CS & IT ENGINEERING

C Programming

Pointers & Arrays

Lecture No.- 01

By- Satya sir

Recap of Previous Lecture

- Computer Fundamentals
- Bits, Bytes
- Types of Prog. Languages
- Translators
- 'C' features, history, Structure of 'C' Program
- Types of Expressions
- Data Types
- 'C' Tokens - Identifiers, keywords, Constants, operators, Strings, Special Symbols
- I/O Statements - scanf( ), Printf( )
- Control Statements - if, if-else, Switch, Loops, break, continue, goto

Topics to be Covered

Pointers*

- What is Pointer ?
- Pointer declaration
- Pointer Initialization, Assignment
- Pointer size, datatype
- Pointer Arithmetic
- Types of Pointers

Topic : Pointers

Computer Functionality

O.S

[fastest]
CPU

ALU

CPU Registers

CU

I_3

Primary Memory (RAM)

I_3

Addrem Y

[Slowest]
(Hard disk)
Secondary memory (File System)

S/w

Add X

I_1
I_2
I_3
I_4

Data

Addrem

(Low level lang)
i/p → Instruction
(Part of Drivers)

** * ⇒ For accessing content from memory, addrem is needed.

Data, Instruction
Content

I/O unit

Topic : Pointers

Pointer : A Variable, that holds address of ~~another Variable~~ any content [data, Instruction]

- Variable
- Array
- Structure
- Union
- function
- File
- Pointer

- Pointer Declaration

Syntax: `DataType * Pointer Name ;`

- DataType → Any datatype
- Pointer Name → Any valid Identifier

Ex:
```
int *p;
char *q;
float *r;
struct S *t;
FILE *i;
void *j;
```

Topic : Pointers

& == address of

Initialization

While declaration Syntax: datatype *Name = address of data;

Ex: int a, *P=&a;

Later declaration Syntax: datatype *Pointer;

⋮

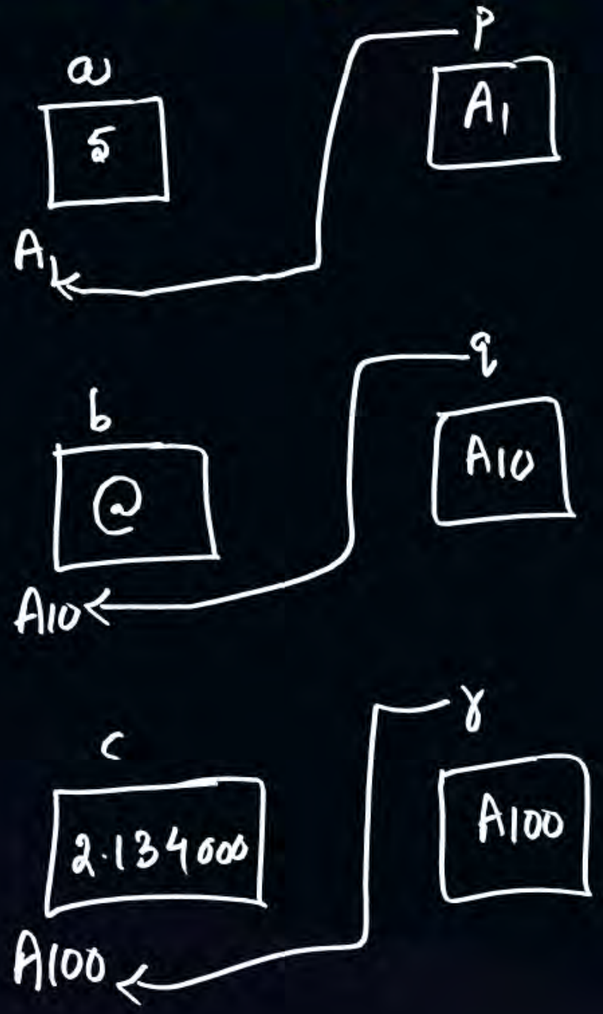
Pointer Name = address of data;

Ex:
```
int a=5, *P;
Printf("%d",a);
a++;
a=a---a;
P=&a;
```

Topic : Pointers

Pointers DataType : The datatype of Pointer is always unsigned Integer.

```
int a=5, *P;
char b='@', *q;
float c=2.134, *r;
P=&a;
q=&b;
r=&c;
```



a is of type Integer

b is of type character

c is of type float

P is of type integer → holds integer address

q is of type integer → holds character address

r is of type integer → holds float variable address

Unsigned (+ve) Integer [Whole Number]

{ P=&b (or) P=&c
q=&a (or) q=&c
r=&a (or) r=&b } Invalid, Error.

