

Computación I (CI-2125)

Clase 2

Prof. Mireya Morales



Repaso de algunos tópicos

- **Abstracción:** Consiste en extraer las propiedades más importantes de un objeto, dejando los detalles para el diseño específico.
- **La encapsulación:** permite ofrecer a los usuarios una visión de caja negra, de manera que solo se exporte la interfaz de usuario.
- **Ocultamiento de Información:** Consiste en no mostrar al exterior datos o funciones que no sean necesarias.
- **Modularidad:** Proceso de dividir un objeto en piezas más pequeñas

Repaso de algunos tópicos

Pasos a seguir para el diseño de un Algoritmo

- Definición o especificación del problema.
- Descomposición del problema en subproblemas más simples (análisis descendente).
- Combinar las estructuras algorítmicas básicas para resolver cada subproblema.
- Ensamblar las soluciones de cada subproblema

Repaso de algunos tópicos

Definición o especificación del problema

- Especificación de Entrada. Descripción de los datos de entrada del programa
 - Valores específicos de entrada del programa
 - Formato
 - Rango de validez para cada dato de entrada. (Precondición)
- Especificación de salida. Descripción de los datos de salida del programa
 - Que valores deben ser producidos y que propiedades deben cumplir, expresadas en término de relaciones entre los datos de entrada y de salida (Postcondición)
- Procesamiento especial (“Casos de borde”). Verificar y procesar ciertas condiciones que podrían llevar a error

Repaso de algunos tópicos

Definición o especificación del problema

- Ejemplo. Dada una cantidad en segundos, calcular su equivalente en horas minutos y segundos

```
Var totalseg:entero      /*Entrada*/
```

```
Var horas, min, seg: entero /*Salida*/
```

```
{Precondición: totalseg > 0}
```

```
{Postcondición: totalseg=seg + 60*min +3600*horas y 0<=seg,  
min < 60 y horas >=0}
```

Historia del Lenguaje C

MULTICS



UNIX

BCPL

Martin Richards, 1967



B

Ken Thompson, 1970



C

Dennis Ritchie, 1972



Historia del Lenguaje C

- El lenguaje C fue implementado e inventado por Dennis Ritchie bajo UNIX; basado en dos lenguajes anteriores: BCPL de Martín Richards y B de Ken Thompson.
- Con la popularidad de las microcomputadoras muchas compañías comenzaron a implementar su propio C por lo cual surgieron discrepancias entre sí. Por esta razón ANSI (American National Standards Institute, por sus siglas en inglés), estableció un comité en 1983 para crear una definición no ambigua del lenguaje C e independiente de la máquina que pudiera utilizarse en todos los tipos de C.

Contenido

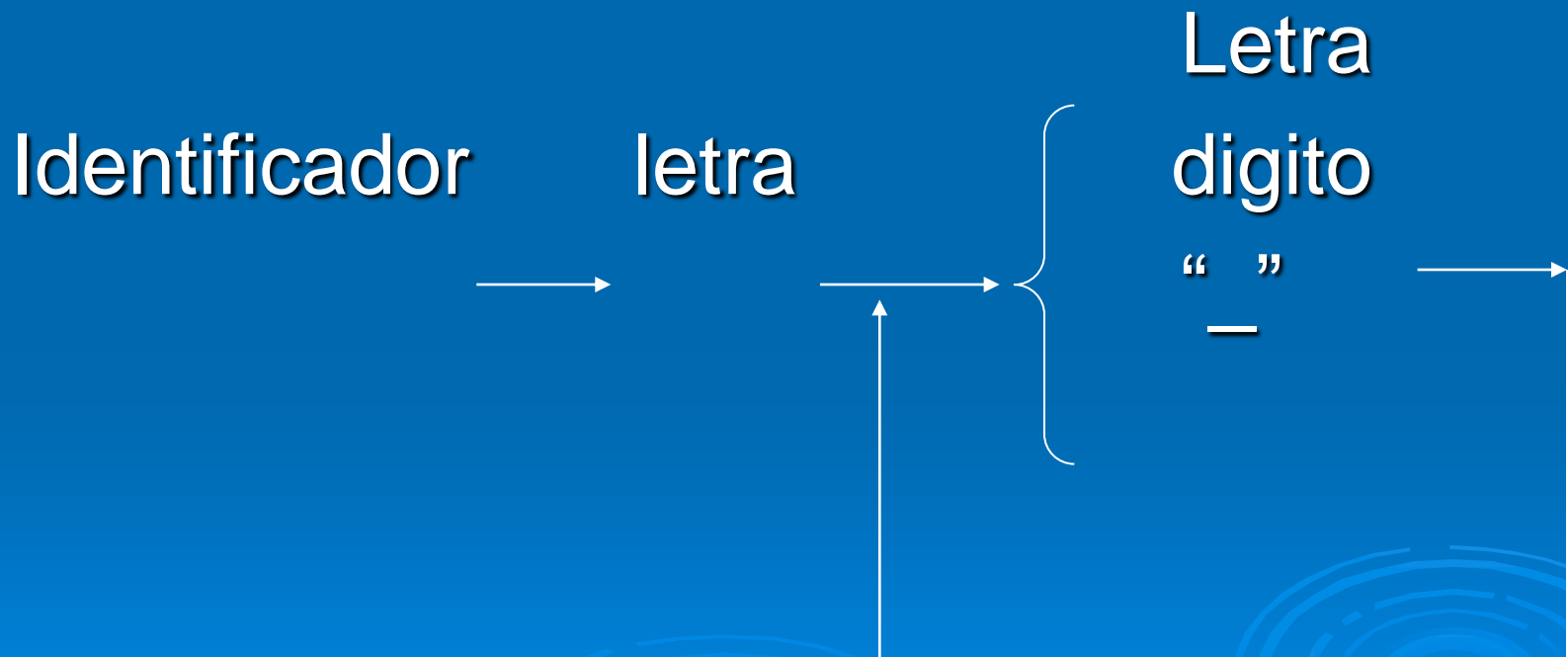
- Estructura general de un programa en C
- Constantes
- Variables
- Tipos de datos básicos
- Expresiones aritméticas y lógicas
- Orden de precedencia
- asignación
- Entrada y salida (scanf, printf)

