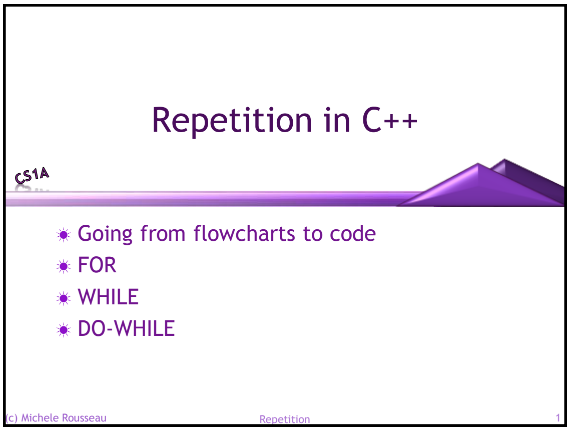
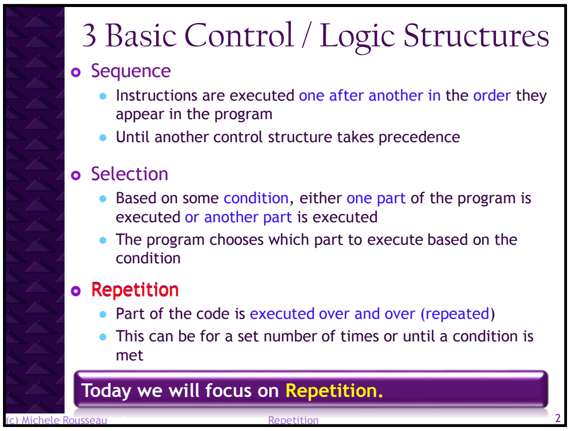


Repetition in C++

CS1A

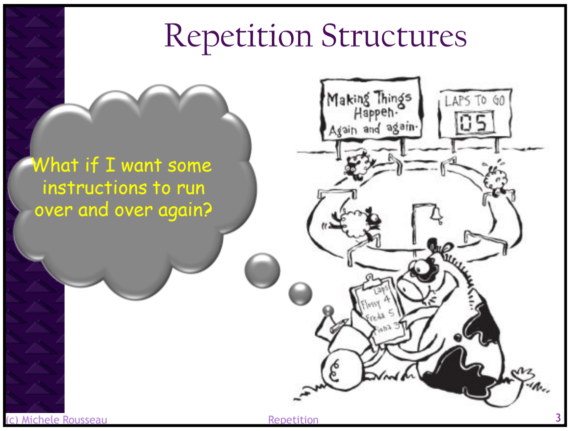
- Going from flowcharts to code
- FOR
- WHILE
- DO-WHILE



3 Basic Control / Logic Structures

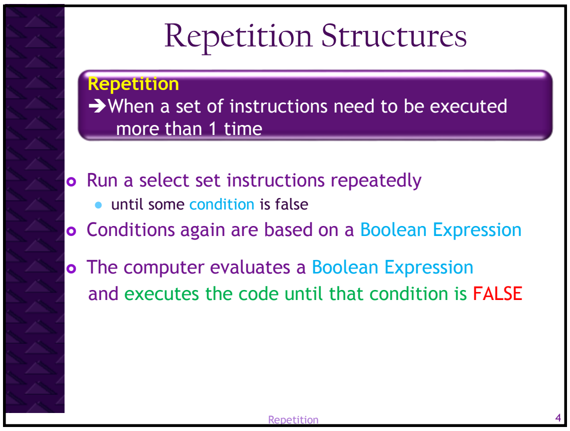
- Sequence
  - Instructions are executed one after another in the order they appear in the program
  - Until another control structure takes precedence
- Selection
  - Based on some condition, either one part of the program is executed or another part is executed
  - The program chooses which part to execute based on the condition
- Repetition
  - Part of the code is executed over and over (repeated)
  - This can be for a set number of times or until a condition is met

Today we will focus on Repetition.



Repetition Structures

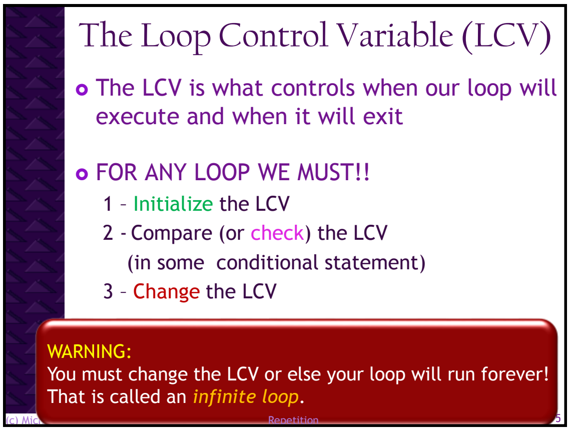
What if I want some instructions to run over and over again?



