**Learning Objectives**

**Content**

Top of Form



Bottom of Form

[Previous item](https://virtualcampus.pupr.edu/webapps/blackboard/execute/displayLearningUnit?course_id=_18759_1&content_id=_817362_1)Page 2 of 10 [Next item](https://virtualcampus.pupr.edu/webapps/blackboard/execute/displayLearningUnit?course_id=_18759_1&content_id=_817364_1)

**Upon completion of this module, you will be able to:**

1. Diferenciar las estructuras de repetición de otros elementos de programación.
2. Utilizar estructuras de repetición para resolver problemas de programación.
3. Crear declaraciones que incluyen estructuras de repetición en C++ para resolver un problema sencillo de programación.

Repeticiones en programas

Lesson Introduction:

Ya vistos los operadores de incremento y decremento en el módulo 7, podemos entender que existe un método para incrementar o decrementar valores que facilita la creación de repeticiones a la vez que proveen un modo de controlar el número de repeticiones.



Para repasar:

++ anade uno al valor y - – quita uno al valor Vean el contenido en detalle para repasar en el módulo anterior C++ provee en total siete estructuras. Estudiaremos ahora las tres estructuras que nos permiten crear repeticiones en nuestro código. La primera que estudiaremos es la estructura while.

Subtema: While

E*l while* hace que una instrucción se repita dependiendo de la veracidad de una condición. Esta es una estructura de “pre-test” porque prueba la condición antes de ejecutar. Si la condición no es cierta, no ejecutará. El formato de una expresión *while* es como sigue:

while (condición)

Instrucción;

La instrucción puede ser una o múltiples, en cuyo caso se recogen las líneas sujetas a la condición dentro de corchetes { }.

En ese caso, el formato será:

while (condición)

{ instrucción;

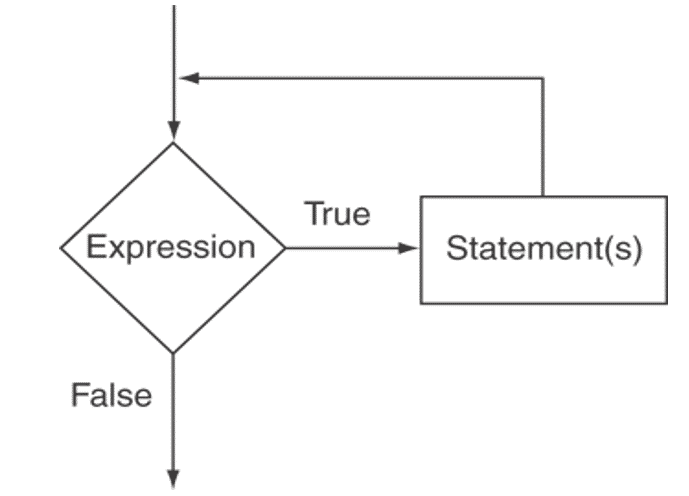
instrucción;

instrucción;…

}

La condición es evaluada y la instrucción o instrucciones ejecutan si la condición es cierta.

La representación gráfica (tomada de Gaddis) es:



La estructura while es útil para validación de datos suministrados por el usuario o para controlar el número de repeticiones a través de un valor centinela.

SUBTEMA DO/ WHILE

El *do while* ejecuta una o varias instrucciones y luego verifica si la condición es cierta o falsa. Es una estructura de post-test porque verifica la condición luego de ejecutar. Esta estructura siempre ejecutará por lo menos una vez. El formato de la expresión es como sigue:

do

instrucción;

while (condición);

Si es más de una instrucción, entonces se requiere encerrarlas entre corchetes { }.

do

{ instrucción;

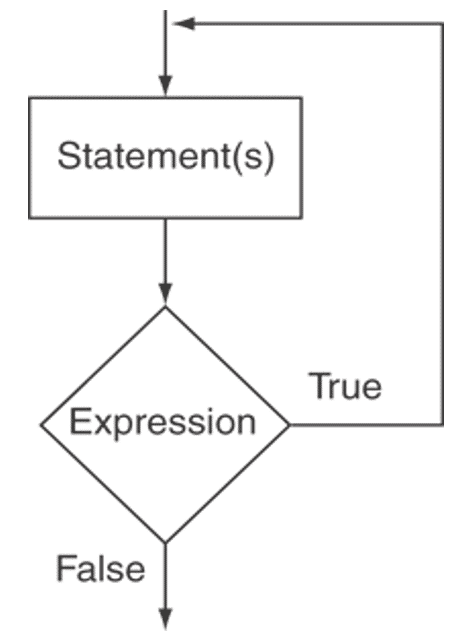
instrucción;

instrucción;…

}

while (condición);

