Lesson 6: Arrays

6.0. Introduction
In our previous lesson, we learnt about program control structures. Today we will learn about arrays. An array is a secondary data type that stores data of the same type in contiguous memory locations. The elements in the array are easier to access because they are stored in contiguous memory locations and are of the same type.

6.1. Lesson objectives
By the end of this lesson, you will be able to:
- Define an array
- Declare and initialize array
- Discuss different types of arrays
- Describe advantages of arrays
- Describe disadvantages of arrays
- Write programs that use arrays

6.2. Lesson outline
This lesson is structured as follows:

6.3. Definition
An array is a secondary data type that stores data of the same type in contiguous memory locations. It stores collection of elements.

6.4. Declaration of arrays
An array can be declared using the syntax below:
Type Identifier[size];
Where;
Type is any valid data type, identifier is any valid variable name and size is the total number of elements contained in the array.
Examples:

int nums[10]; // array of ten elements of
type integer cha gender[2]; // array of
two members of type character

6.5. Initialization of arrays

Like any other variable(s) an array can be initialized to assign its elements
the initial value(s). This can be done in the following ways:

    a)   Declare and initialize in the same
    statement Example:
    int nums[3]={4,8,9};
    b)   Declare and initialize in different
    the same statement Example: char
    grade[4]; grade[0]='a'; grade[1]='b';
    grade[2]='c'; grade[3]='d';
    c)   Declare and initialize the array in
    the same stamen but without the array
    size. Example int nums[]={4,8,9};

6.6. Accessing arrays

An element is accessed by indexing the array name. This is done by placing
the index of the element within square brackets after the name of the array.
For example:

double salary = balance[9];

The above statement will take 10th element from the array and assign the
value to salary variable.

6.7. Types of arrays

Arrays can be classified into the following types:

    a) Based on memory allocation:
        • Static arrays
        • Dynamic arrays
    b) Based on dimensions
        • 1-dimensional
        • Multi dimensional

6.8. Advantages of arrays

    • Easy to access
    • Do not need to declare several variables of same type.

6.9. Disadvantages of arrays

    • Wastage of memory
    • How to use them

6.10. A program that uses array

The following program demonstrates the concepts learnt

```
#include <iostream>
using namespace std;
void main()
{
    int n[10]; /* n is an array of 10 integers */
    int i; /*i is the counter or index*/
    for (i = 0; i < 10; i++) //loop to input values into the array
    {
        n[i] = i + 100; /* set element at location i to i + 100 */
    }
    /* output each array element's value */
    for (i = 0; i< 10; i++) //loop to print array values
    {
        cout << "Element at " << i << "  = " << n[i] << endl;
    }
}
```

6.11. Revision questions
a) Define an array
b) Describe how you can initialize an array
c) Write a program that computes the average of 20 double values using arrays

6.12. Summary
In this lesson you have learnt about arrays: how to declare, initialize and use arrays in a program.

613. Suggested reading
[1]. Object oriented programming with C++ by E Balagurusamy 3rd ed; publisher: Tata Mcraw Hill

[2]. Sams teach yourself c++ in 24 hours by Jesse Liberty and Rogers Cadenhead. [3]. Object oriented programming in c++ by Joyce Farrel [4]. Object-oriented programming with c++ by Sourav Sahay.