Repetition Structures

Repetition
➔When a set of instructions need to be executed more than 1 time

- Run a select set instructions repeatedly
  - until some condition is false
- Conditions again are based on a Boolean Expression
- The computer evaluates a Boolean Expression and executes the code until that condition is FALSE

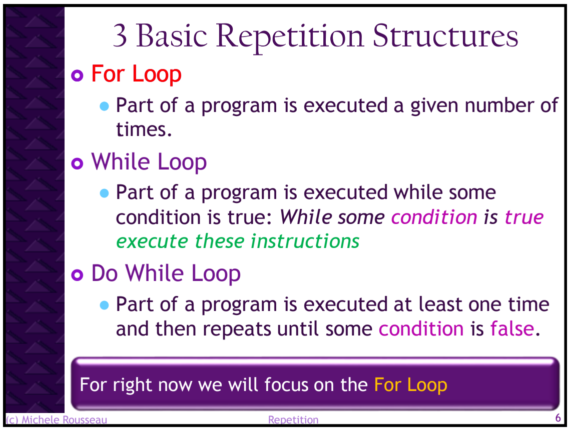


The Loop Control Variable (LCV)

- The LCV is what controls when our loop will execute and when it will exit
- FOR ANY LOOP WE MUST!!

  1 - Initialize the LCV
  2 - Compare (or check) the LCV
  (in some conditional statement)
  3 - Change the LCV

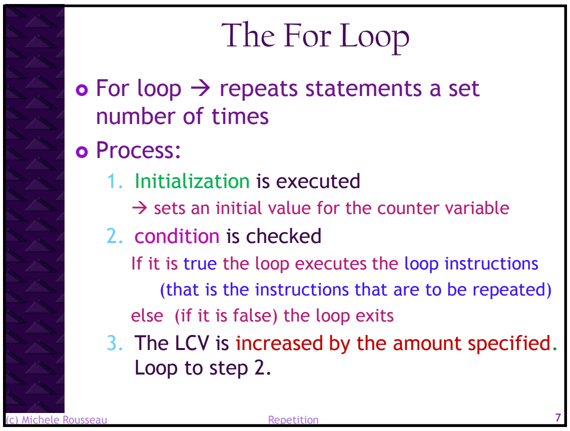
WARNING:
You must change the LCV or else your loop will run forever!
That is called an *infinite loop*.



3 Basic Repetition Structures

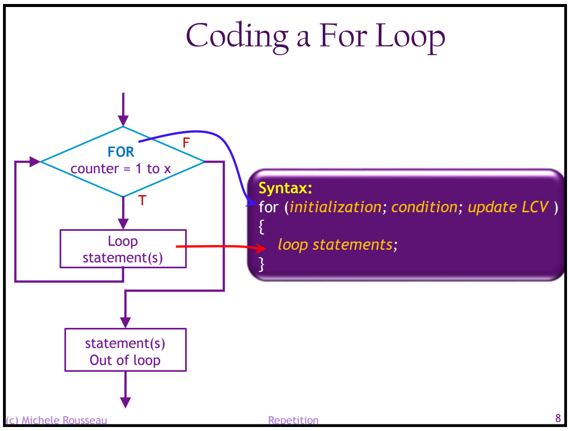
- For Loop
  - Part of a program is executed a given number of times.
- While Loop
  - Part of a program is executed while some condition is true: *While some condition is true execute these instructions*
- Do While Loop
  - Part of a program is executed at least one time and then repeats until some condition is false.