Topic : Pointers

Void Pointer : A Pointer, that Can Point to any type.

```
int i;  char j;  float k;
Void *x;
```

```
x = &i;   (or)
x = &j;   (or)
x = &k;
```

Valid assignments

- Void Pointer is also known as Generic Pointer.

Topic : Pointers

sizeof() → keyword / Operator / Predefined Function } It returns the memory allocated in Bytes

Let 16-bit Processor

```
int i=5, *P;
char j='$', *q;
float k=1.017, *x;

P=&i;
q=&j;
x=&k;

Printf("%d", sizeof(i)); // 2
Printf("%d", sizeof(j)); // 1
Printf("%d", sizeof(k)); // 4
```

```
Void *v;    Printf("%d", sizeof(v)); // 2
Struct S *t; Printf("%d", sizeof(t)); //2
```

Pointer Size is always Equal to Integer Size

```
Printf("%d", sizeof(P)); // 2
Printf("%d", sizeof(q)); // 2
Printf("%d", sizeof(x)); // 2
```

Topic : Pointers

Pointer Arithmetic

Valid operations on Pointers

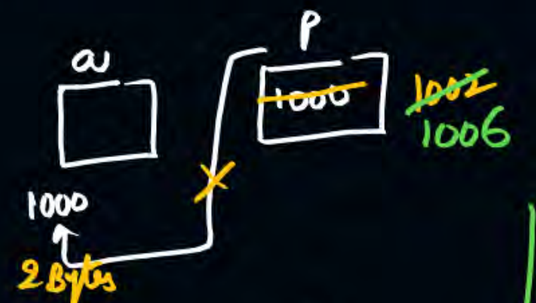
1. Increment (++)
2. Decrement (--)
3. Addition or Subtraction with Integer
4. Addres assignment
5. Assignment to another Pointer of Same type
6. Comparision (!=, ==)
7. Subtraction b/w Pointers of Same type

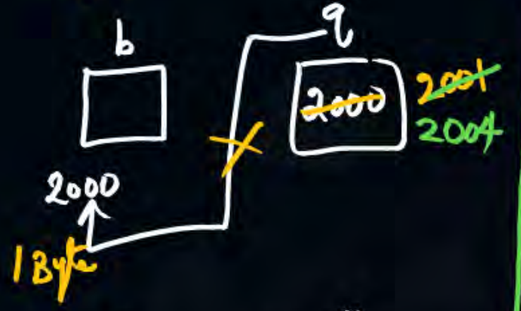
The amount of addition (or) Subtraction to Pointers depends on data-type, to whom it Points to.

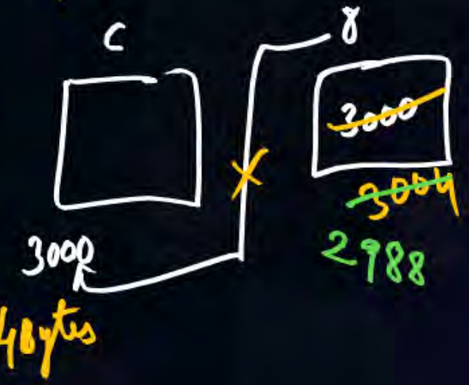
Let 2Bytes Per integer

Ex:

```
int a, *p;
char b, *q;
float c, *r;
p = &a;
q = &b;
r = &c;
```








```
P = P+2;  1000+2*2
          = 1004
q = q+3;  = 2001+3*1
          = 2004
r = r-4;
  = 3004 - 4*4
  = 2988
```

P ⇒ 1000
q ⇒ 2000
r ⇒ 3000

P++ ⇒ P+1
⇒ 1000+1*2
= 1002

q++ ⇒ q+1
⇒ 2000+1*1
⇒ 2001

r++ ⇒ r+1
⇒ 3000+1*4
⇒ 3004

2 mins Summary

- Pointer ?
- Declaration
- Initialization
- Void Pointer
- Pointer Type, Size
- Pointer Arithmetic

Types of Pointers

- Void Pointer
- NULL Pointer
- Dangling Pointer

Outdated

- near Pointer
- far Pointer
- Huge Pointer
- Wild Pointer

To be Contd...

THANK - YOU