



UANL
UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
FACULTAD DE CIENCIAS FÍSICO MATEMÁTICA



FCFM
FACULTAD DE CIENCIAS FÍSICO MATEMÁTICAS

UNIDAD TEMÁTICA I

FUNDAMENTOS DE C Y C++

Propósitos:

- ✓ Familiarizarse con la historia del lenguaje de Programación C y su evolución.
- ✓ Identificar las características del Lenguaje C.
- ✓ Familiarizarse con la historia del Lenguaje de programación C++.
- ✓ Identificar las características del Lenguaje C++.
- ✓ Identificar las diferencias entre lenguaje C y C++.
- ✓ Identificar las palabras claves o reservadas de C y C++.

Historia del lenguaje C

El lenguaje C evolucionó a partir de dos lenguajes previos, BCPL y B.

BCPL fue desarrollado en 1967 por Martín Richards, como un lenguaje para escribir software y compiladores de Sistemas Operativos.

Ken Thompson modeló muchas de las características de su lenguaje B basándose en el BCPL, y utilizó B en 1970 para crear versiones iniciales del sistema operativo UNIX en los laboratorios Bell, sobre una computadora PDP-7 de DEC. Tanto BCPL como B eran lenguajes < sin tipo >, donde cada elemento de datos ocupaba una palabra en memoria y quedaba a cargo del programador el tratar un elemento de datos como si se tratara de un número entero o de un número real.



UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

UANL

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
FACULTAD DE CIENCIAS FÍSICO MATEMÁTICA



FCFM

FACULTAD DE CIENCIAS FÍSICO MATEMÁTICAS

El lenguaje C fue derivado del lenguaje B y creado por Dennis Ritchie, de los Laboratorios Bell en 1972, y se implantó en una computadora PDP-11 de DEC. C al inicio se hizo muy conocido como lenguaje de desarrollo del sistema operativo UNIX. Hoy en día, virtualmente todos los sistemas principales están escritos en C y/o C++, C utiliza muchos de los conceptos importantes de BCPL y B, además de añadir los tipos de datos y otras características poderosas.

ANSI C

Hacia finales de los 70's, C había evolucionado a lo que hoy se conoce como C tradicional.

La publicación en 1978 del libro de Kernighan y de Ritchie, Te C Programming Language, atrajo gran atención sobre este lenguaje. Esta publicación se convirtió en uno de los libros científicos de computadoras de más éxito de todos los tiempos.

La expansión rápida de C sobre varios tipos de computadoras (plataformas) trajo consigo muchas variantes. Estas eran similares, pero a menudo no eran compatibles. Esto resultaba ser un problema serio para los desarrolladores de programas, que necesitaban escribir códigos que pudieran funcionar en varias plataformas. Se hizo cada vez más evidente que era necesaria una versión estándar de C. en 1983, se creó el comité técnico X3J11, bajo el American National Standards Committee on Computers and Information Processing (X3), para proporcionar una definición no ambigua e independiente de máquina del lenguaje.



UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

UANL

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
FACULTAD DE CIENCIAS FÍSICO MATEMÁTICA



FACULTAD DE CIENCIAS FÍSICO MATEMÁTICAS

En 1989 el estándar o norma quedó aprobado. El documento se conoce como ANSI / ISO 9899:1990. La segunda edición de Kernighan y Ritchie, publicada en 1988, refleja la versión que se conoce como ANCI C, la cual ahora se utiliza en todo el mundo (Ke88). ANCI C es la versión del lenguaje de programación C estandarizada en 1989, tanto en Estados Unidos, a través del American National Standards Institute (ANSI), como en todo el mundo a través de la International Standards Organization (ISO).

Características del Lenguaje C

Las **características** principales del Lenguaje C son:

- ✓ C es considerado como de Nivel Medio, debido a que combina los elementos de los lenguajes de Alto Nivel con la funcionalidad del Lenguaje Ensamblador.
- ✓ Una característica muy especial de C es que permite la manipulación directa de bits, bytes, palabras y direcciones (elementos básicos con los que trabaja la computadora).
- ✓ Genera un código objeto (en lenguaje maquinal) altamente optimizado y portable, porque es posible adaptar el software escrito para cualquier computadora (plataforma). Es un lenguaje que está disponible para cualquier computadora (por su diseño cuidadoso, es posible escribir programas en C que sean portátiles hacia la mayor parte de las computadoras).
- ✓ C permite casi todas las conversiones de tipos.