For right now we will focus on the For Loop



The For Loop

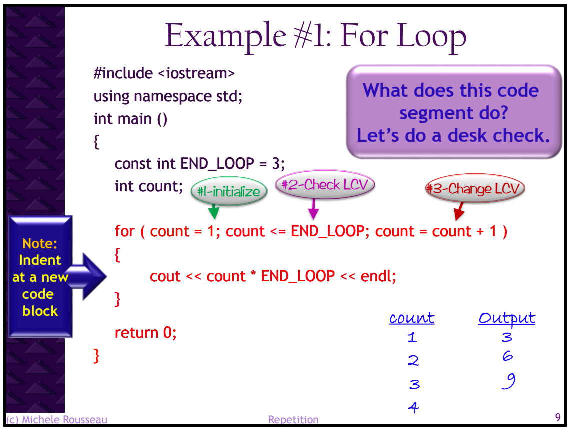
- For loop → repeats statements a set number of times
- Process:
  1. Initialization is executed
     → sets an initial value for the counter variable
  2. condition is checked
     If it is true the loop executes the loop instructions
        (that is the instructions that are to be repeated)
     else (if it is false) the loop exits
  3. The LCV is increased by the amount specified. Loop to step 2.



Coding a For Loop

Syntax:

```
for (initialization; condition; update LCV )
{
    loop statements;
}
```



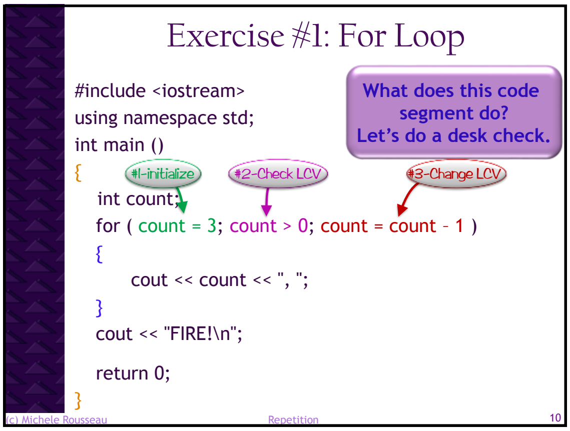
Example #1: For Loop

```
#include <iostream>
using namespace std;
int main ()
{
    const int END_LOOP = 3;
    int count;

    for ( count = 1; count <= END_LOOP; count = count + 1 )
    {
        cout << count * END_LOOP << endl;
    }
    return 0;
}
```

#1-initialize
#2-Check LCV
#3-Change LCV
Note: Indent at a new code block

What does this code segment do?
Let's do a desk check.

count	Output
1	3
2	6
3	9
4	



Exercise #1: For Loop

What does this code segment do? Let's do a desk check.

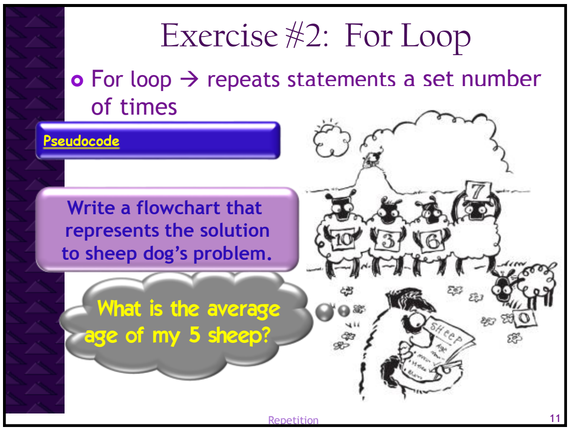
#1-initialize

#2-Check LCV

#3-Change LCV

```
#include <iostream>
using namespace std;
int main ()
{
  int count;
  for ( count = 3; count > 0; count = count - 1 )
  {
      cout << count << ", ";
  }
  cout << "FIRE!\n";

  return 0;
}
```

(c) Michele Rousseau Repetition 10



Exercise #2: For Loop

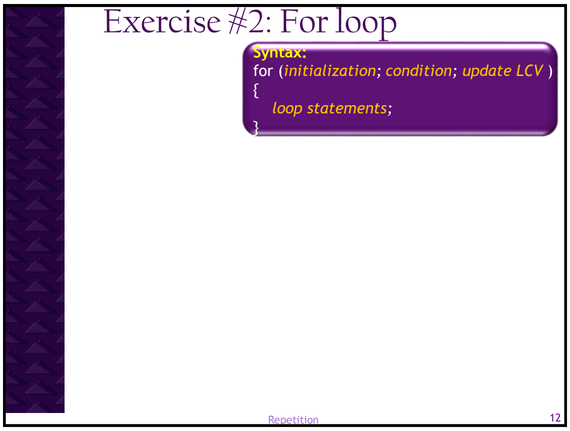
- For loop → repeats statements a set number of times

Pseudocode

Write a flowchart that represents the solution to sheep dog's problem.

