Sentencias de Control de Flujo

Sentencias de Control de Flujo de Ejecución

Sentencias Condicionales

Sentencia if

Su forma general es:

```
If (condición)
   {
     sentencia ;
   }
```

Se evalúa la condición. Si el resultado es true (!=0), se ejecuta sentencia; si el resultado es false (=0), se salta sentencia y se prosigue con la línea siguiente del programa.(*) La sentencia puede ser una sentencia simple o compuesta(con más de una sentencia).

Si la condición es verdadera se ejecuta la sentencia1, de lo contrario la sentencia 2.

Sentencia if ----else

```
If (condición)
   {
     sentencia 1;
   }
else
    {
      sentencia 2;
    }
```

Ejemplo

condición

```
if (car = = 'a')
sentencia 1   printf("el carácter es a \n ");
              else
sentencia 2   printf("el carácter NO es a \n );
```

Finalizada cada sentencia se prosigue con la línea siguiente del programa.(*)

Sentencias de Control de Flujo de Ejecución

Sentencia if …else múltiple

```
If (condición 1)
   {
     sentencia 1;
   }
  else if (condición 2)
          {
            sentencia 2;
           }
          else if (condición 3)
                  {
                    sentencia 3;
                  }
                   else if (….)
                          …….
                    [else
                          sentencia n;]
```

Se evalúa la condición 1 si es true se ejecuta la sentencia 1, si el false se evalúa la condición 2 si es True se ejecuta la sentencia 2 , si es false se evalúa la condición 3 y así sucesivamente. Si ninguna condición se cumple sigue por el último else ejecutándose la sentencia n. Puede ser la sentencia n vacía en tal caso se elimina el último else. (*)

Sentencias de Control de Flujo de Ejecución

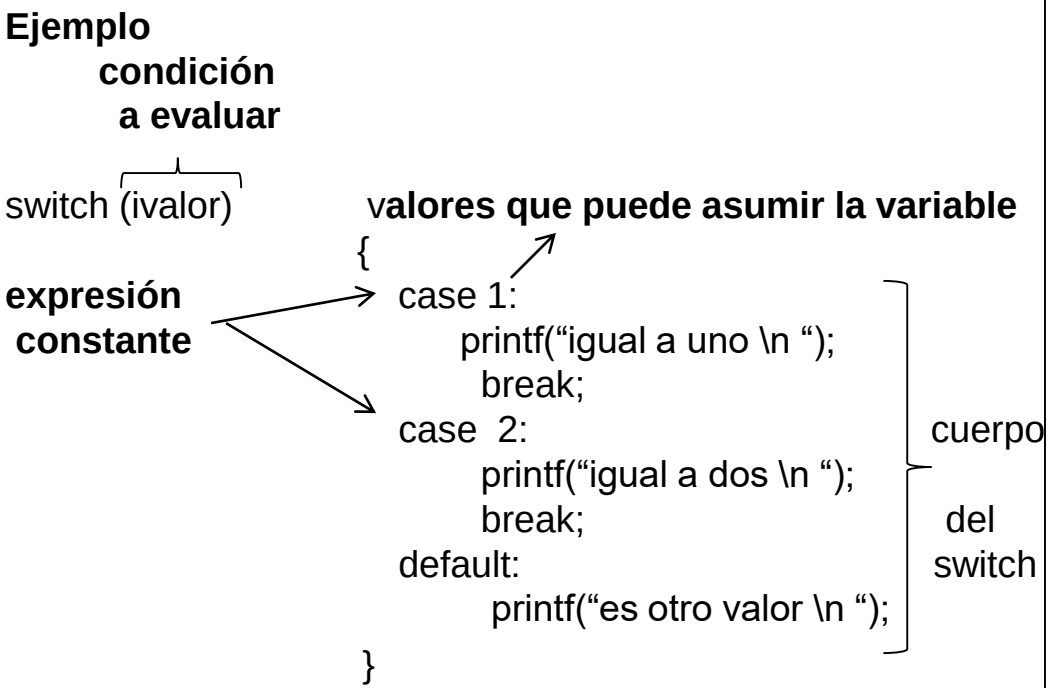
Sentencia switch

Permite dividir el flujo de ejecución en función del valor de una determinada variable. La sentencias Switch es especialmente útil cuando se requiere dividir el flujo en varios caminos mutuamente excluyente.

