
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL AVELLANEDA

CLASE 1

Historia:

Antecedentes del lenguaje C

Año	Lenguaje
1960	CPL (Combined Programming Lenguage)-Universidad de Cambridge y Londres.
1966	BCPL(Basic Combined Programming Lenguage)-Martin Richards Universidad de Cambridge
1970	B –Laboratorios Bell – Kenneth L. Thompson y Dennis Ritchie
1978	C – Dennis Ritchie y Brian Kernighan publican la Biblia de C

CLASE 1

Historia:

-Es un lenguaje para “Programadores”, por su flexibilidad para programar y baja comprobación de incorrecciones, es decir no corrige, es el programador quien debe hacerlo. Otros lenguajes poseen auto-correcciones.

-Es un lenguaje estructurado como Pascal, ADA, etc.

-Permite la conversión y asignación entre diferentes tipos de datos.

-Tiene una función principal que se ejecuta primero y se llama MAIN quien llama a otro conjunto de funciones.

-Es case-sensitive.

-Existen diferentes versiones de C, para unificarlo se creó un standard llamado ANSI-C.

CLASE 1

ANSI-C:

Solo utiliza un total de 32 palabras reservadas.

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

CLASE 1

Identificador:

Son los nombres de variables, funciones o etiquetas definidas por el programador. Puede ser de 1 a 32 caracteres, y el primer carácter puede ser una letra o un guión bajo.
Nota: No olvidar que es Case-sensitive.

Estándar: Facilita la lectura de un programa

-Variables: En minúsculas o solo 1er. Letra mayúscula.
-Constantes: Mayúsculas

CLASE 1

Tipo de datos:

Tipo de dato	Bits	Rango (con signo)	Rango (sin signo)
char	8	-128 a +127	0 a 255
int(1)	16	-32768 a +32767	0 a 65535
long(2)	32	–2147483647 a +2147483647	0 a 4294967295
float	32	$-3.2x10^{+38}$ a $+3.2x10^{+38}$	
double	64	$-1.7x10^{+308}$ a $+1.7x10^{+308}$	
void	0	sin valor	sin valor

El tipo *int y long* debe tener 16 y 32 bits **como mínimo**, comúnmente en sistemas operativos para PC regulares, tiene 32 bits

CLASE 1

Tipo de datos:

Tipo de dato	Bits	Rango (con signo)	Rango (sin signo)
char	8	-128 a +127	0 a 255
int(1)	16	-32768 a +32767	0 a 65535
long(2)	32	–2147483647 a +2147483647	0 a 4294967295
float	32	$-3.2x10^{+38}$ a $+3.2x10^{+38}$	
double	64	$-1.7x10^{+308}$ a $+1.7x10^{+308}$	
void	0	sin valor	sin valor

El tipo *char* se utiliza normalmente para almacenar valores definidos en la tabla ASCII

CLASE 1

Tipo de datos:

Tipo de dato	Bits	Rango (con signo)	Rango (sin signo)
char	8	-128 a +127	0 a 255
int(1)	16	-32768 a +32767	0 a 65535
long(2)	32	–2147483647 a +2147483647	0 a 4294967295
float	32	$-3.2x10^{+38}$ a $+3.2x10^{+38}$	
double	64	$-1.7x10^{+308}$ a $+1.7x10^{+308}$	
void	0	sin valor	sin valor

El tipo *int* se utilizan para guardar números enteros y los *float* y *double* se utilizan para números reales.

CLASE 1

Modificadores del tipo de dato:

signed	Aplicable a *char*, *int* y *long*
unsigned	
short	Aplicable a *int*
long	Aplicable a *int* y a *long*

Por defecto, las variables son signed.

Unsigned se utiliza cuando la variable no llevará signo.

Long extiende el rango a 32 o 64 bits.

CLASE 1

Constantes:

Son valores fijos que no se cambian en el transcurso de la ejecución del programa.

Se definen con la directiva #define.

Los valores se escriben de la siguiente manera:

Tipo Char: Entre apostrofes ej.: ‘a’ o ‘\n’

Tipo Int: Directamente ej.: 123 o -2000

Tipo Float / Double: Con punto para los decimales ej.: 123.3 o -0.05

Tipo Cadena de Carácter: Entre comillas ej.: “Hola Mundo”

CLASE 1

Operador Aritmético:

Operador	Significado
+	Suma
-	Resta
*	Multiplicación
/	División
%	Resto de la división
++	Incremento
--	Decremento

CLASE 1

Operador Aritmético, Relacional y Lógico:

Le indica al compilador que operación realizar.

Operador	Significado
+	Suma
-	Resta
*	Multiplicación
/	División
%	Resto de la división
++	Incremento
--	Decremento

Aritmético

Operador	Significado
>	Mayor
>=	Mayor o igual
<	Menor
<=	Menor o igual
==	Igual
i=	Distinto

Relacional

Operador	Significado
&&	AND
| |	OR
i	NOT

Lógico

CLASE 1

Forma General de un Programa en C:

Las directivas *include* pueden no estar al principio del programa y los *define* no deben estar inmediatamente después de los *include.*

- Archivos de cabecera	#include <stdio.h>
- Declaración de constantes y macros	#define MAXNOM 20 #define MAXDIR 30 #define MAXPERS 100

- Declaración de estructuras	struct gente{
	char nombre[MAXNOM];
	char dir[MAXDIR];
	int edad;
	};

- Declaración de prototipos de funciónes	void CargaGente(struct gente *); int ValidaEdad(void);
- Declaración de variables globales	int flag;

CLASE 1

Forma General de un Programa en C:

Es conveniente para que se mantenga un orden en la escritura de todos los programas. Partes de un Programa en C.

- Archivos de cabecera	#include <stdio.h>
- Declaración de constantes y macros	#define MAXNOM 20 #define MAXDIR 30 #define MAXPERS 100

- Declaración de estructuras	struct gente{
	char nombre[MAXNOM];
	char dir[MAXDIR];
	int edad;
	};

- Declaración de prototipos de funciónes	void CargaGente(struct gente *); int ValidaEdad(void);
- Declaración de variables globales	int flag;

CLASE 1

Forma General de un Programa en C:

- Desarrollo de la función principal	

void main(void)
{
variables locales de main
- - - - - - -
- - - - - -
}
	void main(void)
	{
	struct gente agenda[MAXPERS];
	int i, cant;
	- - - - - - - - - -
	}

- Desarrollo de otras funciones	

int f1 (char x)
{
Variables locales de f1
}
void f2 (void)
{
Variables locales de f2
}
	void cargaGente(char * p)
	{
	int i , cant;
	}
	int validaEdad(void)
	{
	int edad;
	}


UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL AVELLANEDA