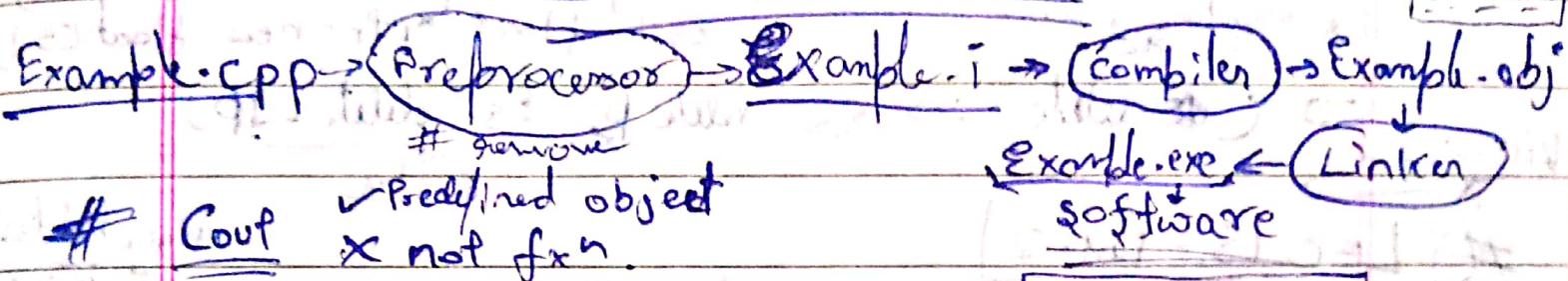


C++ //



Ex) `Cout << "Hello world";` at `\n == endl`

3) cout << "Sum of " << a << "and " << b << " is " << c;

cin >> a;

Ex. $\sin x \gg a \gg b$;

inline fix

#. Struct is not necessary

to use variable

- May use Public: / Private: to make data accessible or not out of this Project Block.

The members of class are by default private

Lec 7
Part - I

membership of class

of class.

~~the type~~ Class Name (::) delimit ()
scale resolution open.

Constructor => it is an (no return type) class member which is undefined fxn in a class which gets automatically called when we make an obj using this class.

friend/fxn keyword (friend)
in class → friend void func
• defined out of class.
• called in main like fun(c1);

Constructor in Inheritance.

in constructor
• Call order: child to parent class
Run order: parent to child class.

→ In child class Constructor B() : Constructor A()

DMA (new) :-
* malloc/alloc
e.g.: Complex *ptr = new Complex;
e.g.: float *ptr = new float[5];

→ (delete) :> e.g.: delete p; e.g.: delete []p;

LEC 18 - Part 2

Template :- template < class X, >

Note → In Constructor (Inheriting)
main {
 class B {
 int x; int y;
 public: x=x; y=y;
 };
 B B1(B2);
 B B3(B2);
} # Can run.

X fxn name (X a), X b
&

Let # File Handling :- #include <fstream.h>
main ()
{ ifstream fin; // reading from file
 char ch; fin.open("file.dat");
 while (!fin.eof())
 { cout << ch; fin >> ch; OR [ch=fin.get();]
 if (fin.eof()) // end of file
 fout << "Hello";
 fout.close(); // Writing in file
 }
 fin.close(); // Readings from file;
}

