Static Arrays

Lesson 2.1

Learning Objectives

LO 2.1.1 **Create** and **initialize** static arrays

LO 2.1.2 **Access** data in licit locations of the static array

Static Array Declaration

- Prototype:
  `<data type> <variable name>[<const int>];`
- Example:
  `float grades[5];`
- On what memory is this allocated?
  - Stack memory
  - Contiguously allocated
- The initial values of a recently declared static array are based on the default value of the data type.

Initializing the Static Array

- Prototype:
  `<variable name>[<index>] = <value>;`
- Example:
  `grades[0] = 94.35;`
- During assignment, the declared type of the variable name should be compatible with the type of the value assigned.

Accessing Elements in the Static Array

- Prototype:
  `<variable name>[<index>]`
- Example:
  `grades[2]`
- After accessing, you can use it in an expression as if the statement (*example*) is the value itself.
- Example:
  `float ave = (grades[0]+grades[1])/2;`

Strengthening the Learning Objectives

LO 2.1.1 Create and initialize static arrays
LO 2.1.2 Access data in licit locations of the static array

Example:

1. Create a static array named `weights` containing 10 `double` values.
2. Initialize a random value between 0 and 10 for all elements of `weights`.
3. Display all values of `weights`.
4. Display all values of `weights` greater than the average of all `weight` values.

LO 2.1.1 Create and initialize static arrays
LO 2.1.2 Access data in licit locations of the static array

Example:

1. Create a static array named `arr` containing 7 integer values.
2. Initialize the values of `arr` with 10, 6, 3, 8, 5, 1, 2, respectively.
3. Create a static array named `cumulative` containing 7 integer values then assign it with the cumulative sum of the values in `arr`.
4. Display the values of `arr` and `cumulative`.