

# Clase #05 de 27

Tipos,  
Símbolos & Alfabeto,  
Expresiones & Iteraciones,  
y Constantes Simbólicas

*Abril 15, Lunes*

# Agenda para esta clase

- Tipos de Datos
- Símbolos o Caracteres & Alfabetos
- Revisión de Trabajo #1
- Intervalo
- Expresiones & Iteraciones
- Constantes simbólicas
- Funciones del Preprocesador – Parte II

# Tipos de Datos

# Algunos tipos de datos

- Tipos Enteros, ordenados por rango
  - **bool** Lógico (C99, **stdbool.h**)
  - **char** Código
  - **short** Corto
  - **int** Palabra
  - **long** Largo
- Tipos Flotantes Reales, ordenados por rango
  - **float** Simple precisión
  - **double** Doble precisión
- El tamaño depende de la máquina, aunque hay restricciones y mínimos
- Variantes de **int**
  - **unsigned**
  - **short**
  - **long**
  - **long long** (C99)
- Variantes de **double**
  - **long**
- Tipos Flotantes complejos
  - **float \_Complex**
  - **double**
  - **\_Complex**
  - **long double \_Complex**
- Tipos Aritméticos
  - Tipos Enteros
  - Tipos Flotantes, Reales y Complejos
- Tipos Básicos
  - **char**
  - Tipos enteros
  - Tipos flotantes
- Enumeraciones
  - **enum**
- Tipos Derivados (de los anteriores)
  - Arreglos
  - Estructuras
  - Uniones
  - Punteros a todos ellos.

# Objetos: Tipos, Declaraciones y Variables



- **Declaración**
  - Introduce un **nombre** en el programa
  - Especifica el **tipo** para una entidad nombrada
- **Tipo**
  - Conjunto de **valores** y conjunto de **operaciones** sobre esos valores
- **Objeto**
  - **Bloque de memoria principal** que contiene un **valor** de un **tipo**
- **Valor**
  - Secuencia de **bits** que se **interpretan** según un **tipo**
- **Variable**
  - **Objeto nombrado**
- Algunos tipos fundamentales de C
  - Entero ( $\mathbb{Z}$ ), por ejemplo, 1, 42, y 1066
    - `int`
  - Carácter ( $\Sigma$ ), por ejemplo, 'a', 'z', y '9'
    - `char`
  - Booleano ( $\mathbb{B}$ ): `true` y `false`
    - `bool`
  - Número de punto flotante de doble precisión ( $\mathbb{R}$ ), por ejemplo, 3.14 y 2999793.0
    - `double`
- Declaraciones Ejemplo:
  - `int i=42;`
  - `char c='9';`
  - `bool b=true`
  - `double d=3.14;`

# Símbolos ó Caractereres & Alfabetos

[MUCH2012] 1.1.1

# Símbolo o Carácter

- Es Indivisible
- Es cualquier elemento que se defina como símbolo
- Pertenece a un Alfabeto ( $\Sigma$ )
- Ejemplo 3 de [MUCH2012]  
víci: El alfabeto  
 $\Sigma = \{0, 1\}$  proporciona los caracteres para construir los números binarios
- Ejemplo: El  $\Sigma = \{\text{if}, \text{for}\}$  tiene dos caracteres
- Ejercicio 1 de [MUCH2012]  
víci: Escriba el  $\Sigma$  para construir el conjunto de los números enteros con signo en base 10
- Operaciones
  - Comparación

# Alfabeto $\Sigma$

- Es un Conjunto, cuyos elementos son símbolos o caracteres
- Restricciones básicas
  - Finito
  - No vacío
- Ejemplos
  - $\{\} = \emptyset \Rightarrow$  no es alfabeto, por ser vacío
  - Naturales  $\Rightarrow$  no es alfabeto, por ser infinito
  - ASCII
  - Dígitos =  $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
  - Meses =  $\{\text{Lunes, Martes, Miércoles, Jueves, Viernes, Sábado}\}$
- Restricción avanzada
  - Sus elementos no pueden formarse por yuxtaposición de otros elementos
- Ejemplos
  - $\{0, 1, 1001\}$
  - $\{\text{if, for}\}$
  - $\{a, b, c, \text{ch}, d, e, f, g, h, i, j, k, l, \text{ll}, m, n, \text{ñ}, o, p, q, r, \text{rr}, s, t, u, v, w, x, y, z\}$ .



