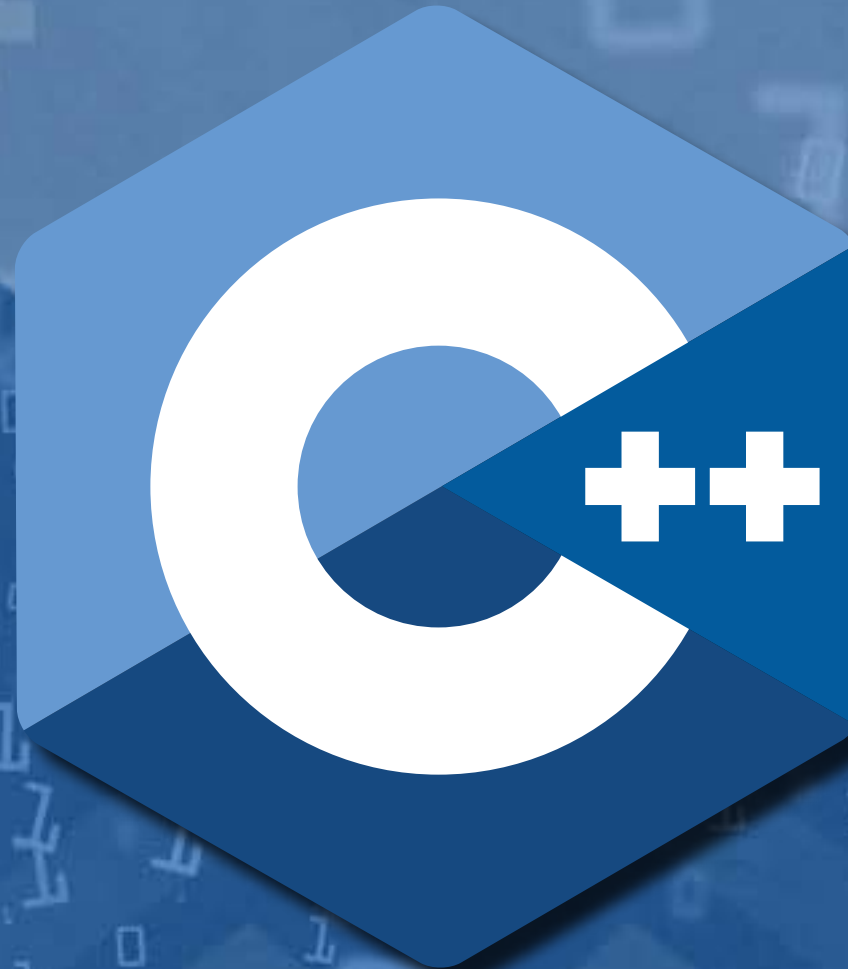


CONCEPTUALIZACIÓN DEL LENGUAJE DE PROGRAMACIÓN C++.



Material de Actividad de Aprendizaje 4



CONCEPTUALIZACIÓN DEL LENGUAJE DE PROGRAMACIÓN C++.

Material Actividad de Aprendizaje 4.

4. Estructuras de control por Iteración: bucles.

4.1 El bucle while.

4.2 El bucle for.

4.3 El bucle do.



C CONCEPTUALIZACIÓN DEL LENGUAJE DE PROGRAMACIÓN C++.

Material Actividad de Aprendizaje 4.

◆ 4. Estructuras de control por Iteración: bucles.