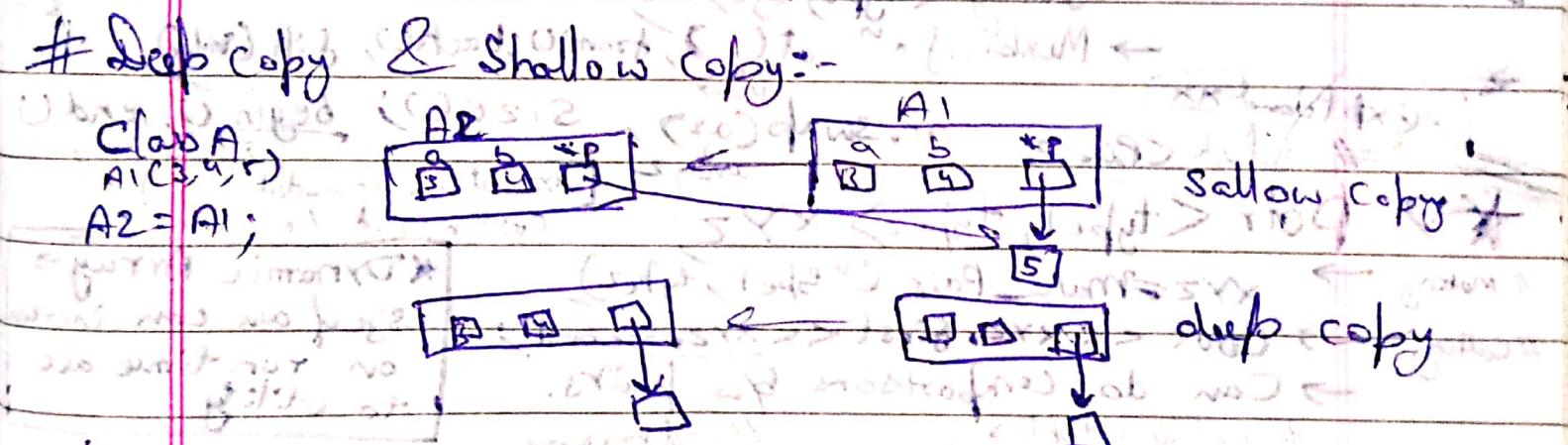
File opening modes
 1) read mode $\text{fopen("f1.dat", ios::app);}$
 2) write mode $\text{fopen("f1.dat", ios::out);}$

* tellp(); // returns position
 int pos; pos = fstream::tellp();

* seekg(); → seek the tellp point to any position
 1) fin or fpt. seekg (int pos); (current, end)
 2) f1.seekg (int pos, ios_base::beg)

* seekp(); → seek the tellp point to any position
 1) f1.seekp (int pos); (current, end)
 2) f1.seekp (int pos);

Initializer list
 constructor () : a(5), b(6)



Type conversion:
 1) Class type = Primitive type (Using Constructor)
 2) Primitive type = Class type (Casting operator operator type())
 3) Class type \neq Class type (Using Constructor or in class)

④ (② Product)
 a = Product.getX();
 b = Product.getY();

* Direct casting (Object b)

Ques 1 # Exception Handling:

If namespace XYZ
If {
 // Declarations

* namespace xyz=XYZ;
 // Name change

// Using Namespace XYZ;

Nested Class →

try {
 y // throw in if
 y // throw to
} catch (Type1 arg) {
} catch (Type2 arg) {
}

if Virtual & Destructor
{ }

Containers → Array #include <array>

array <type, size> xyz;
→ Member fxn at(), front(), back(), fill(value),

(Explained xx)

#include <pair>

at(index), swap(pair), size(); begin(), end();
array

xyz.at(i).show-data();

Customized for class

// making xyz = make_Pair("type1", "type2");

// calling cout << xyz.first << xyz.second;

→ Can do comparison b/w pairs.

#include <tuple>

tuple <type1, type2, type3> xyz;

→ xyz = make_tuple(type1, type2, type3);

→ cout << get<0>(xyz);

* Dynamic Array →
Size of arr can increase
on run time acc
to utility

* → Vector Class → Arr. that supports Dynamic arr.

#include <vector>

* Capacity only doubles (x2)



→ Vector <type> xyz {10, 30, 20};

// Declaration
 Vector <char> v2(5, 'a');
 Vector <char> v3(5, 'a', 'b');

 Vector <type> xyz;

→ xyz.capacity(), xyz.push_back(x), xyz.pop_back();

xyz.size(), xyz.clear(), front() & back();
(No. of elements)

|| last value



//(iterator)

left

vector <int> :: iterator \neq if $\rightarrow xyz.begin()$ Date _____
Page _____

vi.insert(i+3, x);

#include <iostream>

* → **List Class** → list <int> l;

//(Program using iterator) → li.end(), push-back(x), push-front(x)

Numeric array :- indexes are numbers
Key → 0 1 2 3 ...
Value ←

Associative arr:- (Customized
ie, name of indexes.(key))
Amit ranjith Baskar

→ clear(), empty(), size(),

xyz.insert(pair<key_type, value_type> (x, y));

front(), back(),
pop-back(), pop-front(), sort(),
reverse(), remove(x), clear(),
//(element list)

* → **map class** → (associative arry)

#include <map>

→ map <key type, value type> xyz;

//(Assignment) li xyz[{key, value} {key, value}, ...]

