UNIT:2

Pointers

Topics To be Covered…

2.1 Introduction and Features of Pointers
2.2 Declaration of Pointer
2.3 Void Pointers
2.4 Array of Pointers
2.5 Pointers to Pointers

2.1 Introduction and Features of Pointers

- **Pointer is a variable which can hold the address of another variable.**
- A pointer is a derived data type.
- It contains memory addresses as their values.
- Pointers are used to manipulate data using the address of variables.
- The pointer accessing method is faster than array indexing.
- Dynamic memory allocation is done using pointers.

2.2 Declaration of Pointer

Declaration of pointer variable

Syntax:

```
Data-type    *pointer-name;
```

✓where, * tells that variable pointer-name is a pointer variable.
✓Pointer-name needs a memory location.
✓Pointer-name points to a variable of type data-type.

Example:

```
int  *p;
float  *q;
```

CONT…

Initialization of pointer variable

Syntax:

Pointer variable = &variable name;

Example:

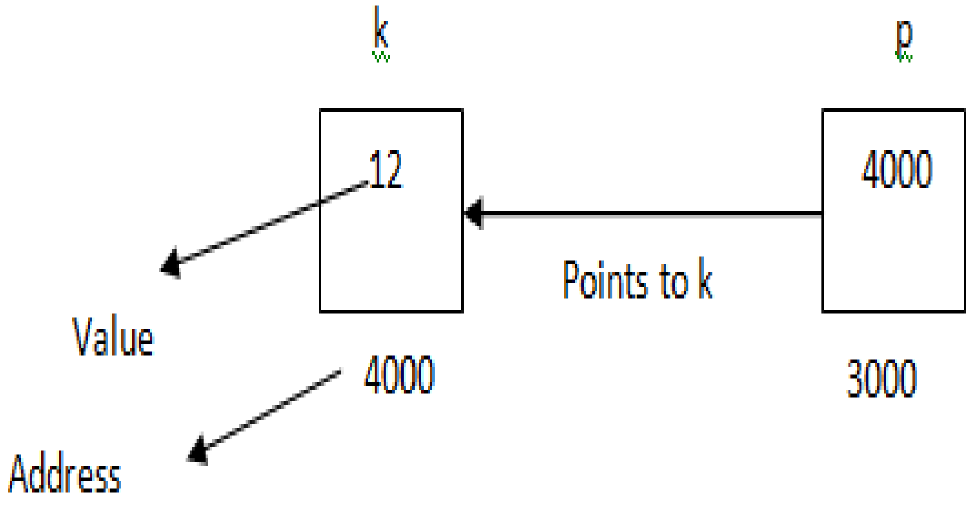
```
int k =10;
int *p;        // Declaration of pointer
```

p=&k;// Initialization of pointer

OR

```
int k = 10;
int *p=&k;
```

Cont…



CONT…

✓To access value of variable k, there are two ways.

printf (%d", k);

printf(" %d", *p);

✓ Two operators used with pointer:

*(Asterisk) Stands for "Contains at"

& (Ampersand) Stands for "Address of"

✓ %u Control string is used to print the address of the variable.

CONT…

Write a program to print address and content of variable using pointer.

```
#include<stdio.h>
#include<conio.h>
void main()
{
        int a,*ptr;
        clrscr( );
        a=100;
        printf(“\nAddress of variable a = %d”,&a);
        printf(“\nContent of variable a = %d”,a);
        printf(“\nAddress of variable a = %d”,ptr);
        printf(“\nContent of variable a = %d”,*ptr);
        getch( );
}
```

Cont…

Write a program to find the sum two variable using pointer.

Program:

```
#include<stdio.h>
#include<conio.h>
void main()
{
int c=0,a,b;
int *p,*q,*r;
printf("Enter a\n");
scanf("%d",&a);
printf("Enter b\n");
scanf("%d",&b);
```

CONT…

```
p=&a;
q=&b;
r=&c;
*r = *p + *q;
printf("Sum is %d",*r);
getch();
}
```

Output:=

```
Enter a
10
Enter b
2
Sum is:12
```

2.3 Void Pointers

- A Void pointer is pointer which has no specified data type.
- It does not have any data type.
- It is also known as Generic Pointer.
- The void pointer can be pointed to any type of variable.
- When void pointer is declared, two bytes of memory is assigned to it.

Syntax:

```
void *pointer_name;
```

Cont…

Example:

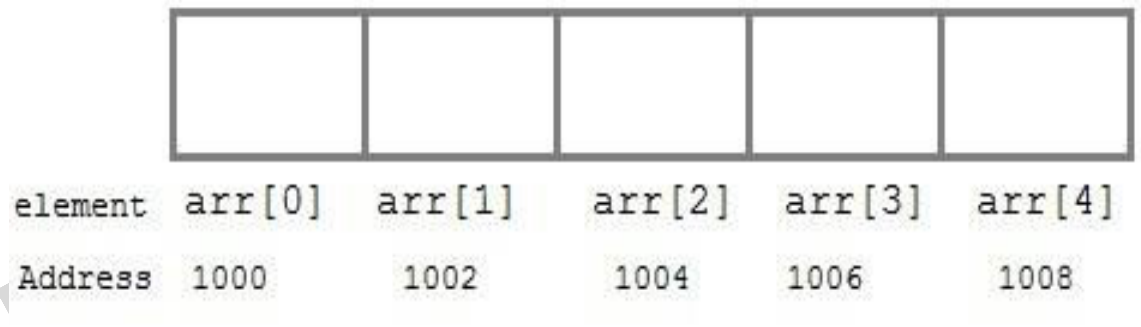
```
#include<stdio.h>
#include<conio.h>
void main()
{
        void *ptr;
        int a=10;
        float b=5.5;

        ptr=&a;
printf(%d", *(int *) ptr);      //before display, type casting the
pointer variable.
ptr=&b;
        printf(%f", *(float *) ptr);

getch();
}
```