What is the average age of my 5 sheep?

Repetition 11



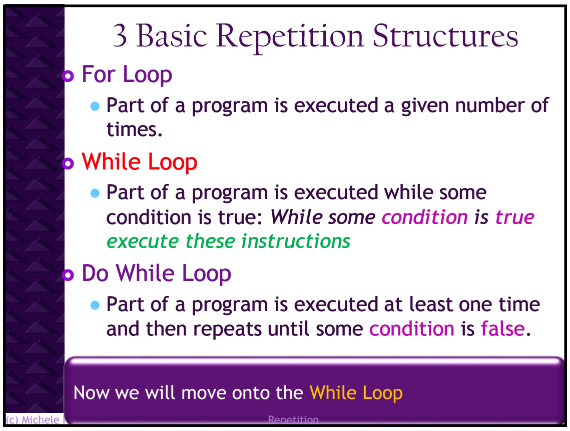
Exercise #2: For loop

Syntax:

```
for (initialization; condition; update LCV )
{
   loop statements;
}
```

Repetition 12

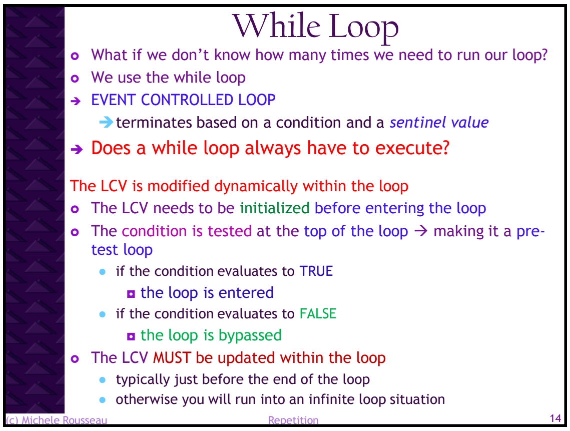
© Michele Rousseau



3 Basic Repetition Structures

- For Loop
  - Part of a program is executed a given number of times.
- While Loop
  - Part of a program is executed while some condition is true: *While some condition is true execute these instructions*
- Do While Loop
  - Part of a program is executed at least one time and then repeats until some condition is false.

Now we will move onto the While Loop

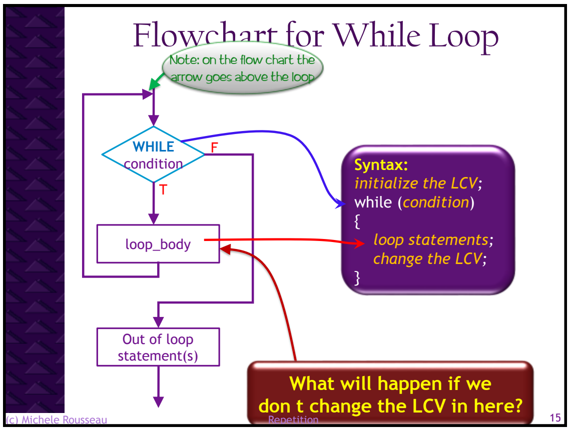


While Loop

- What if we don't know how many times we need to run our loop?
- We use the while loop
- → EVENT CONTROLLED LOOP
  - → terminates based on a condition and a *sentinel value*
- → Does a while loop always have to execute?

The LCV is modified dynamically within the loop

- The LCV needs to be initialized before entering the loop
- The condition is tested at the top of the loop → making it a pre-test loop
  - if the condition evaluates to TRUE
    - the loop is entered
  - if the condition evaluates to FALSE
    - the loop is bypassed
- The LCV MUST be updated within the loop
  - typically just before the end of the loop
  - otherwise you will run into an infinite loop situation



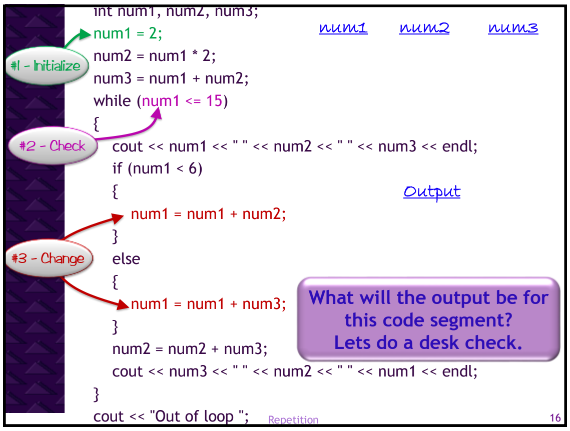
Flowchart for While Loop

Note: on the flow chart the arrow goes above the loop

Syntax:

```
initialize the LCV;
while (condition)
{
   loop statements;
   change the LCV;
}
```

