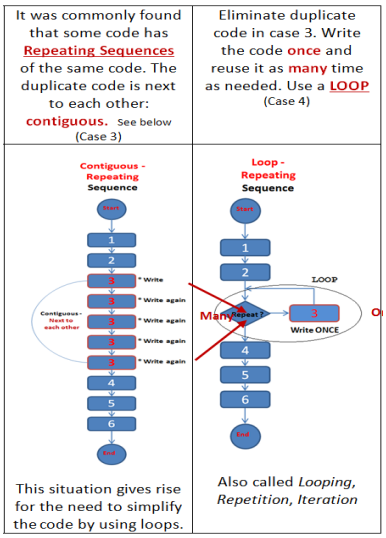
Repeating Sequences and the 'While' and 'Do While' Loops

"Where you see CODE REUSE, you see a good programmer"
Quote Professor Schwarz

LOOP



Loop, Repetition, Iteration

With **looping**, you avoid having to type or copy a code block multiple times. You only have to *write it once*.
The Criteria/Conditions determine the number of times the single code block is repeated.

The Mechanics of programming loops. There are ***three*** (3) '***Loop Elements'*** important to coding a loop.

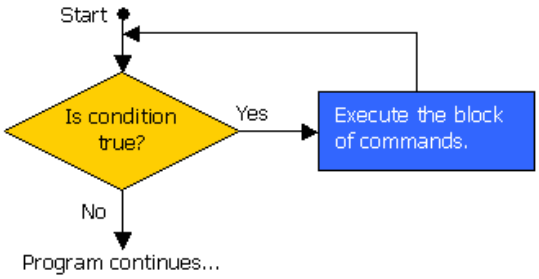
- A *Variable* with a start value
- Criteria/Condition that checks the *variable* to determine when to end the loop or to execute code block again
- *Variable* modification

TYPES OF LOOPS

There are two types of loops: **Pretest** and **Posttest**.

A pretest simply means the criteria/condition is checked **before** the code block is executed.
A posttest simply means the criteria/condition is checked **after** the code block is executed at least once.

** WHILE LOOP



C++ Keyword: **while**

The 'while' loop is a pretest loop. The condition is at the top of the 'while' loop. Pretest means the condition/criteria is checked before the code block is executed. If the condition/criteria is initially FALSE, the block of code is NEVER run.

The 'while' loop is easy to use. Look at the location of the three (3) 'loop elements'

Basic Syntax:

```
… declare variable = initial value;
while ( condition – check variable ) {
    … modify variable
    …  code block..
}
```

Code Example 1 – **Try it**

```
#include <iostream>
using namespace std;
int main( ) {
   // Sample program to print the value of 'i' at each loop..

   int i = 0;  // 1st element of loop - declare an initialize value

   while ( i < 10 ) {  // 2nd element of loop - condition/criteria
      cout << "i = " << i << endl;
      i++; // 3rd element of loop - update value of condition variable
   }

   system ("pause");
   return 0;
}
```

Code Example 2 – **Try it**

```
#include <iostream>
using namespace std;
int main ( ) {
   // Sample INFINITE LOOP
   // Condition is always true

   int i = 0;  // 1st element of loop - declare and set initial value

   while ( i >= 0 ) {  // 2nd element of loop - condition
      cout << "i = " << i << endl;
      i++; // 3rd element of loop - update value
   }

   system ("pause");
   return 0;
}
```

Programmer Alert – Not paying proper attention to the **criteria/conditions** in your 'loop' statements is an easy way to ruin your program. Always double, triple check your conditions.

Code Example 3 – **Try it**

```
#include <iostream>
using namespace std;

int main ( ) {
   // Calculate grade average

   int numberGrades = 0;
   int sumGrades = 0;
   int aGrade = 0;

   cout << "How many grades to enter: ";
   cin >> numberGrades;

   int i = 1; // 1st element of loop - declare and set initial value
   while ( i <= numberGrades ) {  // 2nd element of loop - condition
          cout << "Enter Grade: ";
          cin >> aGrade;

          sumGrades = sumGrades + aGrade;
          i++; // 3rd element of loop - update value
   }

   cout << "Grade Average = " << sumGrades/numberGrades << endl;

   system ("pause");
   return 0;
}
```

Programmer Alert: In a conition: 0 is always false, Any non zero value is always true.

Nested 'while' loop

You can place any valid statement inside the body of a 'while' loop, even another 'while' loop.

Code Example 4 – **Try it**

```
#include <iostream>
using namespace std;

int main ( ) {
      // Print 10 by 10 square of diamonds
      // Declare and initialize loop variable.
       int i = 0;
       int j = 0;
       while ( i < 10 ){
           i++;
            while ( j < 10 ) {
               j++;
               cout << "*";
            }
            j=0;
           cout << endl;
       }
      system ("pause");
      return 0;
}
```

The 'WHILE' loop essentially does the same thing as the 'FOR' loop.