2.4 Array of Pointers

- Pointer variable points to the first element of the array.
- Address of the next element is obtained by incrementing pointer variable by 1.
- Contains of the variable is obtained by *p.



CONT…

Example:

```
#include<stsio.h>
#include<conio.h>
void main()
{
            int a[5]={10,20,30,40,50};
            int *p;
            int i;
            p=a;    or   p= &a[0];
            printf("Array elements are:\n");
            for(i=0;i<5;i++)
            {
                  printf("%d \n",*p);
                  p++;
            }
            getch();
}
```

CONT…

Output: Array elements are:

```
        10
        20
        30
        40
        50
```

Cont…

W.A.P to find the sum of array elements using pointer.

```
#include<stsio.h>
#include<conio.h>
void main()
{
                int a[5]={10,20,30,40,50};
                int *p;
                int i,sum=0;
                p=a;
                for(i=0;i<5;i++)
                {
                        sum=sum+ *p
                        p++;
                }
                printf("Sum=%d",sum);
                getch();
}
```

CONT…

W.A.P to find the length of string using pointer.

Program:

```
#include<stdio.h>
#include<conio.h>
void main()
{
char name[10]={"hello"};
char *p;
clrscr();
int i,count=0;
p=name;
while(*p!='\0')
{
        count++;
        p++;
}
```

CONT…

```
printf("Length = %d",count);
getch();
}
```

Output: Length=5

2.5 Pointers to Pointers

- Pointers to pointers means pointer points to another pointer and it is called as chain of pointer or Double pointer.
- Double pointer indirectly points to the variable.
- The declaration of pointer to pointer is by ** sign before variable name and it always points to single pointer variable.

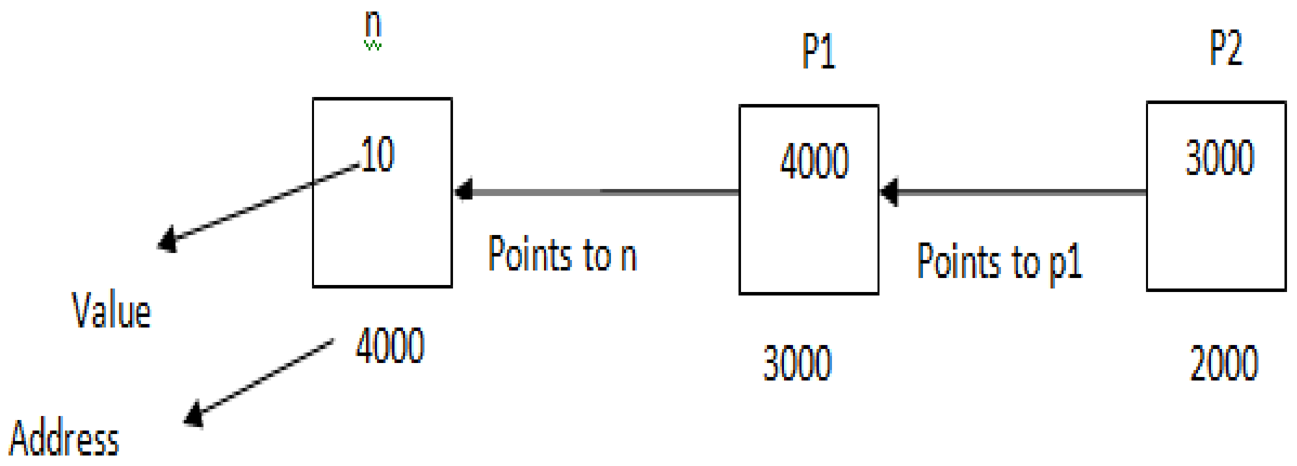
Syntax:

```
        Data-type  **pointer_name;
```

CONT…

Example:

```
int n,*p1,**p2;
n=10;
p1=&n;
p2=&p1;
```



Cont…

Program:

```
#include<stdio.h>
#include<conio.h>
void main ()
{
int n=123;
int *p1;
int **p2;

clrscr();
p1=&n;
p2=&p1;
printf("value of n is: %d\n",n);
printf("value of n is:%d\n",*p1);
printf("value of n is:%d",**p2);
getch();
}
```

Cont…

Output:=

```
value of n is: 123
value of n is: 123
value of n is: 123
```

2.6 ADVANTAGES OF POINTER.

- Using pointer, lines of code can be reduced.
- Pointer reduces the complexity of program because it makes data manipulation easy.
- Program execution speed is increase using pointer.
- Efficient use of memory is possible using pointer especially in array.
- Function can return more than one data using pointer.
- It can be used for manipulating data structures such as structures, linked list, queues, stacks and trees
- It can be used for dynamic memory management.