# The University of Azad Jammu and Kashmir

Name: Zarnab Fatima

Roll NO: 2024-SE-14

Course Title: DSA

Course Code: CS-2101

Submitted To: Engr. Sidra Rafique

*Department of Software Engineering*

# LAB 02

## Pointers in Arrays

In programming, pointers are variables that store memory addresses. When working with arrays, pointers can be used to access and manipulate array elements.

## Key Concepts:

**Pointer Arithmetic:** Pointers can be incremented or decremented to point to different elements in the array.

**Dereferencing:** The dereference operator (*) is used to access the value stored at the memory address pointed to by a pointer.

**Array Name as a Pointer:** The name of an array is equivalent to a pointer to the first element of the array.

## Advantages:

**1**. **Efficient Memory Access:** Pointers allow for direct access to memory locations, making it efficient for array operations.

**2. Dynamic Memory Allocation:** Pointers are essential for dynamic memory allocation, which is useful for arrays with dynamic sizes.

## Disadvantages:

**1. Error-Prone:** Pointer arithmetic and dereferencing can lead to errors if not used correctly.

**2. Memory Leaks:** Improper use of pointers can result in memory leaks or dangling pointers.

## Code:

```cpp
#include<iostream>
using namespace std;
int main()
{
    int num[5];                              //Declare anarray of 5 integer
    int *ptr=num;                            // Pointer pointing to the first element of array
    int sum=0;
    int max;
    cout<<"Enter 5 Numbers"<<endl;           //Input using Pointer
    for(int i=0;i<5;i++)
    {
        cout<<"Number"<<i+1<<":";
        cin>>*(ptr+i);                       //Using pointer arithmetic to access array element
    }
    max=*ptr;                                //Initialize max with the first element
    for(int i=0;i<5;i++)                     //calculate sum and find maximum using pointer
    {
        sum+=*(ptr+i);                       //Add value at address (ptr+i)
        if(*(ptr+i)>max)
        max=*(ptr+i);
    }
    cout<<"\nYou entered:" ;                 //Display result
    for(int i=0;i<5;i++)
    {
        cout<<*(ptr+i)<<" ";
    }
    cout<<"\nSum of numbers= "<<sum;
    cout<<"\nMaximum number= "<<max<<endl;
    return 0;
}
```

## Output:

```
Enter 5 Numbers
Number1:34
Number2:56
Number3:367
Number4:23
Number5:65

You entered:34 56 367 23 65
Sum of numbers= 545
Maximum number= 367

--------------------------------
Process exited after 32.84 seconds with return value 0
Press any key to continue . . .
```

***Thank You***