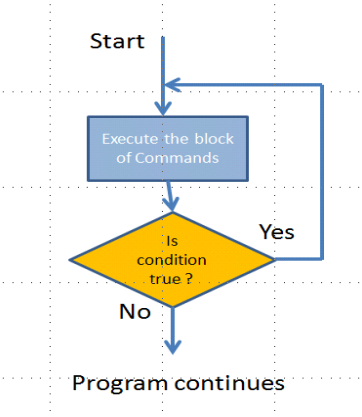
An advantage of the 'FOR' loop is that all three (3) 'loop elements' are together.

A disadvantage of the 'WHILE' loop is that the three (3) 'loops elements' are separate.

Over all, either one is OK to use.

Video – Watch me – While loop: http://www.youtube.com/watch?v=KLKhsaOPnLk

** DO WHILE LOOP



C++ Keywords: **do while**

The 'do while' loop is a posttest loop. The condition is at the bottom of the 'do while' loop. Posttest means the condition/criteria is checked AFTER the code block is executed once. If the condition/criteria is initially FALSE, the block of code is only executed once. Again: THE CODE IS ALWAYS EXECUTED AT LEAST ONCE.

The 'do while' loop is easy to use. Look at the location of the three (3) 'loop elements'

Basic Syntax:

```
... declare variable = initial value;
do {
   ... code block..
   ... modify variable
while ( condition – check variable ) ; // ← Note the ; at end of this statement
}
```

Code Example 5 – **Try it - Runs ONLY ONCE**

```
#include <iostream>
using namespace std;

int main( ) {
   // Sample program to print the value of 'i' at each loop..

   int i = 6;  // 1st element of loop - declare an initialize value
   do
   {  // 2nd element of loop - condition/criteria
      cout << "i = " << i << endl;
      i = 11; // 3rd element of loop - update value of condition variable - to 11
   } while ( i < 10 );

   system ("pause");
   return 0;
}
```

Code Example 6 – **Try it – RUNS Many Times..**

```
#include <iostream>
using namespace std;
int main ( ) {
   // Sample INFINITE LOOP
   // Condition is always true

   int i = 0;  // 1st element of loop - declare and set initial value
   do
   {  // 2nd element of loop - condition
      cout << "i = " << i << endl;
      i++; // 3rd element of loop - update value
   } while ( i <= 10 );

   system ("pause");
   return 0;
}
```

Video – Watch me – Do While loop: http://www.youtube.com/watch?v=yRdPe2acogw

Assignment

1) Draw flow charts for the code examples 1, 3 and 5 above in this lecture. Use a program to draw them.

2) Variable Initialization

Look at program Example 3 above

```
// Calculate grade average
```

Change this line of code: `int sumGrades = 0;`
to: `int sumGrades;`

Run the code with the '=0' removed. Click on 'continue' if asked to do so.

Explain what happens with the '=0' removed.

Explain why you need to initialize variables.

Hint: When you declare a variable in C++, what is still in the bytes allocated for the variable.

3) Write a program that prompts the User if they want to convert temperatures.

'Fahrenheit to Celsius' or 'Celsius to Fahrenheit'.

(This was done in a previous assignment) . Be sure to use decimal numbers only.

Take this program and place it inside a 'while' loop, so it can be repeated… as long as they need it.

The purpose of the while loop is to ask the user if they want to do it again…

4) Modify this program to use while loops. It determines prime numbers.

```
#include <stdio.h>
#include <iostream>
using namespace std;

 int main(void)
{
  int i = 0,j = 0,k = 0;
  double n = 0;

  cout << "Enter a number: ";
  cin >> n ;

  cout << "The prime numbers form 2 to " << n << "are " << endl;

  for(i=2;i<=n;i++)
  {
    k=0;
    for(j=1;j<=i;j++)
       if(i%j==0) k++;
    if(k==2)
    cout << ",  " << i;
  }
  return 0;
}
```

5) Use program Example 1 above.

```
// Sample program to print the value of 'i' at each loop..
while ( i < 10 ) {  // 2nd element of loop – condition/criteria
```

Modify the condition in example 1 from (i < 10) to:

- ( )
- (0)
- (1)

- (42)
- (true)
- (false)
- ((i * 2) < 10)
- (i = 7)
- ((i + 1) < 10)

Run the program with each condition. What are the results.

Explain what each new condition means.

6) Questions

a) Why use a 'while' loop instead of a 'for' loop ?

b) Why use a do while loop ? General answer.

c) What is the most common reason to use a do while loop ? Research the answer on internet.

Not the same answer as question b

d) Are there other types of loops in C++, besides 'for', 'while' and 'do whiles' ?

If so, what are they ?

e) Why use a loop ?

e) What does contiguous mean ? Give an example…

Hello Class,

Did you know I have an in-class version of CIS 118 on Monday and Wednesday, from 8:10 to 11PM every week.

The online students are welcome to come and sit in. It will help you tremendously.

Sincerely,

Professor Schwarz