Su forma general es:

```
Switch (variable)
        {
          Case valor1:
                  Sentencias 1;
                  Break;
           Case valor2:
                  Sentencias 2;
                   Break;
           Case valor3:
                   Sentencias 3;
                   Break;
             .
             .
             .
            Default:
                   Sentencias n ;
        }
```

Ejemplo



Sentencias de Control de Flujo de Ejecución

La sentencia **switch** se basará en la evaluación del valor de la variable a analizar. Dentro del bloque del **switch** existirán diversas etiquetas del tipo **case** y quizás una del tipo default. El flujo de ejecución saltará a la etiqueta **case** que corresponda al valor de la variable evaluada en la cabecera del **switch**; en caso de no coincidir el valor de la variable con ninguno de los valores propuestos, saltará a la etiqueta **default**(si es que hay alguna). Si la variable toma el valor de 1 se ejecutarán las sentencias 1 hasta encontrar un **break** que permite salir del control del flujo de **switch.**

Sentencia de bucle

En ciertas ocasiones se requiere que cierta porción del código escrito sea repetido una cierta cantidad de veces o hasta que se cumpla una determinada condición. C++ cuenta con varias sentencias para cumplir con este fin.

Sentencia While

La sentencia while(que significa “mientras”) indica que una sentencia o grupo de sentencias debe repetirse mientras que una expresión asociada sea verdadera.

Su forma general es:

```
While (condición)
    {
       sentencias;
     }
```

Ej.:

```
while (ivalor != 0)
      {
        printf(“Ingrese un número \n “);
        scanf(“%i”,&ivalor);
      }
```

condición

cuerpo del while

Mientras la condición se cumple se ejecutan las sentencias, cuando se deje de cumplir la condición finaliza el bucle del while. **El bloque de código(sentencias) a repetir no se ejecuta nunca si la expresión es falsa la primera vez.**

Sentencias de Control de Flujo de Ejecución

Sentencia do … While

A diferencia del **while** en cierta circunstancias en necesario ejecutar al menos una vez las sentencias y luego evaluar la expresión. Para esto existe la sentencia llamada **do….while.**

Su forma general es:

```
do
    {
       sentencias;
     }
While (condición);
```

Ej.:

```
do
      {
        printf("Ingrese un número \n ");
        scanf("%i",&ivalor);
      } while  (ivalor != 0);
```

cuerpo del do.. while

punto y coma

condición

Sentencia for

En ciertas ocasiones existirá la necesidad de ejecutar un bloque de código un número determinado de veces, para esto se puede crear un contador que incrementa su valor en uno bucle tras bucle, luego la expresión evaluaría dicho contador a un número determinado para ver si se alcanzó el número de repeticiones deseado.

Su forma general es:

```
for (inicialización;condición;incremento)
     {
        sentencias;
     }
```

Sentencias de Control de Flujo de Ejecución

La inicialización se ejecuta una sola vez.
La condición se evalúa antes de ingresar a cada bucle.
El incremento se ejecuta al finalizar cada bucle.
Las sentencias se ejecuta siempre que se cumpla la condición.

Ej.:

sentencia inicial | **condición fin del bucle** | **incremento**

```
for (ivalor = 0;  ivalor < 10;     ivalor++)
        {
          printf("El nro mostrado es el %i \n ", ivalor);   } cuerpo del for
        }
```

Bucle Infinitos

Todo programa que utilice sentencias de repetición es posible que ingrese en bucles infinitos. Esto significa, por ejemplo, que la condición de un while sería siempre cierta y nunca se saldría del mismo, Por lo que el programa nunca terminaría.