STRUCTURES

A structure consist of data Items of different data types grouped together.

Variables in structure are called structure elements or members

Structure elements are reference through the use of dot (.) operators also known as membership operator

Syntax

- **Structure_name.element_name**

Example

- **("%s, Opeyemi.students_name);**

INITIALIZING STRUCTURES

Structure variables are initialized at the point of declaration

- **Struct students**
- **{**
- **Int no;**
- **Char name [200];**
- **};**
- **Struct students stud1 = { 24, "Opeyemi"};**
- **Struct students stud2 = { 243, "Opethaiwoh"};**

Stud1, stud2 are the variables of the type students

It is possible to assign the values of one structured variable to another variable of the same type using an assignment operator

- **Example**
- **Book1 = book2;**

OR

You can use a built-in function called memcpy()

- **Syntax:**
- **Memcpy(char*destination, char &source, int nbytes);**
- **Example**
- **Memcpy(&book1, &book2, sizeof(struct cat));**

It is possible to have structure within structure but a structure cannot be nested by itself

A structure variable can be passed as an argument to a function. The type of argument should pass the type of parameter

A common use of structure is in Array of structure. A structure is first define and then array variable of that type is declared.

Example

- **Struct cat books[50];**

Structure arrays are initialized by enclosing the list of values of its element within a pair of braces

Example

- **Struct Opeyemi**
- **{**
- **Char ch;**
- **Int I;**
- **};**
- **Struct Opeyemi new_series [5] =**
- **{**
- **{'o', 300}**
- **{'p', 400}**
- **{'k', 300}**
- **{'i', 300}**
- **{'u', 300}**
- **};**

POINTERS TO STRUCTURE

Structures are declared by placing asterisk * in front of structure variable name

The → operator is use to access the elements of a structure using a pointer

Examples

- **Struct cat *ptr_bk;**
- **Ptr_bk = &books;**
- **Printf ("%s", ptr_brk→ author);**

THE TYPEDEF KEYWORD

A new data type can be defined by using a keyword typedef

It does not create a new data types but defines a new name for an existing data type

Syntax

- **Typedef type name;**

Example

- **Typedef char opeyemi**