**1. Declaration of Pointer Variables**

* **Syntax**:  
  data\_type \*pointer\_name; Example: int \*ptr;
* **Description**:
  + A pointer is a variable that stores the memory address of another variable.
  + The \* is used to define a pointer variable, which points to a specific data type (e.g., int, char).
  + Example: If int \*p is declared, p can store the address of an integer variable.

**2. Pointer Arithmetic**

* **Operations**:
  + **Increment (++)**: Moves the pointer to the next memory location (depends on the data type size).  
    Example: If p points to an integer, p++ moves to the next integer's address.
  + **Decrement (--)**: Moves the pointer to the previous memory location.
  + **Addition/Subtraction**:  
    ptr = ptr + n; moves ptr by n positions in memory.
  + **Pointer Difference**:  
    Subtracting two pointers gives the number of elements between them.
* **Key Point**: Pointer arithmetic works by considering the size of the data type the pointer points to (e.g., sizeof(int) is 4 bytes).

**3. Returning Multiple Output Values Through a Function Using Pointers**

* **Concept**:  
  C functions can return only one value directly, but pointers can be used to modify multiple variables outside the function.
* **Steps**:
  + Pass pointers to the function arguments.
  + Modify the values of the variables using dereferencing (\*).
* **Example**:

c

void swap(int \*x, int \*y) {

int temp = \*x;

\*x = \*y;

\*y = temp;

}

Calling this function swaps two values.

**4. Strings**

* **Declaration**:  
  char str[50]; (Fixed size)  
  char \*str; (Dynamic size using pointers)
* **Definition**:
  + Strings are arrays of characters ending with a null character \0.
* **Input/Output**:
  + scanf("%s", str); for input.
  + printf("%s", str); for output.
* **Pointer and String**:  
  A string can be manipulated using pointers. Example: char \*ptr = str; allows traversing the string with pointer arithmetic.
* **String Operations**:  
  Common string functions include strlen(), strcpy(), strcmp(), and strcat().