

Pointers, Pass By Address, Array



C Programming

Trainer : Smita Kadam

Email ID : smita@sunbeaminfo.com

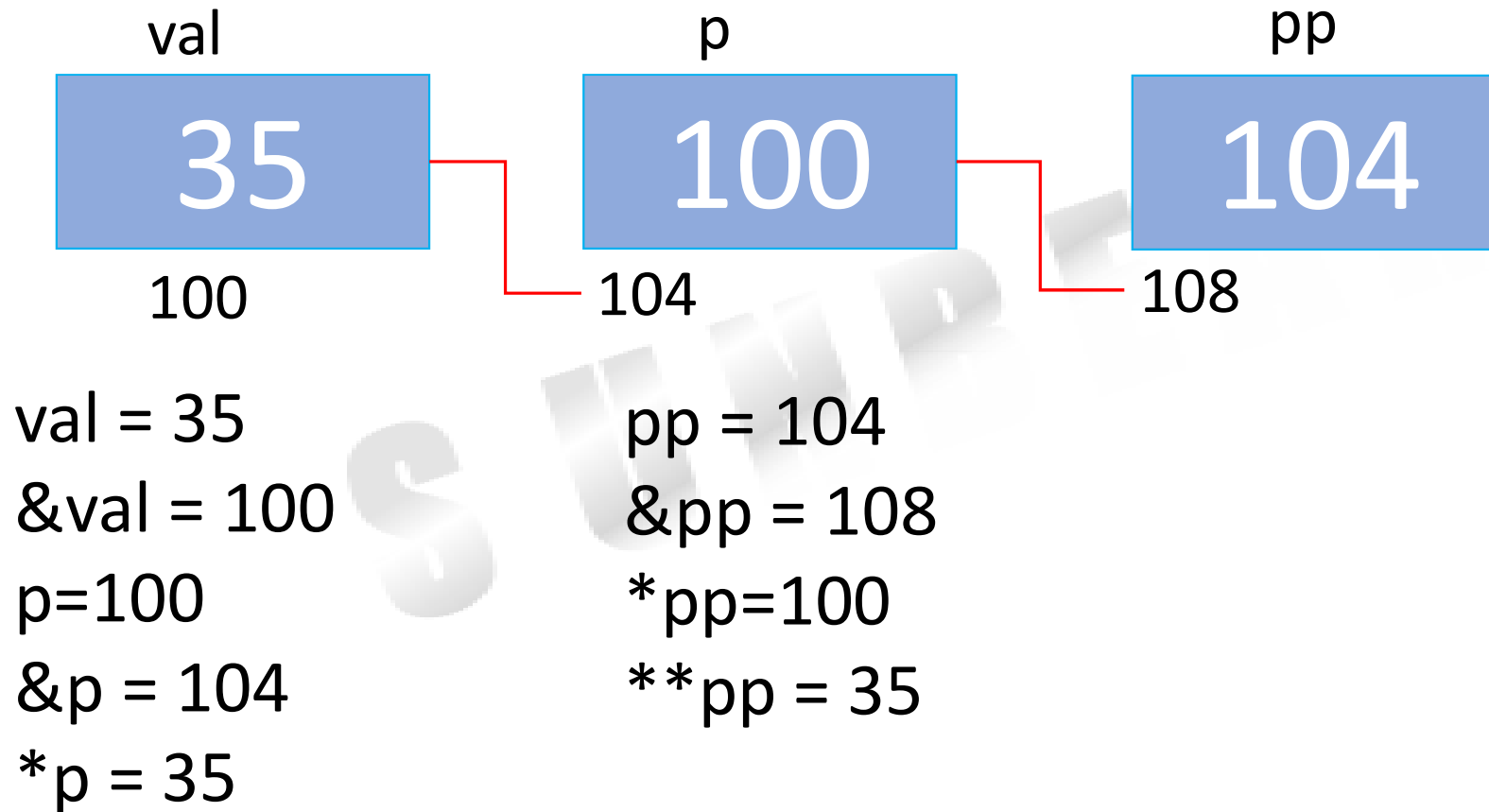


Pointer

- Derived data type.
- Pointer is a variable designed to store address of valid memory location.
- System provides address always in unsigned int format.
- Size of pointer to any type of location is equals unsigned int of respective compiler.
- We can declare pointer till n level of indirection.
- Operators used when you deal with pointers
 - * Value at operator / dereferencing operator / indirection operator
 - & Address Operator



