

Array as a Data Structure

Lesson 2.3

Learning Objectives

- LO 2.3.1 **Utilize** static and dynamic arrays as an efficient storage medium of data
- LO 2.3.2 **Solve** computing problems effectively using static/dynamic arrays

Arrays

- Array is a bound collection of elements with the same data type
- Two types of arrays:
 1. Static array – allocated in the *stack* memory
 2. Dynamic array – allocated in the *heap* memory

Static vs. Dynamic Array

	Strength	Weakness
Static Array	Very low memory footprint	Fixed memory size
Dynamic Array	Size can be determined at runtime Can be resized	Memory must be managed to not cause memory leak

Strengthening the Learning Objectives

- LO 2.3.1 Utilize static and dynamic arrays as an efficient storage medium of data**
- LO 2.3.2 Solve computing problems effectively using static/dynamic arrays**

Example:

Create a function, `findEven`, that accepts an array of integers, `arr`, and return an array containing non-negative even integer elements of `arr`.

LO 2.3.1 Utilize static and dynamic arrays as an efficient storage medium of data

LO 2.3.2 Solve computing problems effectively using static/dynamic arrays

Example:

Implement the function:

double computeMedian(double* data, int length)

that returns the median of all elements in data.

- LO 2.3.1 Utilize static and dynamic arrays as an efficient storage medium of data**
- LO 2.3.2 Solve computing problems effectively using static/dynamic arrays**

Example:

Create a function, `generateHintMap`, that accepts an array of zeroes (0) and ones (1), `field`, where 1 represent the mines for the Minesweeper game and 0 is a safe space. The function must return an array of integers containing the number of mines around a specific location (*including diagonal*). If the location contains a mine, assign it with -1.