# Caracteres y Alfabetos – Implementación en ANSI C

- Alfabeto tipo de dato char
- Carácter valor y variable
  - A
    - Literal 'A'
    - Literal 65
  - $c \in \Sigma$ 
    - char c;

# Revisión de Trabajo #1

# Expresiones & Iteraciones

K&R 1.2-1.3 Variables, Tipo de datos, Expresiones Aritméticas y la Sentencia For

# Problema – Tabla Fahrenheit-Celsius

$$^{\circ}\text{C} = 5/9 ( ^{\circ}\text{F} - 32 )$$

0	-17
20	-6
40	4
60	15
80	26
100	37
120	48
140	60
160	71
180	82
200	93
220	104
240	115
260	126
280	137
300	148

# Resolución – Fahrenheit-Celsius

$$^{\circ}\text{C} = 5/9 ( ^{\circ}\text{F} - 32 )$$

```
// F2C K&R 1988
#include <stdio.h>
int main(void){
    int fahr, celsius;
    int lower, upper, step;

    lower = 0;    //scale lower limit
    upper = 300;  //upper limit
    step = 20;    //step size
    fahr = lower;

    while (fahr <= upper) {
        celsius = 5 * (fahr-32) / 9;
        printf("%d\t%d\n", fahr, celsius);
        fahr = fahr + step;
    }
}
```

- Comentarios
- Variables
  - Abstracción de datos
  - Tipo de datos
- Declaraciones de variables
  - Anuncio de propiedades
  - Asociación
  - Declaración con varios Declaradores
- Comienzo de la Ejecución (Cómputo)
- Sentencias
- "Sentencia de asignación"
- Sentencia while
- Sentencia compuesta
- Indentación
  - Estilos de codificación
- Secuencia
- Expresiones aritméticas
- División entera y real
- Formateo de la salida
  - ¿Cómo mejorar la alineación?.

# F-C 2 – Formato y Precisión

```
#include <stdio.h>
int main(void){
    float fahr, celsius;
    int lower, upper, step;

    lower = 0;    // lower limit of temperature scale
    upper = 300;  // upper limit
    step = 20;    // step size
    fahr = lower;

    while (fahr <= upper) {
        celsius = (5.0/9.0) * (fahr-32.0);
        printf("%3.0f %6.1f\n", fahr, celsius);
        fahr = fahr + step;
    }
}
```

# Otros formatos para printf

- Diferencia entre Valor y Formato
  - 65 , 41, 101, LXV, A
  - 1000001
- %d entero decimal
- %i también entero decimal, la diferencia está en la función scanf
- %6d entero decimal, por lo menos ancho 6
- %f punto flotante
- %6f punto flotante, por lo menos ancho 6
- %.2f punto flotante y 2 caracteres luego del punto
- %6.2f punto flotante, por lo menos ancho 6 y 2 caracteres luego del punto
- %o octal
- %x hexadecimal
- %c carácter
- %s string
- %% por ciento

# F-C 3 – For generaliza While y una "Best Practice"

```
#include <stdio.h>
int main(void){
    float fahr, celsius;
    int lower, upper, step;

    lower = 0;
    upper = 300;
    step = 20;
    fahr = lower;

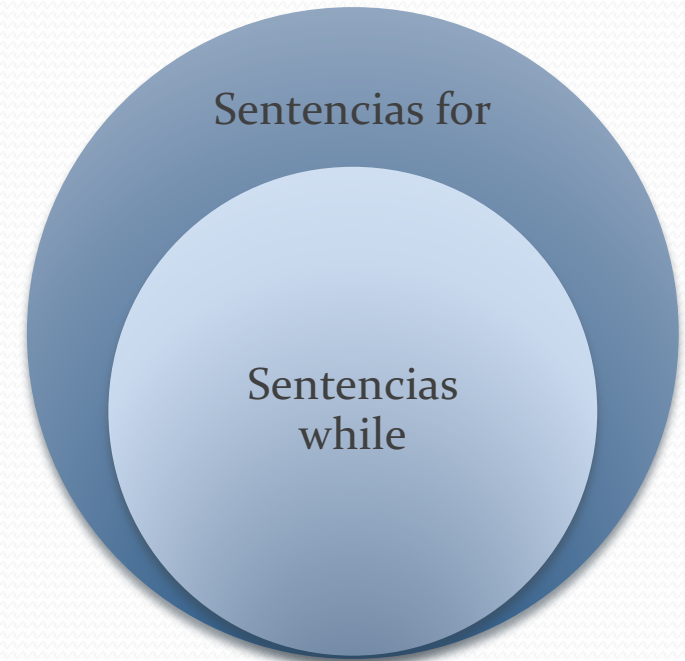
    while (fahr <= upper) {
        celsius = (5.0/9.0) * (fahr-32.0);
        printf("%3.0f %6.1f\n", fahr, celsius);
        fahr = fahr + step;
    }
}
```

