Datatypes

- Peermitive → int char (decimal Number Alpha)
  - Float | double (Floating Number)
  - void → Function | Pointer
    - Null

- Not Peermitive
  - Pointer
  - Array
  - Function

- User define → ① struct ② union
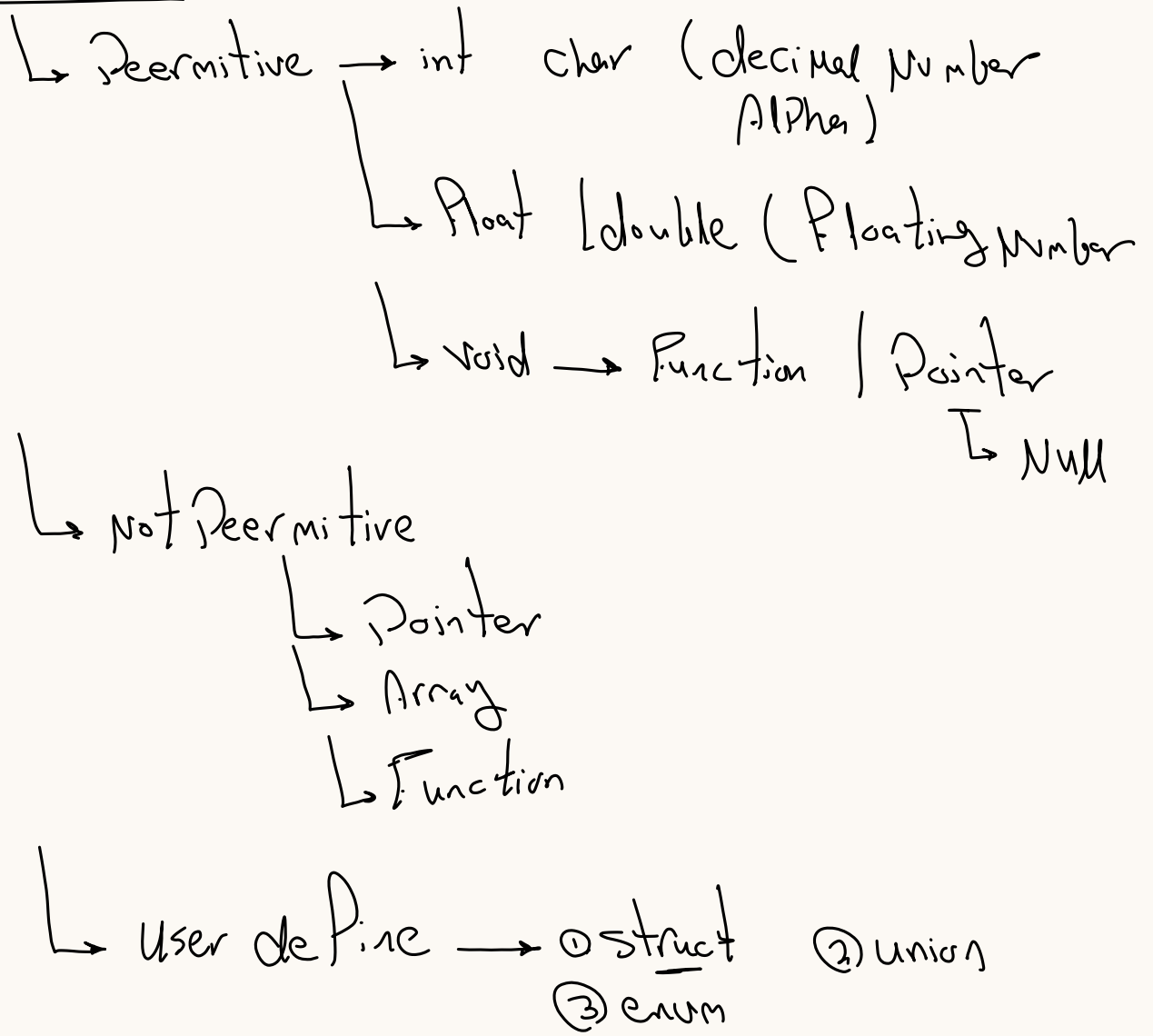 - ③ enum

---

- struct → How create new datatype (struct);
  - → How create new object (var) →
  - → How Access this object
  - → How Handle in memory ⇐
  - → Packing & Padding
  - → how create Pointer → Point for object from struct & How Access. (Pointer to struct)

