POINTER

A pointer is a variable that contains the address of the memory location of another variable. A pointer can point to a variable of different data types.

Example: int A → char B

Uses of Pointers

1. To return more than one value from a function
2. To pass array and string more conveniently from one function to another
3. To allocate memory and access it
4. To manipulate arrays easily by moving pointers to them instead of moving arrays itself
5. Addition and Subtraction are the only operations that can be performed on pointers

Declaration

Data type and a variable name preceded by asterisk*

Syntax: int *school;

POINTERS OPERATORS

Recall: Operands are used with operator to form expression

A * B = C

- A, B, C are Operands
- *, =, are Operators

There are two special types of operator in POINTER

1. The & operator
2. The * operator

& Operator: returns the memory address of the operand

The * Operator: returns the value in the address of the memory location

POINTER AND ARRAY

The address of the array can be expressed in two ways:

1. By writing the actual array element preceded by the ampersand sign (&)

2. By writing an expression in which the subscript is added to the array name

HOW WE CAN USE POINTERS

- POINTERS AND MULTIDIMENSIONAL ARRAY

Declaration:

```
DATA TYPE (*var) [expre2];
```

- POINTERS AND STRINGS

See the example on the dev++

ALLOCATING MEMORY IN C

- Malloc function: is one of the most commonly used functions. It permits the allocation of memory from the pool of C memory

The Malloc () parameter: is an Integer that specifies the number of byte needed

Syntax:

```
#Include<malloc.h>

Int p

P = (int*) malloc (n*sizeof (int)) ;
```

- FREE ()

The free function is use to de-allocate memory that is no longer in use

Declaration

```
Void free (void *ptr);
```

- Calloc ()

The Calloc function: allocates the requested memory and returns a pointer to it. The difference in **malloc** and **calloc** is that malloc does not set the memory to zero where as calloc sets allocated memory to zero.

Declaration:

void *calloc(size_t nitems, size_t size)

nitems – This is the number of elements to be allocated.

size – This is the size of elements.

- Realloc ()

Realloc function: Is used to allocate more bytes to memory without loosing data

Syntax:

void *realloc(void *ptr, size_t size)

Parameters

- **ptr – This is the pointer to a memory block previously allocated with malloc, calloc or realloc to be reallocated.**
- **size – This is the new size for the memory block, in bytes.**