Object oriented Programming

- Const keyword
  - ↳ Used to declare the variable, function or object is immutable
    - ↳ cannot be reassigned.

* Compiler stores const variable in the ROM part of memory :- Optimisation

1) const int x = 10;

lvalue → variables having memory location
Rvalue → " doesn't have mem. loc

int &a = b
↓ ↓
lvalue Rvalue.

2) Const with pointers

(a) Const int *a = new int(2)

↳ const data, Non-const pointer

*a = 20 → This cannot be possible

b = 20;
a = &b → possible

int const * a = new int(2)

→ same as before

(b) constant Pointer, Non-constant data.

int * const a = new int(2);

↳ constant pointer

but content is non-constant

(c) Const pointer, const data.

const int *const a = new int(10);

↳ both are constant

Const with Function

Method (const)

class

↳ It's member variable cannot be modified by that method

int getx() const

{

}

if kisi function k andar a const obj le liya then use/ce andar se if we call any function then that function should be constant. ~~(should)~~

mutable int x; (used for changing the value of members in when const is declared int method.

→ INITIALIZATION LIST

abc(int -x, int -y, int -z = 0) : x(-x), y(new int(-y)), z(-z) { }

↳ This is required when the one of the data variables are constant so we initialize constructor using this.

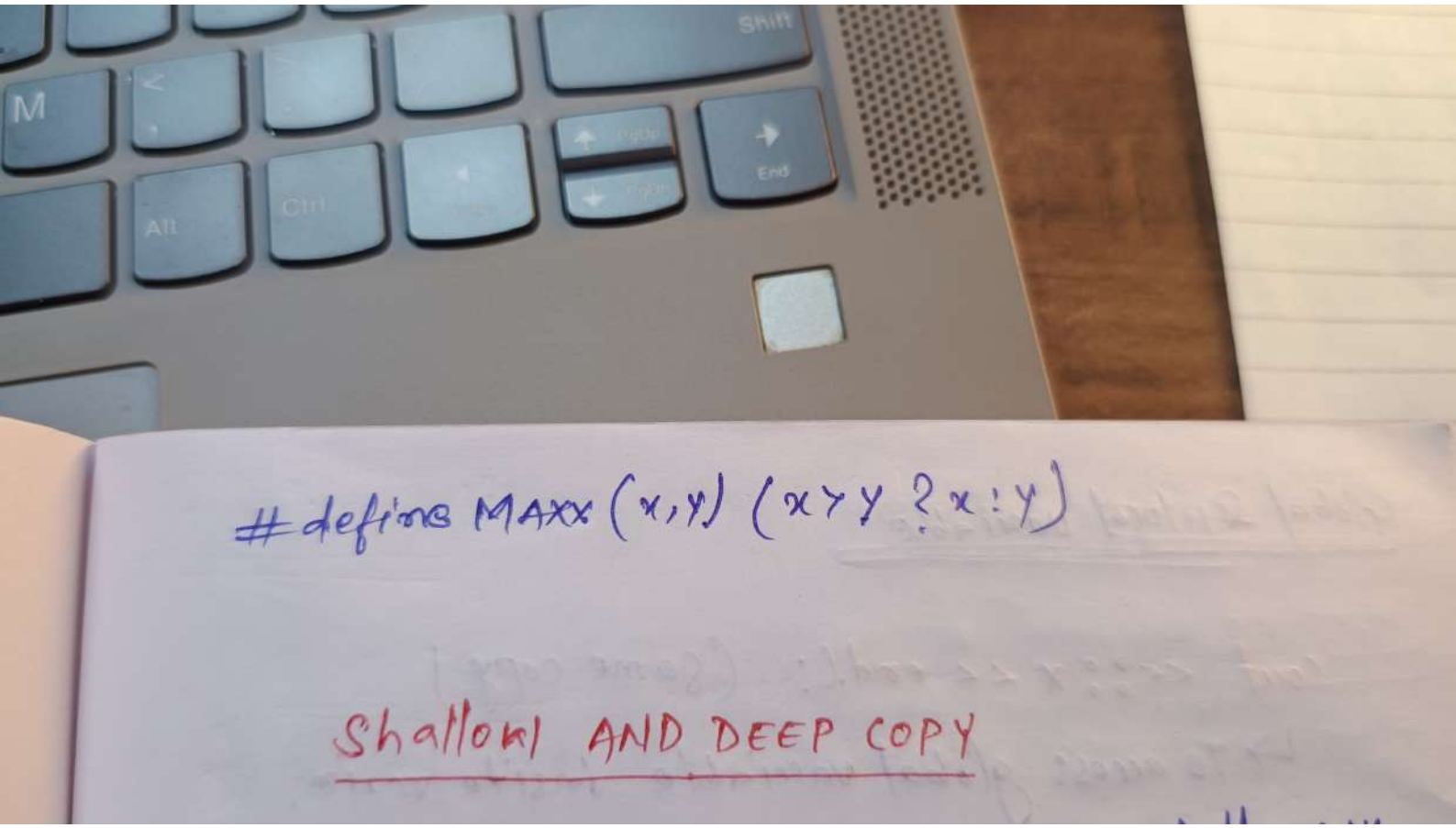
→ Macros keyword (preprocessor) → Before compile

↳ It replace all # define in the place where it is used

#define PI value.

variable → its value.

→ It increases readability




```
#define MAXX(x,y) (x>y ? x : y)
```

Shallow AND DEEP COPY

Global & local variable

cout << ::x << endl; (same copy)

↳ To access global variable inside main.

⇒ Most local is first preferred

local variable is scoped inside a function.

Global Variable:

1. written outside the fun^{n}
2. Accessible by all fun^{n}

local variable:

1. Writth^{n} inside the func^{n}
2. Acc. ~~to all~~ inside tha f^{n} scope only
3. scoped.

static keyword in class

Static data Members:

That variable is going to share memory with all of the class instances.

Static Membes function :-

There is no instance of that class being passed into that method.

⇒ static member function can only static members data.

⇒ ~~static~~ they will be called as class member or class function like

```
abc obj1;
abc::print();
```

ABSTRACTION

→ Delivering only essential information to the outer world while masking the bg. details

→ seperate interface and implement.

Abstract class

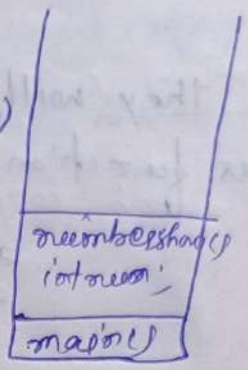
→ That it contains at least one pure virtual function and these classes cannot be instantiated.

~~→ Abst~~

INLINE FUNCTION

Syntax:

```
Inline void membershow(int num)
{
  cout << num << endl;
}
int main()
{ membershow(10);
```



}

Advant

inline
→ function calling overhead is bachta hai
→ Time bachta hai

Dis
→ Used in small size function

Inline function

Instead of calling function the statements of functions are pasted in calling function.

preprosser → Before compiling
Inline → Compiler think whether the size of function is large or not

friend keyword in C++

- Syntax

for friend class

```
friend class class_name;
```

for function

```
friend ~~void~~ funct^n_declarat^n
```

eg:

```
friend void print (const A &);
```

Can constructor be made private (ctor)

ctor → constructor
dtor → destructor.

→ yes it can be made private.

→ Singleton class.