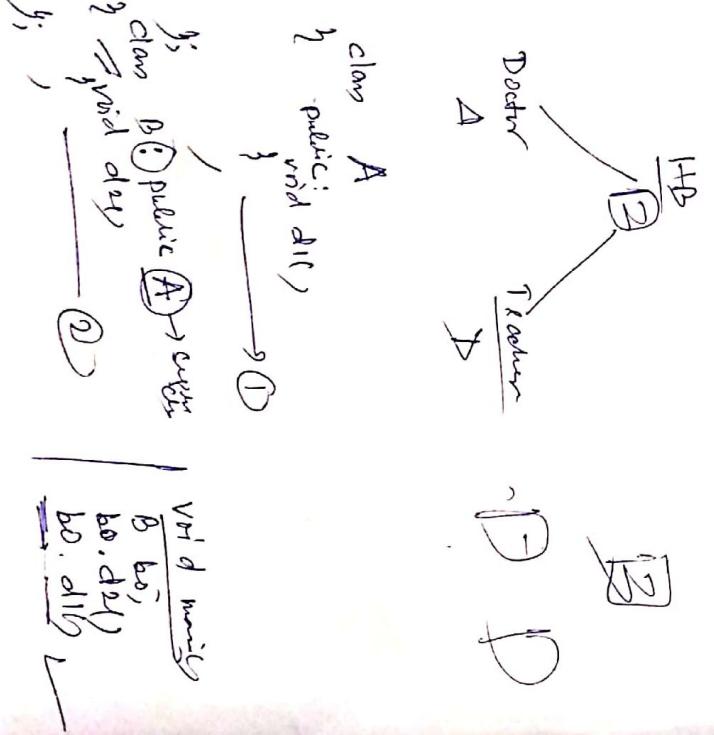
obj.add();

a1.show();

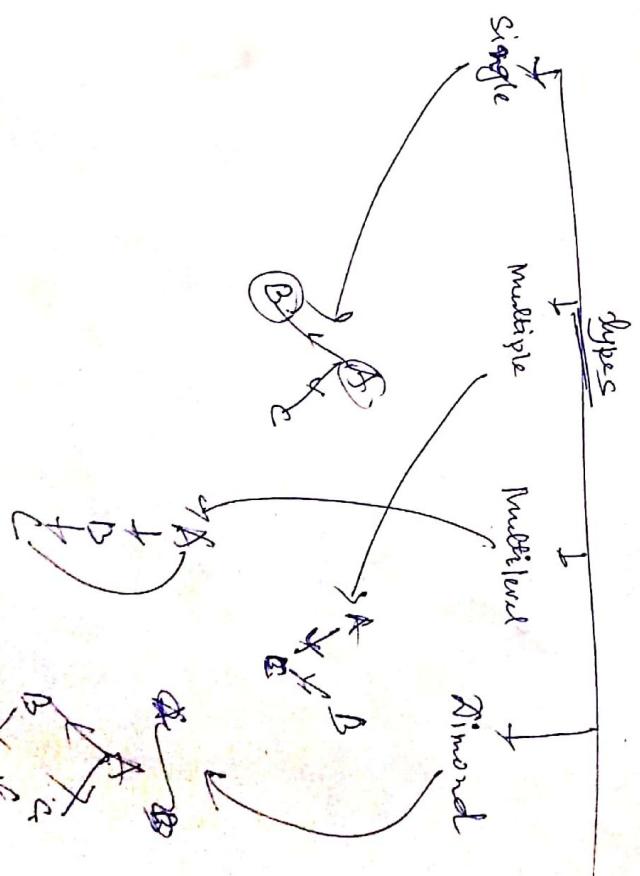
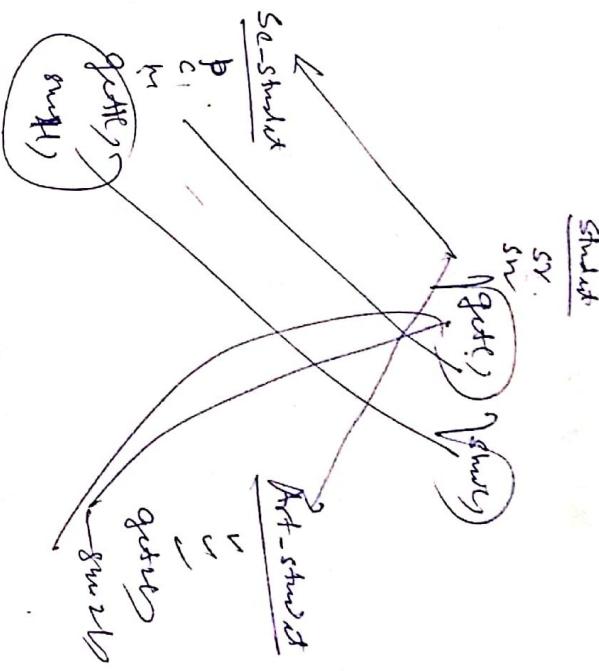
cout <<

## Inheritance

- Using this feature a class can get the property from another class. that mean from a super class all its child class is inherit the common and generalized property.
- # It is a ~~new~~ hierarchy structure of oops.
- # Super / base / parent class
- # Sub class / derived / child class



## Smart pointer



## Multiple Inheritance

A  
B  
d1()

class C : public A, public B

Function overriding is a procedure where sub class function overrides the definitions of a super class function. This feature is called function overriding. Function Overriding is a part of polymorphism.

1

Show

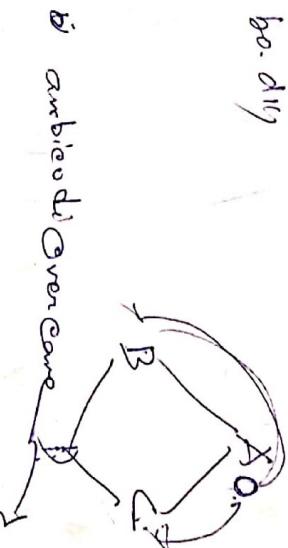
Ex:  
Ambiguity of m1

A  
↓  
d1()

B  
↓  
d1()

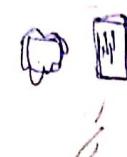
C  
w. d1()

A::d1()  
B::d1()



A  
↓  
d1()  
B  
↓  
d1()  
C  
↓  
d1()  
=

A::d1()  
B::d1()  
=



Use virtual key word to solve.  
ambiguity.

class A {  
public:  
 A();  
}

## Function Overriding

Function overriding is a procedure where sub class function overrides the definitions of a super class function. This feature is called function overriding. Function Overriding is a part of polymorphism.

## Friend Function

Class A

m

friend void add(A &),  
int S,  
S = m + n

void add(A &)

m

n

m = m + n

Class B

m

friend class B;

A()  
(x=100)

m = 200

mid add(A = B)

int S

S = m + n

mid add

m

B  
m  
m = m + n

add()

mid add(A = m, B = n),  
int S,  
S = m + n

add(A)

m

n

m = m + n

Class A

m

n

m = m + n



① operator function → return type  
② object return type  
③ object array

