



CIMAT

Informática Aplicada I

Clase 5

Marcela Morales Quispe

CIMAT

Septiembre 11, 2013

marcelamq@ciamat.mx

- 1 Tomando desiciones
- 2 Operadores
- 3 Estructura de repetición

Sentencia **if**

La sentencia **if** de los dos bloques que se muestra a continuación son equivalentes, éstos pueden usarse indistintamente. Si se tuviese un programa en el que es necesario validar muchas condiciones entonces es mejor hacer uso de la sentencia **switch**, el cual veremos más adelante.

<pre> if(condición) { sentencia-1; sentencia-n; } else { if(condición) sentencia-4; else sentencia-5; } </pre>	<pre> if(condición) { sentencia-1; sentencia-n; } else if(condición) sentencia-4; else sentencia-5; </pre>
--	---

Operadores de asignación

C dispone de varios operadores de asignación para la abreviatura de las expresiones de asignación. Por ejemplo, el enunciado `c = c + 3;` puede ser abreviado usando el operador de asignación `+=` como `c += 3;`. El operador `+=` añade el valor de la expresión, a la derecha del operador, al valor de la variable a la izquierda del operador, y almacena el resultado en la variable a la izquierda del operador.

Cualquier enunciado de la forma

$$\textit{variable} = \textit{variable operador expresión};$$

pueden ser escritos de la forma

$$\textit{variable operador} = \textit{expresión};$$

En la tabla 1 aparecen los operadores asignación aritméticos, con expresiones de muestra utilizando éstos. Asume `int c = 3, d = 5, e = 4, f = 6, g = 12;`

Operador	Ejemplo	Explicación	Asignación
<code>+=</code>	<code>c += 7</code>	<code>c = c + 7</code>	10 a c
<code>-=</code>	<code>d -= 4</code>	<code>d = d - 4</code>	1 a d
<code>*=</code>	<code>e *= 5</code>	<code>e = e * 5</code>	20 a e
<code>/=</code>	<code>f /= 3</code>	<code>f = f / 3</code>	2 a f
<code>%=</code>	<code>g %= 9</code>	<code>g = g % 9</code>	3 a g

Tabla 1: Operadores de asignación aritméticos.

Operadores incrementales y decrementales

C también tiene el *operador incremental* unario ++, y el *operador decremental* -- que se resumen en la tabla 2. Si una variable *c* es incrementada en 1, el operador incremental ++ puede ser utilizado en vez de las expresiones $c = c + 1$ o bien $c += 1$.

Operador	Ejemplo	Explicación
++	++a	Se incrementa <i>a</i> en 1 y a continuación se utiliza el nuevo valor de <i>a</i> en la expresión en la cual reside <i>a</i> .
++	a++	Utiliza el valor actual de <i>a</i> en la expresión en la cual reside <i>a</i> , y después se incrementa <i>a</i> en 1.
--	--b	Se decrementa <i>b</i> en 1 y a continuación se utiliza el nuevo valor de <i>b</i> en la expresión en la cual reside <i>b</i> .
--	b--	Utiliza el valor actual de <i>b</i> en la expresión en la cual reside <i>b</i> , y después se decrementa <i>b</i> en 1.

Tabla 2: Operadores incrementales y decrementales.

Sentencia **for**

La estructura de repetición **for** maneja de manera automática todos los detalles de la repetición controlada por contador. La forma general de esta estructura es:

```
for(expresión-1 ; expresión-2 ; expresión-3)  
{  
    sentencia-1;  
    sentencia-2;  
    sentencia-n;  
}
```

donde *expresión-1* inicializa la variable de control del ciclo, *expresión-2* es la condición de continuación del ciclo, y *expresión-3* incrementa la variable de control.