(iteration) OR xyz[key] = value;
→ iterator p → xyz.begin();

p → first OR p → second
(key) (value)

"operators
(+, +, ==, !=,
<, <, >, >,
, , ,)
>>, <<

include <string>

* → **String Class** → string s1 = "Hello";

→ // (mixed operations) s1 = s2 + "123" / s2 + (char arr)

→ xyz.assign("..."), append("..."), length

xyz.insert(index, "..."), replace(index, "...")

.erase(); • find("...") // ((2, 2, "A"))

• // (return index)

• rfind("..."), s1.compare(s2); .. c-str();

• // (from reverse) // (return 0 if same
-1 if opp to dic order
1 if in dictionary order)

and (& return
, -size()

FIFO

* → **Stack class** →

stack <int> s;

s.empty()

s.size()

* → **Queue class** → queue <int> q;

s.top()

s.pop()

s.push(g)

q.empty(); q.size(); q1.swap(q2); **LIFO**

q.push(g); q.pop(); q.front(); q.back();

$5 = 0 \underline{101}$

$\underline{0111}$

stl:

--builtin_popcount(x);

5

[No. of Setbit = ?]

Bitwise Operators ; $>>$, $<<$, $|$, $\&$, \sim , \wedge

Right shift, Left shift, OR, AND, XOR

(Is complement)

* [int getBit (int N, int pos) {
 3 return ((N & (1 << pos)) != 0); }]

1 OR 0

* [int setBit (int N, int pos) {
 3 return (N | (1 << pos)); }]

E.g: setBit(5, 1) = 0111 = 7

* [int clearBit (int N, int pos) {
 3 return ((N & ~mask = ~(1 << pos));
 3 return (N & mask); }]

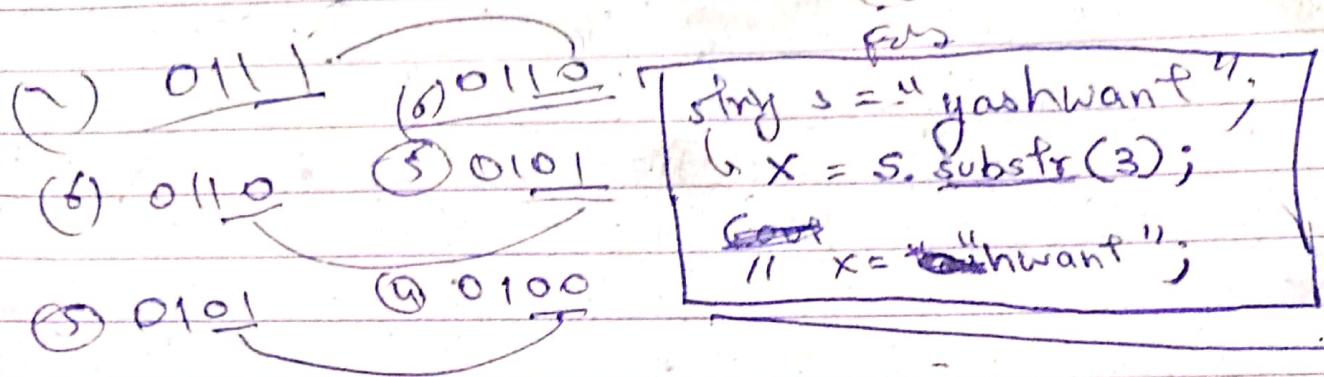
* [int updateBit (int N, int pos, int value) {
 3 int mask = ~(1 << pos);
 3 N = N & mask;
 3 return (N | (value << pos)); }]

Q Number is power of 2.

Q Print all sub-arrays.

Q Arr all no. $\times 2$ except 1 find? (Using XOR)

Q Arr 111, 111, 11 2 find? ([])



* Sieve of Eratosthenes

[[Pre marked for use E.g. All Prime No's in N]]

* Inclusion - Exclusion Eg: choices A, B Student.

