Accessing the Variable's value through Pointer

Example

```
int quantity, *p, n;
quantity=179;
p=&quantity;
n= *p;
```

or `n=*&quantity` → `n=quantity`

* can be remembered as 'value at address'

DEMO: WAP to compute sum of three numbers using pointer.

CLASS ACTIVITY

Declare 4 variables and initialize them with float values. Define a pointer variable that will point to all variables. Display the memory address and values of all variables pointed by the pointer. [5 Minutes]

Declare 5 character pointers. Make all the pointers to point to the same variable called as ‘ch’ who is having a value assigned with ‘q’. Display the memory address of the variable ‘ch’, value of ch and also the address of the pointer variables. [5 Minutes]

Home Assignment

Write at least two program to solve problems using pointer

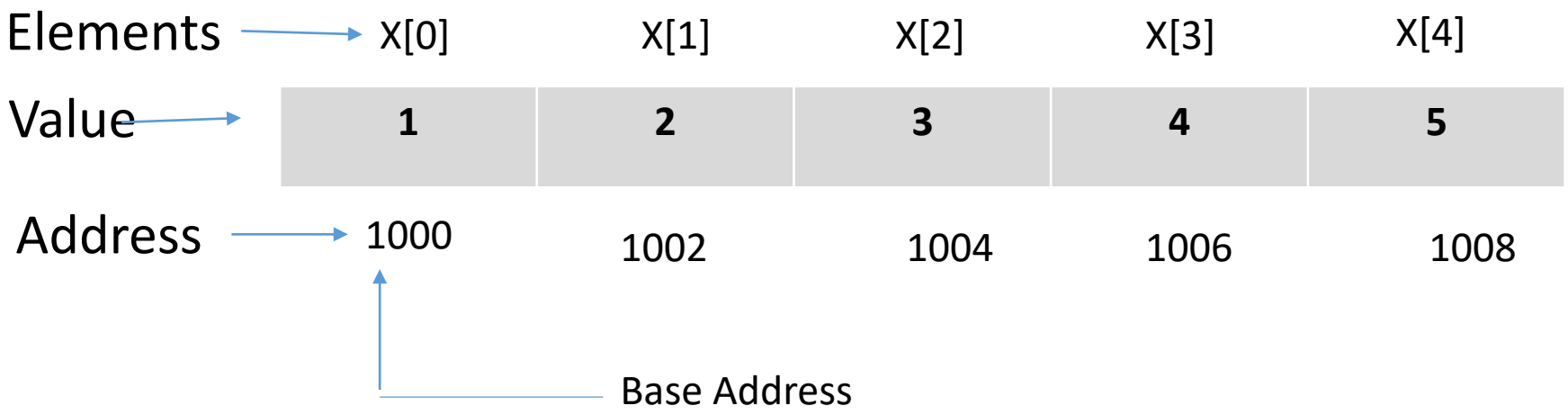
UNIT VI
POINTERS & ARRAY
PART 2

POINTERS AND ARRAY

- The pointers can be used to access the array elements
- And this pointer accessing method is considered to be much faster than array indexing
- When an array is declared, all the array elements are stored in contiguous memory locations
- The base address is the location of the first element (index 0) of the array
- The compiler defines the array name as a constant pointer to the first element

Suppose we declare an array **x**:

int x[5]={1,2,3,4,5};



- Suppose the base address of **x** is **1000** and assuming each integer requires two bytes, the five elements will be stored as shown in the diagram above.

POINTERS AND ARRAY

- The name *x* is defined as the constant pointer pointing to the first element of the array
- The value of *x* will be **1000**, the location of *x[0]*

```
x= &x[0]=1000;
```

- Pointer variable p can be assigned as follows

```
p= x;
```

- Which is equivalent to `p = &x[0];`
- Now, the remaining array element can be accessed as

```
p = &x[0];      (=1000)
p+1 = &x[1];    (=1002)
p+2 = &x[2];    (1004)
```

- Use of pointer to access array element is much faster than array indexing

```c
#include<stdio.h>
void main()
{
    int x[5]={1,2,3,4,5},*p,i;
    printf("\nArray Element   Array Value  Address Of Array Element  Address of a pointer");
    printf("\n*************  ************* ************************** ********************\n\n");
    for(i=0;i<5;i++)
    {
        p=&x[i];
        printf("\nx[%d]\t\t\t%d\t\t%u\t\t\t%u",i,*p,p,&p);
    }
    getch();
}
```

```
Array Element   Array Value  Address Of Array Element  Address of a pointer
*************  ************* ************************** ********************


x[0]                    1               1374772                 1374760
x[1]                    2               1374776                 1374760
x[2]                    3               1374780                 1374760
x[3]                    4               1374784                 1374760
x[4]                    5               1374788                 1374760_
```

- The relationship between pointer **p** and array **x** can be shown as:

p=&x[0]=1000

p+1=&x[1]=1002;

p+2=&x[2]=1004;

p+3=&x[3]=1006;

p+4=&x[4]=1008;

```
#include<stdio.h>
void main()
{
    int x[5]={1,2,3,4,5},*p;
    p=&x[0];
    printf("\n%d %u",*(p),p);
    printf("\n%d %u",*(p+1),(p+1));
    printf("\n%d %u",*(p+2),(p+2));
    printf("\n%d %u",*(p+3),(p+3));
    printf("\n%d %u",*(p+4),(p+4));
    getch();
}
```

```
1 3209080
2 3209084
3 3209088
4 3209092
5 3209096_
```

Ex

```
#include<stdio.h>
void main()
{
    int x[5]={1,2,3,4,5},*p,*p1;
p=x;
p1=&x[4];
printf("\nArray Element   Array Value  Address Of Array Element  Address of a ponter");
printf("\n*************  ************* ************************** **********************\n\n");
printf("\tx[0]\t\t%d\t\t%u\t\t\t%u",*p,p,&p);
p++;
printf("\n\tx[1]\t\t%d\t\t%u\t\t\t%u",*p,p,&p);
p++;
printf("\n\tx[2]\t\t%d\t\t%u\t\t\t%u",*p,p,&p);
p++;
printf("\n\tx[3]\t\t%d\t\t%u\t\t\t%u",*p,p,&p);
p++;
printf("\n\tx[4]\t\t%d\t\t%u\t\t\t%u",*p,p,&p);
p--;
printf("\n\tx[4]\t\t%d\t\t%u\t\t\t%u",*p,p,&p);

printf("\n\nThe sum of two pointers is=>%d",(*p+*p1));
```

```
Array Element   Array Value  Address Of Array Element  Address of
*************  ************* ************************** **********

        x[0]            1               3472136              34
        x[1]            2               3472140              34
        x[2]            3               3472144              34
        x[3]            4               3472148              34
        x[4]            5               3472152              34
        x[4]            4               3472148              34

The sum of two pointers is=>9
```