Pointer



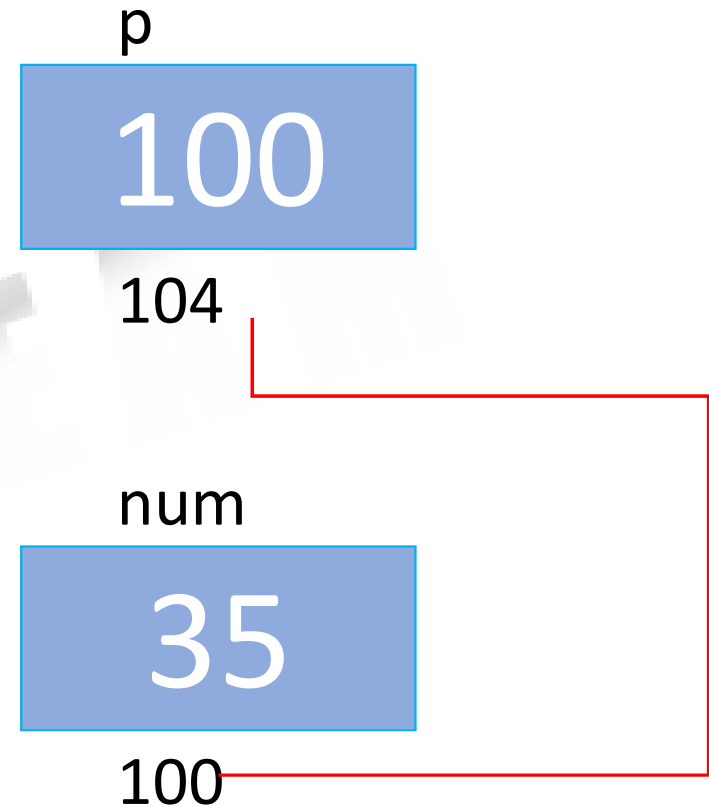
Pass By Address

```
void accept_number(int *p)
{
    printf("Specify number \n");
    scanf("%d",p);
}

p = 100
*p = 35

int main()
{
    int num;
    accept_number(&num);
    printf("Number = %d",num);
}

num = 35
&num = 100
```



Wild Pointer , NULL Pointer

Wild Pointer:

`int *p; // wild pointer a pointer which is not initialised at the time of declaration`

NULL Pointer:

`int *p = NULL; // pointer which is pointing to nothing`



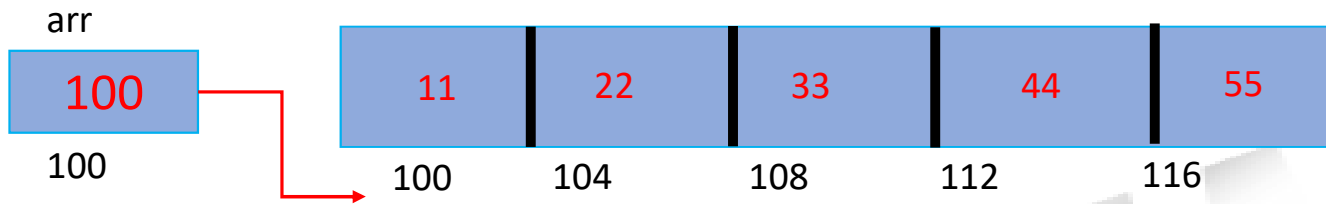
Array

- Derived data type...Can be implemented statically and dynamically.
- Collection of similar type of elements stored in consecutive memory block
- Identifies group of elements with common name but uniquely identified with index value.
- First element always receive index 0 and last element receives n-1 index.
- Array identified location stores address of first element.
- If array is implemented statically then such array can not be shrink or grown at runtime.
- We can declared n dimensions of array.
- It is compulsory to specify last dimension of any type of array except 1D array providing it is initialised at the time of declaration.
- If array is partially initialised at the time of declaration then rest elements are set to zero.
- Programmer will be responsible to handling array bounds.
- Array is always passed by address to function
- If programmer is aware of no. of elements to processed in advance then use static implementation
- Array's base address will be locked by system we can not modify it.
- Array identified location stores an address of first element hence array can be accessed using pointer notation



1D Array

```
int arr[5];
```



arr = 100

&arr = 100

***arr = 11**

arr + 1 = 104

***(arr+1) = 22**

***(1+arr) = 22**

arr[0] = 11

&arr[1] = 104

arr[1] = 22

1[arr] = 22



Pointer Arithmetic

- **Possible Operations:**

- We can add or subtract integer constant from pointer (address). No doubt we can apply increment / decrement operator on pointer(address). One operand as pointer(address) another operand as integer constant.
- We can subtract two pointers (Both operands as address type)

- **Not Possible Operations:**

- We can not add two pointers (addresses) (Both operands as address type)
- Multiplication/division on two pointers (addresses) is meaning less.





Thank you!