Eg: # 1-1000 - No divisible by 5 & 7?

$$\text{total student} = n_1 + n_2 - n_3$$

* Euclid Algo → for GCD (Principle small No's by difference or $\frac{n_1}{n_2}$)

* Tower Of Hanoi → (Recursion)

* DNF SORT (Only for 3 type element)

$O(N)$ Eg: [3, 2, 1, 1, 2]

Low Unknown High

Low Mid

High

while ($mid \leq high$)

$\leftarrow 1$ (mid++)

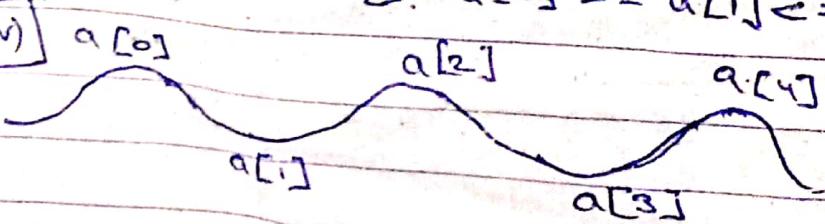
< 1 (low++)

> 1 (high--)

$= 1$ (mid++)

* WAVE-SORT

E: $a[0] \geq a[1] \leq a[2] \geq a[3] \dots$



for ($i=1$ to $n-1$, $i+=2$) {

if $a_i > a_{i-1}$

swap($i, i-1$)

if $a_i > a_{i+1}$ & $i < n-2$

swap($i, i+1$)

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

★ Algo - for Next larger No. using Array digits

① Traverse Array from right find element X which is not in decreasing order

② On right of X , find Y , smallest greater than X .

③ Swap $X \leftrightarrow Y$.

④ Reverse Array sequence after X position

Ex: 1 2 5 4 3 $X=2$; $Y=3$
 SWAP: 1 3 5 4 2
 Reverse: 1 3 2 4 5

Q1) Circular max Sum of Subarrays?

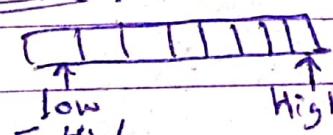
Ans) Kadane's Algo.

HPM

Eg

Q2) Pair Sum = K ?

Q1

Ans) 2-pointers 
 if $low++ \rightarrow$ large No.
 $high-- \rightarrow$ smaller No.
 while($a[low] + a[high] \neq K$) {
 if $low++$
 } else $high--$

for ($i < n_1$)
 for ($i < n_2$)
 for ($K < n_3$)

Matrix Multiplication →
 Ans

$M_{n_1 \times n_2} N_{n_2 \times n_3}$

$$[X_{n_1 \times n_3}] = \{0\}$$

$$X[i][j] = M[i][k] * N[k][j]$$

functions

- ① `max(a, b);`
- ② `sort(a, a+n);`
- ③ `swap(a, b);`
- ④ `to_string(xyz);`
- ⑤ `reverse(str.begin(), str.end());`
↳ // Must \rightarrow #include <algorithm>
- ⑥ ~~int n = size_of(x); (Bytes)~~
- ⑦ ~~*min_element(a, a+n);~~
- ⑧ ~~*max_element(a, a+n);~~
- ⑨ ~~-arr.size();~~
- ⑩ ~~tolower(x);~~
↳ char
- ⑪ ~~stoi("yash");~~
↳ // string-to-int
- ⑫ ~~atoi("yash.55");~~
↳ // gives integer only=55
- ⑬ ~~str.length();~~
- ⑭ ~~str.substr(index);~~
↳ ex: if str="yash"
↳ index=1 \rightarrow ob=yash

values

① `INT_MAX`

↳ string str;

// max value of Long Long Int. //

DATE
PAGE NO.

10/10/2023
1/1

Sort str;

// iteration in: (xyz) //

for-each(str.begin(), str.end(), [char & c])

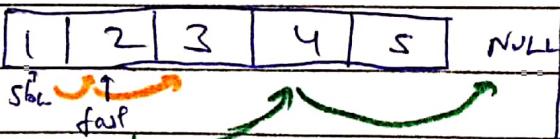
Conf c.

Type conversion.

- ① char to int \rightarrow (int)(c' - 48)
- ② string to int \rightarrow stoi("yash")
- ③ int to string \rightarrow to_string(xyz)
- ④ int to char \rightarrow (char)(i + 48);

- ⑮ `next_permutation(str.begin(), str.end());`
// permute original string to next
Lexicographic order
// also return if next permutation
possible or not (T/F)

Concept Linked List Middle



while (`f`) {
 `s=f+1`
 `f+=2`
 `s->mid`}