- for por while
- Variable celsius reemplazada por expresión

```
#include <stdio.h>

int main(void){
    int fahr;

    for( fahr = 0; fahr <= 300; fahr = fahr + 20 )
        printf("%3d %6.1f\n", fahr, (5.0/9.0)*(fahr-32) );
}
```





# Ejercicios

- 1-3. Encabezado sobre la tabla
- 1-4. C-F.

# Constantes simbólicas

K&R 1.4 Constantes Simbólicas

# “Números mágicos”

```
#include <stdio.h>
```

```
int main(void){  
    int fahr;
```

```
    for( fahr = 0; fahr <= 300; fahr = fahr + 20 )  
        printf("%3d %6.1f\n", fahr, (5.0/9.0)*(fahr-32) );
```

```
    return 0;  
}
```

# Constantes (o Nombres) Simbólicas

```
#include <stdio.h>
#define LOWER 0    // lower limit of table
#define UPPER 300 /* upper limit
#define STEP 20   // step size

int main(void){
    int fahr;

    for( fahr = LOWER; fahr <= UPPER; fahr = fahr + STEP )
        printf("%3d %6.1f\n", fahr, (5.0/9.0)*(fahr-32) );
}
```

# C11: Calificador const y Declaración en for

```
int main(void){  
    const int LOWER = 0,    // lower limit of table  
             UPPER = 300,  // upper limit  
             STEP  = 20;   // step size  
  
    for (int fahr = LOWER; fahr <= UPPER; fahr = fahr + STEP)  
        printf("%3d %6.1f\n", fahr, (5.0/9.0)*(fahr-32));  
}
```

- Ejercicio 1-5. Modifique el programa para que imprima la tabla en orden inverso, es decir, desde 300 hasta 0 grados.

# Funciones del Preprocesador

## Parte II

# Funciones del Preprocesador, Parte II

- Ya conocidas
  - Incluir archivos encabezado, directiva `#include`
  - Reemplazar comentarios por un espacio
  - Concatenar cadenas adyacentes.
- Nueva
  - Atender directivas `#define` que definen nombres macro y su lista de reemplazo
  - Expandir los nombres macro
- Best Practice
  - Evitar, en lo posible, el uso `define`, buscar alternativas más abstractas y con misma eficiencia.
    - [http://www.stoustrup.com/bs\\_faq2.html#macro](http://www.stoustrup.com/bs_faq2.html#macro)

# Términos de la clase #05

Definir cada término con la bibliografía

- Tipos de Datos
  - Tipos Enteros
  - Tipos Flotantes
  - Variantes de tipo
  - Tipos Aritméticos
  - Tipos "Básicos"
  - Tipos Aritméticos
  - Tipos Derivados
  - Arreglos
  - Estructuras
  - Uniones
  - Punteros
  - Precisión: Moneda
- Símbolos ó Caracteres & Alfabetos
  - Símbolo o carácter en LF
  - Concatenar caracteres
  - Potencia de un carácter
  - Alfabeto  $\Sigma$  en LF
  - Símbolo o carácter en ANSI C
- Alfabeto en ANSI C
- Expresiones & Iteraciones
  - Abstracción de datos
  - Declaraciones
  - División entera y real
  - Operación cerrada
  - Valor de una expresión
  - Efecto de lado de una expresión
  - Formateo de valor
- Constantes simbólicas
  - "Números mágicos"
  - Constantes simbólicas ó Nombres simbólicos
  - Directiva #define
  - Calificador const
- Funciones del Preprocesador Parte II
  - Definición
  - Expansión
  - Best Practice: Evitar defines.



# Tareas para la próxima clase

1. Corregir Trabajo #1
2. Leer de [K&R1988] **1.5 Entrada y Salida de a Caracteres**
1. Releer de [MUCH2012] 1.1 (volumen 1, capítulo 1).

The background of the slide features a dark blue gradient with several light blue, wavy, horizontal lines that create a sense of movement and depth.

¿Consultas?

The background of the slide features a dark blue gradient with several lighter blue, wavy, horizontal lines that create a sense of movement and depth. These lines are more prominent in the upper half of the image.

**Fin de la clase**