What will happen if we don t change the LCV in here?



```
int num1, num2, num3;
num1 = 2;
num2 = num1 * 2;
num3 = num1 + num2;
while (num1 <= 15)
{
   cout << num1 << " " << num2 << " " << num3 << endl;
   if (num1 < 6)
   {
      num1 = num1 + num2;
   }
   else
   {
      num1 = num1 + num3;
   }
   num2 = num2 + num3;
   cout << num3 << " " << num2 << " " << num1 << endl;
}
cout << "Out of loop ";
```

#1 - Initialize

#2 - Check

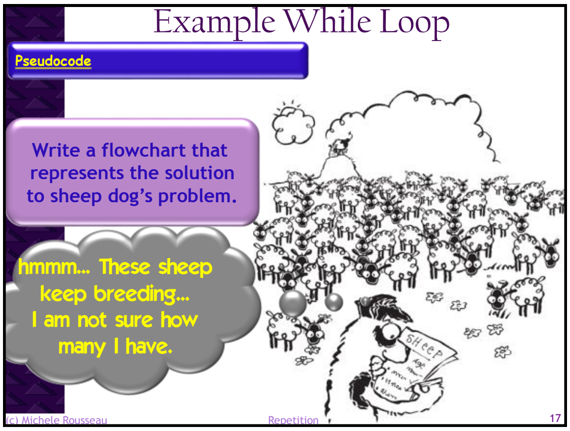
#3 - Change

num1 num2 num3

Output

What will the output be for this code segment? Lets do a desk check.

Repetition 16



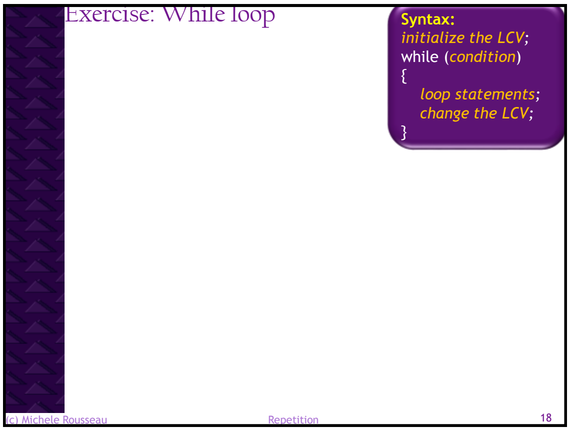
Example While Loop

Pseudocode

Write a flowchart that represents the solution to sheep dog's problem.

hmmm… These sheep keep breeding… I am not sure how many I have.

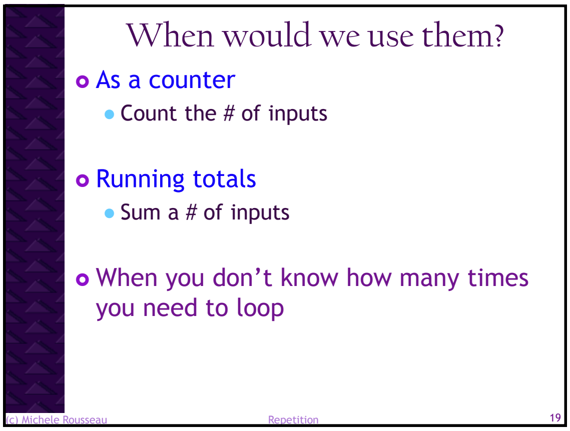
(c) Michele Rousseau Repetition 17



Exercise: While loop

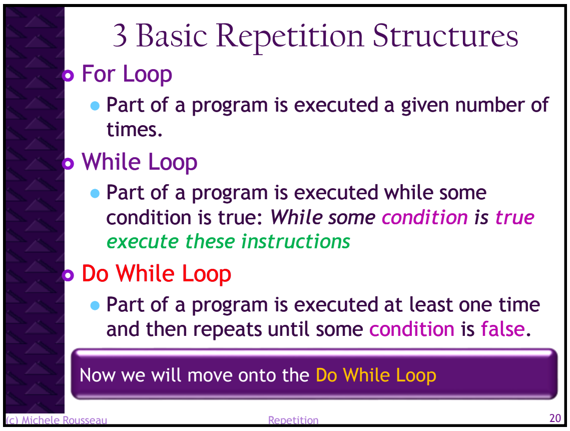
Syntax:
initialize the LCV;
while (*condition*)
{
  loop statements;
  change the LCV;
}

(c) Michele Rousseau Repetition 18



When would we use them?

