***Estructuras***

Una **estructura** contiene varios datos ya que su función es agrupar datos de diferente o

igual tipo o de manejar datos que serían muy difíciles de describir en los tipos de datos

primitivos. La forma de definir una estructura es haciendo uso de la palabra clave

struct. Los componentes individuales de una estructura se llaman **miembros**, y la

estructura puede contener cualquier número de miembros con un nombre único.

struct nombre\_estructura { tipo\_dato miembros\_de\_estructura };

struct MiEstructura{

int variable1;

float varaible2;

char\* variable3;

};

La declaración especifica simplemente el nombre y el formato de la estructura de datos,

pero no reserva almacenamiento en la memoria, para la definición de una variable (crear

un área en memoria) en donde los datos se almacenan de acuerdo al formato

estructurado se pueden definir de dos formas:

Definición:

A: listándolas inmediatamente después de la llave de cierre en la declaración

struct nombre\_estructura { tipo\_dato miembros\_de\_estructura }**variable**;

struct MiEstructura{

int variable1;

float variable2;

char\* variable3;

}var1, var2;

B: listando el tipo de la estructura creado, en cualquier lugar del programa

struct nombre\_estructura nombre\_de\_variable;

struct MiEstructura var3;

Para dar valores iniciales a una estructura se utilizan llaves al momento de la definición,

sin importar por cuál de las dos definiciones se haya optado pero respetando el orden en

que fueron declarados los miembros de la estructura.

Inicializar:

struct nombre\_estructura nombre\_variable\_estuctura = {valor1, valor2, valor3};

struct MiEstructura var4 = {4, 7.2, “Juan”};

Para tener acceso a los miembros de una estructura se usa el operador punto.

Acceso:

nombre\_variable\_estuctura.nombre\_miembro

var4.variable2 = 38.51;