- ✓ C no lleva a cabo la comprobación de errores en el tiempo de ejecución tales como: sobrepasar límites o incompatibilidad de argumentos.
- ✓ Otro aspecto importante de C es que solo tiene 32 palabras claves (27 del estándar de Ritchie y Kernighan y 5 añadidas por el comité de Estandarización del ANSI).
- ✓ C se denomina lenguaje sencillamente estructurado ,debido a que permite compartir códigos y datos.
- ✓ C esta basado en funciones (componente estructural principal).
- ✓ C es un lenguaje para programadores, debido a que fue creado, influenciado y probado en vivo por programadores profesionales. El resultado final es que C da al programador lo que quiere: pocas restricciones, estructura de bloques, funciones independientes y un compacto uso de palabras claves.

- ✓ Cada programador de C puede crear y mantener su propia biblioteca de funciones que se ajusten a su personalidad y que pueden ser usadas en muchos programas diferentes.
- ✓ C permite que los programadores trabajen fácilmente en grandes proyectos y minimiza la duplicidad de esfuerzos.
- ✓ Es un lenguaje versátil, es fácil de aprender y no está orientado a un área en particular.



UANL

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
FACULTAD DE CIENCIAS FÍSICO MATEMÁTICA



FACULTAD DE CIENCIAS FÍSICO MATEMÁTICAS

Biblioteca estándar de C

- ✓ Todos los sistemas C consisten de tres partes: entorno, lenguaje y las bibliotecas estándar. Las funciones de biblioteca no forman parte de lenguaje C mismo; estas funciones ejecutan operaciones como entrada / salida y cálculos matemáticos.
- ✓ Los programas de C por lo regular pasan a través de seis fases para ser ejecutados: edición, preproceso, compilación, enlace, carga y ejecución.
- ✓ El programador escribe un programa utilizando un editor y si es necesario realiza correcciones.
- ✓ Un compilador traduce un programa C en código de lenguaje máquina(o código objeto).

- ✓ El preprocesador de C obedece directrices del preprocesador, que por lo regular indican otros archivos que se deben incluir en el archivo a compilar y que ciertos símbolos deben ser reemplazados con texto de programa (agrega al programa otros archivos y permite la definición de macros).
- ✓ Un enlazador vincula el código objeto con el código de funciones faltantes, para producir una imagen ejecutable (sin ninguna pieza faltante).
- ✓ Un cargador toma del disco una imagen ejecutable y la transfiere a memoria.

El C concurrente(es otra versión del C) es un superconjunto de C, que incluye capacidades para especificar la ejecución de múltiples actividades en paralelo.

Historia y Características del C++

- ✓ C++ es una versión expandida de C.
- ✓ C es flexible y potente y ha sido utilizado para crear algunos de los productos más importantes de software de los últimos 15 años. Sin embargo, cuando un proyecto excede un cierto tamaño, C alcanza sus límites. Viendo este problema, en 1980, Bjarne Stroustrup se dio a la tarea de agregar extensiones al lenguaje C, inicialmente se llamo << C con clases >>; y posteriormente en 1983 le dio el nombre de C++. Se ha convertido ya en una de las opciones más interesantes a la hora de elegir un lenguaje de programación, en tanto como aprendizaje, como para el campo profesional.



UANL

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
FACULTAD DE CIENCIAS FÍSICO MATEMÁTICA



FACULTAD DE CIENCIAS FÍSICO MATEMÁTICAS

- ✓ La mayoría de los agregados a C por Stroustrup soportan la programación orientada a objetos. C++ ha sido revisado dos veces desde su invención, una en 1985 y de nuevo en 1990.
- ✓ Cuando Stroustrup inventó el C++, sabía que era importante mantener el espíritu original de C, incluido se eficiencia, flexibilidad y la filosofía de que el programador es el que manda, añadiendo al mismo tiempo soporte para la programación orientada a objetos.
- ✓ C++ proporciona al programador la libertad y el control de C junto con el poder de los objetos (permiten crear programas estructurados con claridad, extensibilidad y facilidad de mantenimiento sin pérdida de eficiencia).
- ✓ En realidad, los atributos de orientación a objetos de C++ pueden aplicarse con eficiencia virtualmente a cualquier tarea de programación.

- ✓ C++ se usa en proyectos tales como: Editor, Bases de Datos, Sistemas de Archivos Personales y Programas de Comunicaciones.
- ✓ Además pueden construirse sistemas de Software de alto rendimiento ya que C++ comparte la efectividad de C.
- ✓ Existen numerosas versiones de compiladores de C++, excelentes la mayoría de ellas, que siguen la norma estándar ANSI C++ 2.0 y 2.1 (Borland C++ 6.0, Microsoft C/C++ 7.0, etc.).

Los objetos son en esencia componentes de software reutilizables que modelan elementos del mundo real. Está en marcha una revolución en la comunidad de software y sigue siendo una meta no alcanzable: la elaboración del software rápido, correcto y económico; y esto en un momento en que las demandas del software nuevo y más poderoso están encumbrándose.