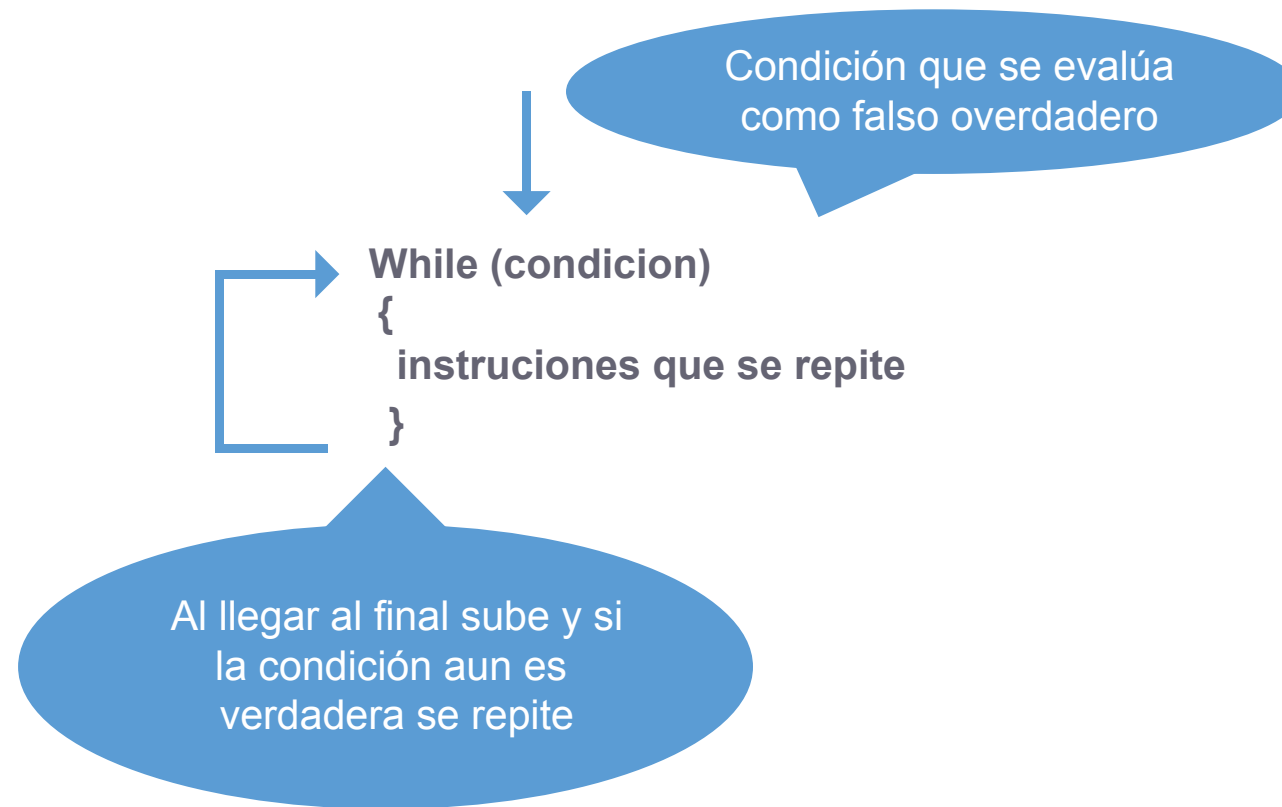
Las computadoras tienen un gran poder de cálculo, pero aumentan notablemente su rendimiento al realizar la misma tarea varias veces, realizándolas con gran velocidad, fiabilidad y precisión. Para el hombre estas tareas repetitivas son tediosas y hasta difíciles de realizar, sin embargo nuestro mundo está lleno de estas tareas.

Situación de ejemplo; el instructor de biofísica les pide a sus aprendices que para iniciar la sesión de entrenamiento repitan algunas rutinas, la forma de afrontar esas rutinas y la condición para terminarlas es lo que da origen al concepto de bucle y a los diferentes tipos de bucles.

Un bucle es cualquier estructura de programa que permite repetir una instrucción o un conjunto de instrucciones. Las dos cosas a tener en cuenta en el diseño de un bucle son las instrucciones que va a repetir (cuerpo del bucle) y como va a terminar el bucle, lo que determinara las veces que se repita.



C CONCEPTUALIZACIÓN DEL LENGUAJE DE PROGRAMACIÓN C++.





C CONCEPTUALIZACIÓN DEL LENGUAJE DE PROGRAMACIÓN C++.

Usando el editor de C++ codeblocks escriban el siguiente código.

```
#include <iostream>
// Se incluye la librería iostream que maneja los flujos de entrada/salida

using namespace std;
//se usa el espacio de nombre std para evitar confusiones en los nombres

int main() //la funcion principal
{
    int a=0;

    while(a<10)
    {
        cout<<"Se repite "<<a+1<<" veces"<<endl;
    }
}
```



C CONCEPTUALIZACIÓN DEL LENGUAJE DE PROGRAMACIÓN C++.

Compila y observen la salida.

Ahora inserten la instrucción a++ como se observa debajo

```
#include <iostream>
// Se incluye la librería iostream que maneja los flujos de entrada/salida.

using namespace std;
//se usa el espacio de nombre std para evitar confusiones en los nombres.

int main() //la funcion principal
{
    int a=0;

    while(a<10)
    {
        cout<<"Se repite "<<a+1<<" veces"<<endl;
        a++;
    }
}
```



C CONCEPTUALIZACIÓN DEL LENGUAJE DE PROGRAMACIÓN C++.

Y observen la salida ahora.

En el ejemplo anterior la variable (a) se comporta como variable de control porque su valor determina la ejecución o no del bucle. Esta variable pasa por varios estados durante el accionar del bucle.

- **Inicialización:** la variable se inicializa con un valor, en este caso $a=0$ podría ser un valor diferente.
- **Evaluación:** la variable se evalúa en la expresión o condición que encabeza el bucle while ($a < 10$)
- **Actualización:** la variable en la iteración o pasada que es cuando se ejecutan las instrucciones del cuerpo del bucle, actualiza su valor dando lugar a una nueva evaluación, si se cumple aun la condición se repite, de lo contrario se termina el bucle.