Identificadores

- Es un nombre que se asigna a distintos elementos de un programa, como pueden ser variables, nombre de funciones, etc.

Identificadores

Los caracteres válidos para formar un identificador son: a-z, A-Z, 0-9, y el _ (underscore).



Identificadores

- Pueden tener cualquier longitud, pero solo los primeros 8 caracteres son significativos.
- Las mayúscula y minúsculas son significativas
- Las palabras reservadas no pueden utilizarse
- Deben utilizarse identificadores nemónicos que tipifiquen el contenido de la variable.
- Se recomienda utilizar identificadores con letras minúsculas para variables y con mayúsculas para constantes simbólicas.

Constantes

- Es un valor que se fija durante todo el período de vida que dura la ejecución de un programa. Para definir una constante en C se emplea la directiva *#define*.

Ejemplo:

```
#define PI 3.1415
```

Variables

- Son objetos que pueden cambiar su valor durante la ejecución de un programa. Cada variable se asocia con una determinada zona dentro de la memoria del computador. El tamaño de esta zona, en bytes, dependerá del tipo de datos del valor que se almacene en la variable.

Ejemplo:

$A1 = 10$

$A2 = A1 + 15$

$A1 = 20$

Tipos de datos básicos

- C dispone de tres tipos de datos básicos: caracteres, números enteros y números reales. La principal característica es que ocupan sólo una casilla de memoria.

Tipos de datos básicos

Tipo de datos	Descripción	Rango
int	Enteros	-32.768 a +32.768
float	Reales	3.4×10^{-38} a 3.4×10^{38}
long	Entero de largo alcance	-2,147,483,648 a 2,147,483,647
double	Reales de doble precisión	1.7×10^{-308} a 1.7×10^{308}
char	caracter	Símbolos encerrados en ' '

Tipos de datos booleanos en C

- C no dispone de un tipo de dato booleano para representar los valores verdadero (true) o falso (false).
- Normalmente se emulan en C utilizando el valor 0 como falso y el valor 1 como verdadero.
- Se suele recurrir a la definición de las constantes TRUE y FALSE mediante la directiva define:
 - `#define TRUE 1`
 - `#define FALSE 0`

Operadores Aritméticos

= Asignación

* Multiplicación

/ División

% Módulo

+ Suma

- Resta



Operadores Relacionales

$=$ Igual

\neq Diferente

$>$ Mayor

\geq Mayor o igual

$<$ Menor

\leq Menor o igual

Operadores Lógicos

- `&&` AND `(x>6)&&(z==3)`
- `||` OR `(y==6) || (y!=7)`
- `!` NOT `!(x>=8)`

Precedencia y Orden de Evaluación

Operador	Asociatividad
() [] -> ●	Izq a Der
! ++ -- (tipo) * & sizeof	Der a Izq
* / %	Izq a Der
+ -	Izq a Der
< > (Shift desplaza bits)	Izq a Der
<<= >>=	Izq a Der
== !=	Izq a Der
& (a nivel de bits)	Izq a Der
&&	Izq a Der
	Izq a Der
= += -= *= %= /=	Izq a Der

Definiciones en el cálculo lógico

Conjunción

- La conjunción es un operador que opera sobre dos valores de verdad, típicamente los valores de verdad de dos proposiciones, devolviendo el valor de verdad *verdadero* cuando ambas proposiciones son verdaderas, y *falso* en cualquier otro caso.

Tabla de verdad para la conjunción (&) es la siguiente:

A	B	$A \wedge B$
1	1	1
1	0	0
0	1	0
0	0	0

Disyunción

- La disyunción es un operador que opera sobre dos valores de verdad, típicamente los valores de verdad de dos proposiciones, devolviendo el valor de verdad *verdadero* cuando una de las proposiciones es verdadera, o cuando ambas lo son, y *falso* cuando ambas son falsas.

La tabla de verdad de la disyunción (\vee) es la siguiente:

A	B	$A \vee B$
1	1	1
1	0	1
0	1	1
0	0	0

Negación

- La negación es un operador que opera sobre un único valor de verdad, típicamente el valor de verdad de una proposición, devolviendo el valor de verdad *verdadero* si la proposición es falsa, y *falso* si la proposición es verdadera.

La tabla de verdad de la negación es la siguiente:

A	$\neg A$
1	0
0	1

Entrada y salida (scanf, printf)

➤ Lenguaje C

