Victor Kneider. V-30.167.855 30-05-22

ASIGNACIÓN ESTÁTICA VS DINÁMICA DE MEMORIA

- **Asignación Estática** = La asignación estática de memoria es la manejada por nosotros hasta la fecha, se conoce como estática debido a que es asignada para propositos especificos durante el tiempo de compilación y no puede ser liberada sino hasta la finalización del proceso o algoritmo. Ej:

```
int x=0; → Se asignan 4 bytes estáticos al compilar
```

- **Asignación Dinámica** = Se conoce como dinámica ya que su estado es dinámico (cambiante) y puede ser asignada o liberada durante el tiempo de ejecución.

En lenguaje C la asignación dinámica de memoria se realiza con las funciones malloc (para asignación) y free (para liberar memoria). Posteriormente con el lenguaje C++ nacen los operadores propios de dicho lenguaje para la asignación dinámica de memoria los cuales son: new e int

Scribe

• **Operador New =** El operador new se encarga de asignar memoria dinámica durante la ejecución del programa dependiendo del tipo de dato que se busque asignar:

*SINTAXIS = Variable puntero = new TIPO_DE_DATO

-ARREGLO= Variable puntero= new tipo_de_dato [nro_de_elementos]

- Puede ser inicializada una variable en la misma declaración del new (=)

* A diferencia de la función malloc en la cual el programador debe de indicar la cantidad de memoria a ser asignada dinamicamente, el operador new se encarga de asignar la cantidad de memoria requerida segun el tipo de dato que sea indicado

• **Operador delete =** El operador delete se encarga de liberar la memoria la cual fue asignada dinamicamente a través del operador new

*SINTAXIS = delete variable_puntero

-ARREGLO= delete []variable_puntero

* Solo se puede utilizar el operador delete en variables punteros que fueron inicializadas con el operador new.

// Programa para calcular el error cuadrado entre mediciones teóricas y experimentales

CÓDIGO EJEMPLO

Scribe

```
#include <stdio.h>

int main (){
int cantidad =0;
printf(" Ingrese la cantidad de mediciones realizadas : );
scanf( "%.i", &cantidad);

float *Teorica = new float [cantidad] ;
float *Experimental = new float [cantidad] ;
float *ERROR = new float [cantidad];
for (int i=0; i<cantidad; i++){
   printf ("\nIngrese la medición teórica nro [%.i]:  ", i);
   scanf ( "%.f", & teorica [i]); }

for (int i=0; i< cantidad; i ++){
   printf("\n Ingrese la medida experimental nro[%.i] : ", i);
   scanf ("%.f, & experimental [i]); }

for (int i=0; i<cantidad; i++){
   error [i] = (experimetal [i]- teorico [i]) * (experimental [i] - teorico [i]);
   printf ("\nError cuadrado medición [%.i] : [%.i]", i, error [i]);}

delete [] teorica;
delete [] experimental;
delete [] error;
return 0; }
```