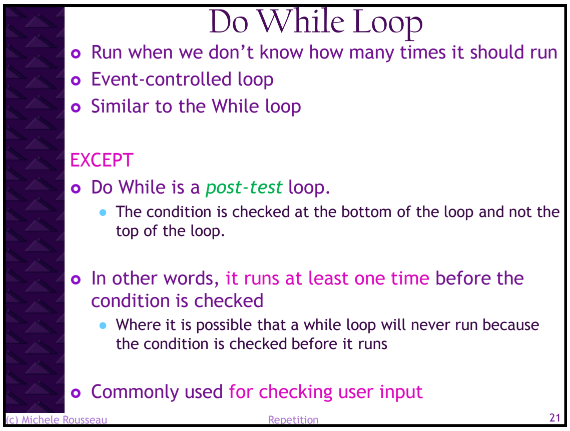
- As a counter
  - Count the # of inputs
- Running totals
  - Sum a # of inputs
- When you don't know how many times you need to loop



3 Basic Repetition Structures

- For Loop
  - Part of a program is executed a given number of times.
- While Loop
  - Part of a program is executed while some condition is true: *While some condition is true execute these instructions*
- Do While Loop
  - Part of a program is executed at least one time and then repeats until some condition is false.

Now we will move onto the Do While Loop

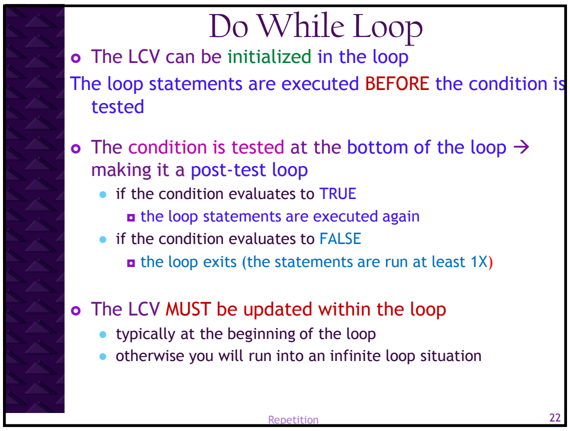


Do While Loop

- Run when we don't know how many times it should run
- Event-controlled loop
- Similar to the While loop

EXCEPT

- Do While is a *post-test* loop.
  - The condition is checked at the bottom of the loop and not the top of the loop.
- In other words, it runs at least one time before the condition is checked
  - Where it is possible that a while loop will never run because the condition is checked before it runs
- Commonly used for checking user input



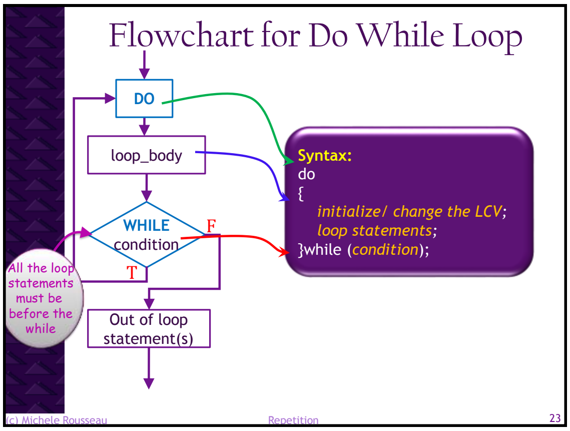
Do While Loop

- The LCV can be initialized in the loop

The loop statements are executed BEFORE the condition is tested

- The condition is tested at the bottom of the loop → making it a post-test loop
  - if the condition evaluates to TRUE
    - the loop statements are executed again
  - if the condition evaluates to FALSE
    - the loop exits (the statements are run at least 1X)
- The LCV MUST be updated within the loop
  - typically at the beginning of the loop
  - otherwise you will run into an infinite loop situation