Todo objeto tiene:

- ✓ Estado (propiedad o característica del mismo).
- ✓ Comportamiento (forma de actuar en términos de algún cambio de estado).
- ✓ Identidad (propiedad que lo distingue).

Los desarrolladores de software están descubriendo que utilizando un diseño e implantación modular orientada a objetos, puede hacer que los grupos de desarrollo de software sean de 10 en 100 veces más productivos que lo que era posible mediante técnicas convencionales de programación.

Lenguajes orientados a objetos

La programación orientada a objetos es una nueva forma de enfocar el trabajo de la programación.

La POO toma las mejores ideas de la programación estructurada y las combina con nuevos y poderosos conceptos que animan una nueva visión de la tarea de programación.

La POO permite descomponer fácilmente un problema en subgrupos de partes relacionadas. Así, puede traducir estos subgrupos en unidades autocontenidas llamadas objetos.

Naturalmente, la programación orientada a objetos requiere el uso de lenguajes de programación idóneos.

Existen dos tipos de lenguajes de POO: puros e híbridos.



UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

UANL

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
FACULTAD DE CIENCIAS FÍSICO MATEMÁTICA



FACULTAD DE CIENCIAS FÍSICO MATEMÁTICAS

Los lenguajes puros son aquellos diseñados en base a las propiedades específicas de la programación orientada a objetos, y tienen unas construcciones léxicas y lógicas muy diferentes de las encontradas en los lenguajes estructurados tradicionales. Su primer representante fue Simula (lenguaje de simulación basado en el ALGOL), hoy en día su representante más genuino es Smaltalk, y la versión Smaltalk/V la más difundida.

Los lenguajes híbridos son lenguajes que añaden a las propiedades tradicionales estructuradas, características orientadas a objetos. Hoy en día son los más difundidos y entre ellos, podemos citar:

- C++
- Turbo Pascal (5.5, 6.0, 7.0)
- Objective Pascal
- Objective C

Otros ejemplos de lenguajes de programación orientados

- | | | |
|--|---|------------------------------------|
| • <u>ADA</u> | <u>C++</u> | <u>Objective C</u> |
| • <u>Java</u> | <u>Smalltalk</u> | <u>Eiffel</u> |
| • <u>Ruby</u> | <u>Python</u> | <u>SDK</u> |
| • <u>OCAML</u> | <u>Object Pascal</u> | <u>CLIPS</u> |
| • <u>Actionscript</u> | <u>Pauscal [En español]</u> | <u>Perl</u> |
| • <u>C#</u> | <u>Visual Basic.NET</u> | <u>Simula</u> |
| • <u>Delphi</u> | <u>PowerBuilder</u> | <u>Maya</u> |
| • <u>Visual Fox Pro V9</u> | | |

Aplicaciones de los lenguajes orientado a objetos:

- jstruct
- engine
- ui
- Windows
- Corel Draw
- Fox
- Office

Diferencias entre C y C++

Para la mayoría, C++ es un superconjunto del C estándar ANSI, virtualmente todos los programas de C son también programas de C++. Sin embargo, existen unas pocas diferencias, de las cuales las más importantes son:

- ✓ Una de las diferencias más importantes, a la vez que sutil, entre C y C++, es que en C una función declarada como: `int f()`, no indica nada sobre ningún parámetro de esa función (cuando no se especifica nada entre los paréntesis que siguen al nombre de la función, puede haber parámetros, pero no se dice nada sobre ninguno de ellos en la declaración), sin embargo, en C++, la declaración de la función anterior indica que la función no tiene parámetros. En C++, cuando se escribe: `int f()` o `int f(void)` indica que son equivalentes.
- ✓ En C++, el tipo `void` es opcional.

En C++, todas las funciones deben tener prototipo. Esto es opcional en C.

Una diferencia pequeña pero importante entre C y C++ es que en C, un carácter constante se pasa automáticamente a un entero. En C++, no sucede así.

En C, no es un error declarar varias veces una variable global, aunque es una mala práctica de programación.

En C++, esto es un error.

En C, un identificador puede tener hasta 31 caracteres de largo. En C++, no existe límite.



UANL

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
FACULTAD DE CIENCIAS FÍSICO MATEMÁTICA



FACULTAD DE CIENCIAS FÍSICO MATEMÁTICAS

Palabras claves o reservadas del Lenguaje C/C++

Las palabras claves representan a conjuntos de caracteres (identificadores) especiales para el lenguaje C y C++ (compilador), por lo que no pueden ser utilizadas por el usuario para otros efectos.

Palabras claves o reservadas de C

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while