Deben tener cuidado en la construcción de un bucle observando que siempre se tenga una condición de parada o fin del bucle, porque de lo contrario se crea un “bucle infinito” es decir sin fin.

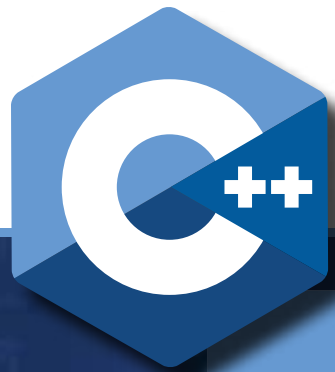


C CONCEPTUALIZACIÓN DEL LENGUAJE DE PROGRAMACIÓN C++.



Un error muy común con el while, es que digiten un punto y coma (;) después de la (condición) ejemplo while(condición);

Pese a que se puede dejar sin llaves la estructura (if) es buena idea mientras se acostumbra a colocarla en la sintaxis.



C CONCEPTUALIZACIÓN DEL LENGUAJE DE PROGRAMACIÓN C++.

Una instrucción while puede contener dentro de su cuerpo otras estructuras de control como, bloques if u otros bloques while así:

Lee más acerca de este ciclo en el siguiente link:
http://es.wikipedia.org/wiki/Bucle_while

```
int main() //la funcion principal
{

    int a=1;
    int primo=0;
    int residuo=0, contador=0;

    cout<<"Digite un numero y averigue si es primo "<<endl;
    cin>>primo;

    while(a<primo)
    {
        residuo=primo/a;

        if(residuo*a==primo)
        {
            contador++;
        }

        a++;
    }

    if(contador<=2)
    {
        cout<<"El numero " << primo<<" es primo" <<endl;
    }
    else
    {
        cout<<"El numero " << primo<<" no es primo" <<endl;
    }
}
```



C CONCEPTUALIZACIÓN DEL LENGUAJE DE PROGRAMACIÓN C++.

◆ 4.2 El bucle for.

El bucle for es un bucle manejado por contador en la mayoría de los lenguajes de programación, sin embargo este no es completamente cierto para C++, es mucho más potente y flexible, proporciona más control sobre la inicialización y el incremento de las variables de control.

Inicializa el
contador

Evalúa la
condición

Incrementa el
contador

```
For (a = 1 ; a <= primo; a ++)  
{  
    residuo = primo / a ;  
  
    if (residuo * a ==primo)  
  
        {  
            Contador++;  
        }  
}
```



C CONCEPTUALIZACIÓN DEL LENGUAJE DE PROGRAMACIÓN C++.



Como lo pueden ver arriba en la imagen, el control que ofrece el bucle for, es superior al del bucle while, porque se puede inicializar directamente la variable de control, además contiene la evaluación de la condición, y el incremento de la variable de control, es decir en teoría no se debería poder crear un bucle infinito por equivocación.



C CONCEPTUALIZACIÓN DEL LENGUAJE DE PROGRAMACIÓN C++.

Este es el código de cómo quedaría el ejemplo anterior usando un bucle for.

```
int main() //la funcion principal
{

    int a=1;
    int primo=0;
    int residuo=0, contador=0;

    cout<<"Digite un numero y averigue si es primo "<<endl;
    cin>>primo;

    for(a=1;a<=primo;a++)
    {
        residuo=primo/a;

        if(residuo*a==primo)
        {
            contador++;
        }
    }

    if(contador<=2)
    {
        cout<<"El numero " << primo<<" es primo" <<endl;
    }
    else
    {
        cout<<"El numero " << primo<<" no es primo" <<endl;
    }
}
```

Puedes aprender más leyendo el artículo en el siguiente link: http://es.wikipedia.org/wiki/Bucle_for



C CONCEPTUALIZACIÓN DEL LENGUAJE DE PROGRAMACIÓN C++.

◆ 4.3 El bucle do.

El bucle do al igual que el ciclo while se controla con decisiones que manejan un número variable de repeticiones, pero a diferencia del while que ejecuta o repite una serie de instrucciones cero o más veces, el bucle do se ejecuta por lo menos una vez.

Inicio del bloque do

do {

Sentencias que se repiten

} while (exp)

Expresión que controla el bucle

En este ejemplo pueden ver en acción el bucle do, y aunque la condición no se cumple la acción se realiza una vez.



C CONCEPTUALIZACIÓN DEL LENGUAJE DE PROGRAMACIÓN C++.

```
#include <iostream>

using namespace std;

int main()
{
    int a=0
    do{

        cout << "Hello world!" << endl;

    }while(!a>=0);

    return 0;
}
```

Puedes consultar más sobre el bucle do while en el siguiente artículo en:
http://es.wikipedia.org/wiki/Bucle_do