Gaddis provee la siguiente visualización del do while:



SUBTEMA FOR

El *for* hace que se repita una o más acciones un número de veces determinado por un contador. El *for* integra dentro de la estructura misma un contador y un incremento que controlan el número de repeticiones. Muchas veces el número de repeticiones del for es determinado por variables o insumo externo. Por ejemplo, si el usuario quiere repetir una acción siete (7) veces, puedes usar un prompt para preguntar el número de repeticiones, asignar el valor a una variable y utilizar esa variable para formar la condición ligada al for.

El formato es:

for (count = 0; count<var;count++)

instrucción;

ó

for (count = 0; count<var;count++)

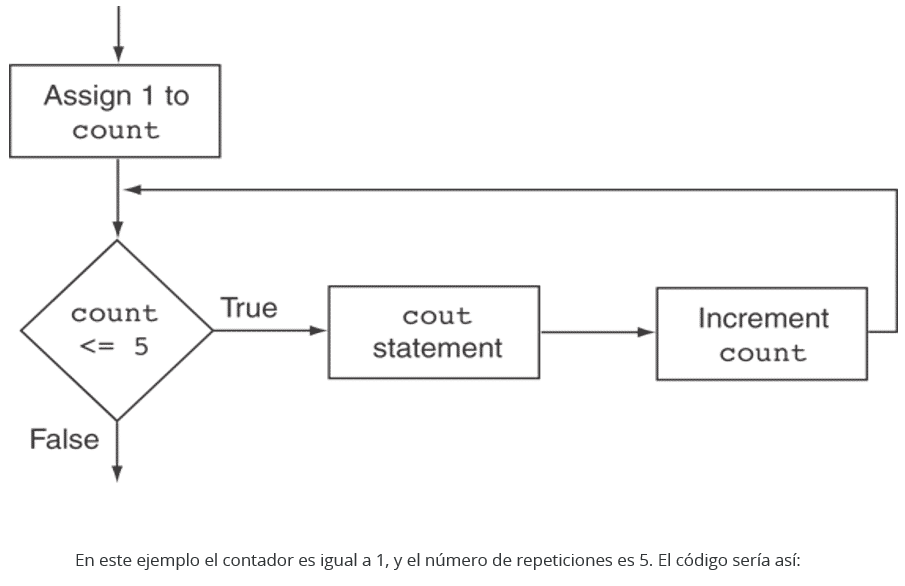
{ instrucción;

instrucción;

instrucción;…

}

la visualización del for provista por Gaddis es:



for (count=1; count<=5; count++)

cout<<”Expresión literal”;

Es importante inicializar el contador antes o durante el for. El ejemplo anterior presume que el contador “count” ya ha sido declarado. También es importante determinar y controlar el número de repeticiones necesario. En el ejemplo anterior el contador es 1 y la condición requiere que el contador sea menor o igual a uno. Si la condición incluyera solo menor a entonces el número de repeticiones se reduciría. Se puede usar un valor de contador distinto, por ejemplo, count =0 o usar diferentes operadores relacionales para cambiar el número de repeticiones.

Todas las estructuras de repetición requieren que se controle el número de repeticiones. El caso de while y do while, el número de repeticiones tiende a ser pre-determinado. La estructura for se utiliza frecuentemente para conteos que dependen de algún input (insumo externo). Estas todas deben tener en cuenta algún tipo de contador que frecuentemente puede ser añadido gracias [NSC1]al uso de los operadores de incremento y decremento estudiados en el módulo anterior.

Vea el video de presentación:

Las estructuras de repetición permiten ejecutar instrucciones repetidamente dependiendo del resultado de la evaluación de una condición. Las estructuras de repetición nos permiten programar instrucciones a ser repetidas un número pre-determinado de veces.  Es importante distinguir el uso de while (pre-test), do while (post-test) y for como estructuras para diferentes usos.