→ How create struct

① struct Namestruct
    ς
        Permitive Datatype var Nam;
            ''   ''   v ι∨ :
            ι.   ''   '' '' ;

    S;

    ex:. struct Plater
        ς
            char Name [20]; ⟶
            char No   ; ⟶
            int Salary ; ⟶
        S;

Notes → Creation
uPeration No take
SPace from Memory

struct Plater Pn;

② typedef struct [✓✓✓]   → Plater P₁;
    ς
        _____
        _____
        _____
    S Plater;

③ typedef struct Plater   → Plater P₁;
    ς   → struct Plater P₂;
        _____
        _____
    S Plater;   unsigned char
                            unint8

④ struct Player ⟨

```
    ≡≡≡
    ≡≡
    S;
typedef struct Player  Real Players;
```

How create object From struct

↳ <u>Declartion</u>

→ struct structName   objectName; →
  struct   Player   P1   ; →

→ struct Name   object name;
  ↳ to use this must be use typedef
  Player   P2 ; →

↳ Definition

↳ struct structName objectName =
  ⟨ "Mo Salah", 32, 50000⟩;

↳ struct Player   P1 =
  ⟨ "CR7" , 7 , 50 ⟩;

Player
  ⟨
  Name ←
  No ←
  salary ←
  ⟩

typedef struct
{
    int ID;
    Float degree;
} student ;

Creation

student S1 = { 0 , 455 },
    ↳ Definition
student S2 ;
    ↳ Declartion

How Access Normal object (Dot access)

student S3 ;

$S_3 \odot ID = 20 ;$

$S_3 \odot degree = 40 ;$

$S_2 \cdot ID = 60 ;$

Printf (" ID for student 3 = %d\n", $S_3 \cdot ID$);

labl
↳ empollyee → Name | ID | Salary
        ↳ SGn from user Information
            ↳ 3 empollyee
                ↳ Print total salary

→ struct student Class1 [20];

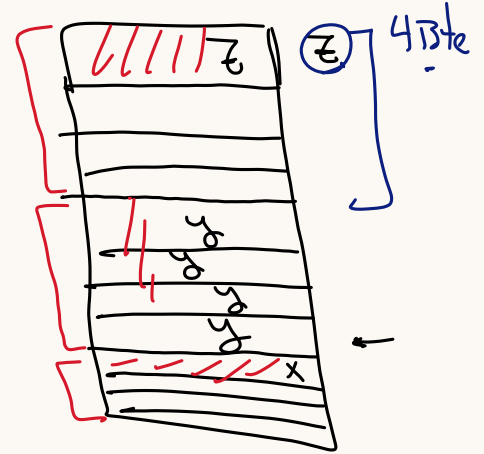Class1 [0] . ID =
Class1 [1] . degree =

---

struct students → step = MaxData
  Char z;
  (int)   y; (4Byte)
  Char x; ←

';

struct student S1;
    └→ sizeof → (12 Byte)   used 6 Byte
                              only

Solution Padding

→ ordered var
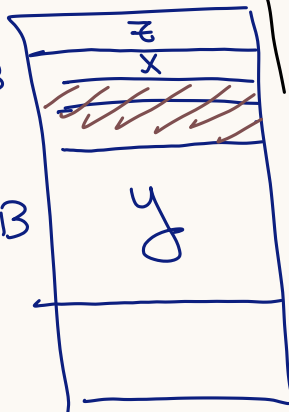└→ struct student q → 4Byte
      Char z;
      Char x;
      int   y; 4B

    &;

struct student S1;

  8Byte

→ 2Byte not
      used.

Solution Packing   → 2Byte

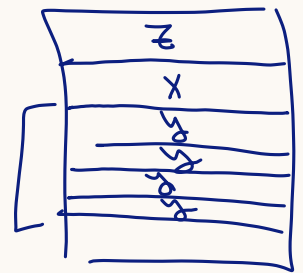#Pragma Pack (2) →
                └→ step = 2Byte
→ struct student
  '
    Char z;
    Char x;
    int y;

  S;
struct student S1;

* Pointer to struct

student S_1 ;

student * Ptr = &S_1 ;

S_1 : ID = 20;

$$[*Ptr . ID = 50;]$$

$$*(0X00) . ID = 60;$$

Ptr -> ID =

Arrow Access (Pointer to struct)

0X00

S_1

20

0X00   Ptr