



Wollo University, Kombolcha

Institute of Technology

College of Informatics

Department of Information System

By Daniel G.

Mar/2021 G.C


A vertical cluster of hexagonal icons in various shades of blue and cyan. From top to bottom, the icons include: a lightbulb, a thumbs-up, a network of nodes, a smartphone, a magnifying glass, a gear, and a speech bubble.

INSY2031

Fundamentals of Programming II



Contents

- ◇ Homogeneous and Heterogeneous Data types.
 - ◇ Difference between Arrays and Structures
 - ◇ Declaring , accessing and programming arrays and structure
 - ◇ Multidimensional Arrays
 - ◇ User Defined Data Types (UTD)
- 



Homogeneous and Heterogeneous Data types.

1

The homogeneous data structures are the one in which the data elements have the same data type. All the data elements in the homogeneous belongs to the single data type.

For example: Arrays => `int[]`, `char[]`, `bool[]`



Homogeneous and Heterogeneous Data types.

1

The Heterogeneous data structures are the one in which the data elements doesn't belong to the same data type. All the data elements have different data type.

For example: classes and Structure;

```
Car ford = new Car();  
ford.model = FD1231;  
ford.product_year = 1998;  
ford.price = 432.12;
```

A decorative pattern of hexagons in various shades of blue and cyan on the left side of the slide. Some hexagons contain icons: a lightbulb, a thumbs up, a network of nodes, a smartphone, a magnifying glass, a gear, and a speech bubble.

2

Difference between Arrays and Structures

A decorative pattern of hexagons in various shades of blue and cyan. Some hexagons contain icons: a lightbulb, a thumbs up, a network node, a smartphone, a magnifying glass, a gear, and a speech bubble.

2.1

Array

An array is a collection of elements of the same type placed in contiguous memory locations that can be individually referenced by using an index to a unique identifier



2.1

Arrays are used to store multiple values in a single variable, instead of declaring separate variables for each value. To declare an array, define the variable type, specify the name of the array followed by square brackets and specify the number of elements it should store.



Array can be one dimensional and multidimensional

One dimensional

- ❖ Simplest form of an array in C++ language.
- ❖ Easily declare, initialize, and manipulate

```
dataType arrayName[arraySize];
```



Array is a collection of elements of the same type.

```
char vowels[5] = ['A','I','O','U','E'];
```

```
cout << vowels[0]; // A
```

```
cout << vowels[3]; // U
```

```
cout << vowels[4]; // E
```

```
Cout << sizeof(vowels) // 5;
```

```
double grade[27];
```

Here, `grade` is an array that can hold a maximum of 27 elements of `double` type.



```
char vowels[5] = ['A','I','O','U','E'];
```

```
for(int i = 0; i < sizeof(vowels); i++){  
    cout << vowels[i] << endl;
```

```
}
```

```
//output
```

```
A
```

```
I
```

```
O
```

```
U
```

```
E
```

Multidimensional Array

- ❖ Is an array of an array.
- ❖ Rectangular arrays
- ❖ Data is stored in tabular form

```
char vowels[2][5] = [['A','I','O','U','E'], ['a','i','o','u','e']];
```

```
cout << vowels[0][0]; // A
```

```
cout << vowels[0][1]; // ?
```

```
cout << vowels[1][0]; // ?
```

```
Cout >> vowels[1][4]; // ?
```

```
type name[rows][columns];
```

//the above example can be read as collection of 2 arrays with size of 5;



```
char vowels[2][5] = [['A','I','O','U','E'], ['a','i','o','u','e']];
```


```
for(int i = 0; i < 2; i++){           // for rows
    for(int j = 0; j < 5; j++){       // for columns
        cout << vowels[i][j];
    }
};
```



2.2

Structures

Is a C++ data structure that can be used to store together elements of different data types. In C++, a structure is a user-defined data type. The structure creates a data type for grouping items of different data types under a single data type



Structure is used for storing an aggregation of elements.

```
Struct CafeMealCard{  
    string studentName;  
    int year;  
    bool isBanned;  
}  
CafeMealCard card1 = { "Alemtsehay", 2013, false};  
  
string name = card1.studentName; // Alemtsehay
```



User Defined Data Types in C++

Eg. Structure, Array, typedef, enumeration



Typedef

define new data type names to the existing ones.

```
typedef <type> <newname>;  
typedef long long ll;
```



Enumeration

It is defined as a set of named integer constants that specify all the possible values a variable of that type can have.

“

```
enum week_days {sun, mon, tues, wed, thur, fri, sat = 9};
```

```
week_days d;
```

```
week_days = x;
```

```
d = mon;
```

```
x = sat;
```

```
cout << d; // 1
```

```
cout << x; // 9
```



END
OF
CHAPTER





Thanks!

Any questions?

Feel free!

