DataStructures :

Structures

COLLECTION OF NON PRIMITIVE USER DEFINED DATATYPES

- Collection of data members under one name is structure
- Data members can be of similar type or non similar type
- When structures is called in the main( ) program then it will consume space accordingly to the data members types it contains in the memory

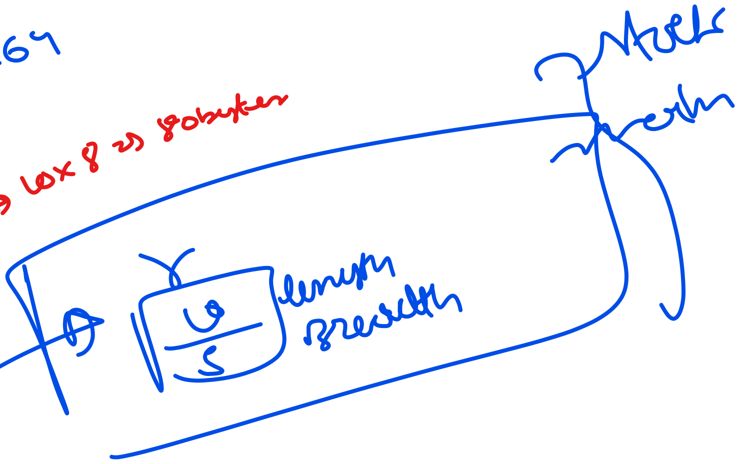
An example of structure is a program of a rectangle

→ memory use, when only used in-main, etc.

```
Struct Rectangle

{
        int length;  +4        cpp x64
        int breath;  +4
                     ------
                     8 bytes
};
```

→ 10 x 8 => 80 bytes

```
Int main( )
                array of struct
{ struct Rectangle arr[10];

struct Rectangle r ;                    - Declaration

Struct Rectangle r = { 10, 5 } ;        - Declaration + Initialisation

r.length = 15 ;                         - . Is is used to access a member

r.breath = 10 ;

cout << r.length << endl << r.breadth < endl;

Printf( " Area of rectangle is %d" , r.length * r.breath ) ;  - Accessing the members

}
```



. } dot operator → Read → write :)

Use of structures :

Structures is used to combine data under one name , thus some example usage of structures is

- In Complex numbers a+ib, i = √-1
- In student details
- In Employee Details
- Bank Details etc
- Defining Shapes etc…
- plays
  cards Four 1-10, J, Q, K
  shapes 0 1 2 3 4
  colour 0 1

char name[25]} 25 bytes

keep in mind (didn't work :))

*

{ int → 4
int → 4
char → ~~1~~ 4
}

Padding → nearest biggest size, which pe given for last

but wrong, will be or 1

Sell one tablet vs Sell strip

accessible Easy for Pharmacist

struct { {≡} r1, r2, r3 }; } global (outside main) ∀

struct r4;

%lu } refer from %d