Programación orientada a objetos - Informática II

Clase 1 – Introducción a C++

¡Hola! ¿Cómo están? Continuamos bajo la modalidad en línea de Informática II y esta semana comenzaremos sobre una nueva unidad donde introduciremos no solo un nuevo lenguaje sino un nuevo paradigma de programación.

Introducción a C++

C++ fue creado en los Laboratorios Bell por Bjarne Stroustrup e inicialmente lo denominó C con clases. El nombre del lenguaje fue una denominación basada en una mejora del C. C++ se propuso mejorar muchas de las características del lenguaje que vimos en Informática I y agregó un nuevo paradigma de programación con el objetivo de incrementar la productividad, calidad y reutilización del software.

A continuación, veremos un pequeño programa en C, su desarrollo en C++ , analizando diferencias y similitudes:

Les proponemos un ejemplo de ingreso de datos donde se manejen los flujos de entrada y salida estándar:

```
#include <stdio.h>

int main(void)
{
	int valor;

	printf("Ingrese un valor: ");
	scanf("%d", &valor);
	printf("El valor ingresado fue: %d\n", valor);

	return 0;
}
```

En este primer caso, incluimos la directiva del preprocesador para sumar el encabezado "stdio.h", a continuación comienza nuestra función *main* , en la cual se declara una variable entera, se le solicita al usuario que cargue un dato y ese dato cargado se almacena en la dirección de memoria indicada. Por último se muestra por pantalla el valor ingresado junto a una leyenda.

Veamos ahora su equivalente en C++

```
#include <iostream>

int main(void)
{
	int valor;

	std::out << "Ingrese un valor: ";
	std::cin >> valor ;

	std::out << "El valor ingresado fue: " << valor << endl ;

	return 0;
}
```

En este segundo caso:

- Utilizamos *iostream* que es el encabezado para el manejo de flujos de entrada y salida.

- Comienza nuestra función *main* , en la cual se declara una variable entera. Si bien la declaramos en el mismo lugar en C++ , las variables pueden declararse en distintos puntos de nuestro programa, muchos en Informática I tenían este vicio y esperemos que con el tiempo se haya corregido!

- A través de *std::cout* se controla el flujo de salida y se le solicita al usuario que cargue un dato.

- Luego, mediante std::cin se carga ese datos como contenido de la variable **valor**

- Por último mediante std::cout se muestra por pantalla el valor ingresado junto a una leyenda.

La notación *std::cout* nos muestra que existe un **espacio de nombres** llamado *std* que posee diferentes "*recursos"* como *cout* y *cin.* De a poco iremos viendo que significan estas nuevas formas de escribir y como las podemos utilizar.

¿Qué cosas tienen en común y qué diferencias hay entre C y C++?

Cuando uno aprende un nuevo lenguaje siempre intenta apoyarse en cosas que ya conoce, es por ello que veremos algunos elementos comunes entre las diferentes sintaxis:

- **Declaración de variables:** los tipos de datos simples, los seguimos manejando de la misma forma. Tal como se comentó previamente en C++ podemos declarar las variables "donde nos guste" sin embargo, es mas desprolijo y poco ordenado. Trataremos de declarar las variables en función del alcance o contexto donde queramos utilizarlas.

- **Comentarios:** ya sabíamos hacer comentarios con /* */ para multiples líneas y algunos ya utilizaban // para realizar comentarios de una línea.

- **Estructuras de selección:** if / if-else / switch-case son estructuras conocidas y las podemos seguir utilizando

- **Estructuras de repetición y control:** for, while , do-while ya las sabemos utilizar y conocemos sus características, ventajas y desventajas, así que las seguiremos utilizando.

- **Constantes:** en C utilizábamos un *define* como directiva para un valor que se repetía en diferentes momentos de nuestro código. Aquí trabajaremos con el operador *const* y declararemos variables constantes (*const int a = 5 ;*). Es decir, una variable declarada e inicializada que no cambia su valor a lo largo de nuestro programa.

- **Pasaje por valor y por referencia:** el manejo de punteros es un tema crítico y diferente entre un lenguaje y otro, razón por la cual, veremos que se manejan con referencias, pero para esta primera clase nos interesa que sepan que se maneja diferente (por ahora nada más).

- **Argumentos de una función:** hasta ahora, salvo casos muy excepcionales las funciones se llamaban de una sola manera, la función *suma* recibía dos enteros y devolvía un entero. En C++ podremos *sobrecargar* a una función *suma* para que admita diferentes argumentos y utilizar la **mejor versión de la función "*suma*"** en función del contexto donde la invoquemos y nuestra necesidad.

- **Manejo de memoria:** en C nos acostumbramos a utilizar *malloc, realloc y free*. Aquí manejaremos otro grupo de funciones: *new* y *delete.* Veremos que su uso evita el casteo a diferentes tipos de datos, el goteo de memoria y el cálculo de cuanta memoria solicitar y liberar. Poco a poco iremos viendo algunos ejemplos al respecto.

Final de esta clase

Bueno, este es el final de la primera parte de nuestra clase sobre C++

Los y las invitamos a ver la video-clase que complementa este material de lectura inicial.

Si surgen dudas o consultas, pueden escribirnos en el foro correspondiente.

En la próxima parte, comenzaremos a trabajar sobre los fundamentos de la programación orientada a objetos

¡Nos leemos!