# CSE 122 : Object Oriented Programming Lab
# Lab - 4

Intake 52
**Section** - 3

January 25, 2024

## Task 1

- Declare an integer array of size 5.
- Initialize the array elements with some integer values.
- Declare a pointer variable of type integer and assign the address of the first element of the array to the pointer.
- Use a loop to traverse the array using the pointer and display the elements.
- Calculate the sum of the array elements using the pointer.
- Display the sum on the console.

## Task 2

- Define a class called **Person** with private member variables like name and age, and public member functions to set and display the values.
- Declare an array of **Person** objects, with a size of 4.
- Use a loop to input the values for each Person object using the set function.
- Declare a pointer variable of type Person and assign the address of the first element of the array to the pointer.
- Use another loop to display the values of each Person object using the pointer.
- Increment the pointer to point to the next element in the array and display its values.

## Task 3

- Create two classes, **X** and **Y**, with private member variables **data1** and **data2**, respectively. Both classes should have public member functions to set and get the values of their respective data members.
- Implement a **friend** function named **sum** that takes an object of class X and an object of class Y as parameters. The sum function should calculate and return **the sum of data1 from object X and data2 from object Y**.
- Implement another friend function named **exchange** that takes references to an object of class X and an object of class Y as parameters. The **exchange** function should swap the values of data1 from object X and data2 from object Y using a temporary variable.

- In the main function, create objects of classes **X** and **Y** named x1 and y1, respectively. Set the values of data1 and data2 to any desired integer values using the member functions.
- Display the result of calling the sum function, passing x1 and y1 as arguments.
- Call the **exchange** function, passing x1 and y1 as arguments.
- Display the updated values of data1 and data2 by calling the respective getter functions of x1 and y1.