Repetition 22

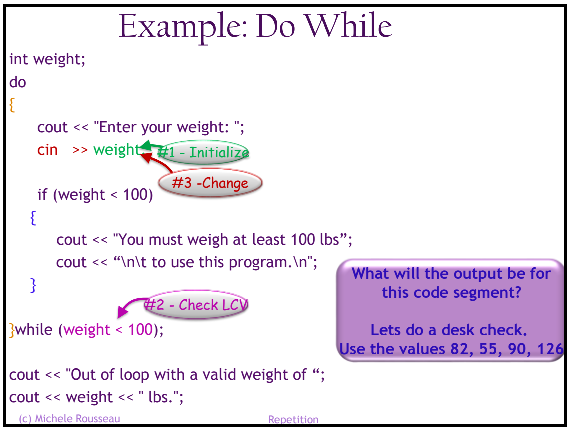


Flowchart for Do While Loop

Syntax:

```
do
{
    initialize/ change the LCV;
    loop statements;
}while (condition);
```

(c) Michele Rousseau Repetition 23



Example: Do While

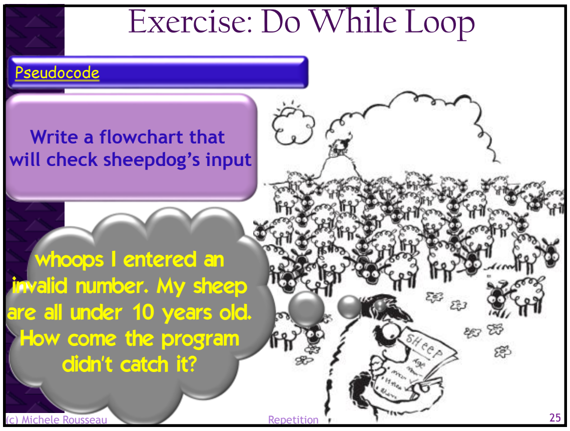
```
int weight;
do
{
     cout << "Enter your weight: ";
     cin   >> weight;
     if (weight < 100)
    {
         cout << "You must weigh at least 100 lbs";
         cout << "\n\t to use this program.\n";
    }
}while (weight < 100);

cout << "Out of loop with a valid weight of ";
cout << weight << " lbs.";
```

#1 - Initialize
#3 -Change
#2 - Check LCV

What will the output be for this code segment?

Lets do a desk check.
Use the values 82, 55, 90, 126

(c) Michele Rousseau Repetition

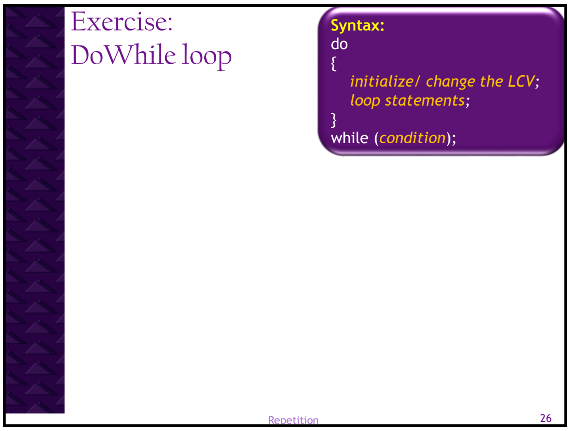


Exercise: Do While Loop

Pseudocode

Write a flowchart that will check sheepdog's input

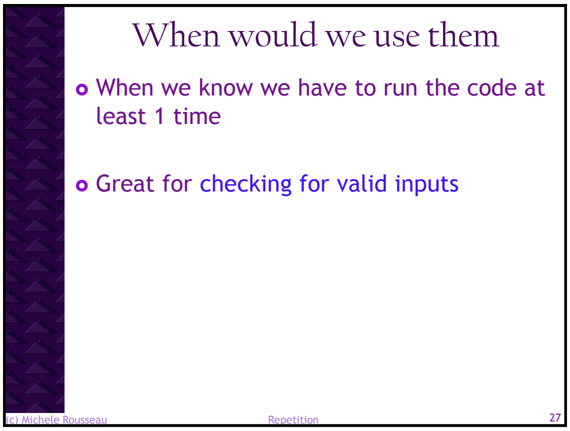
whoops I entered an invalid number. My sheep are all under 10 years old. How come the program didn't catch it?



Exercise: DoWhile loop

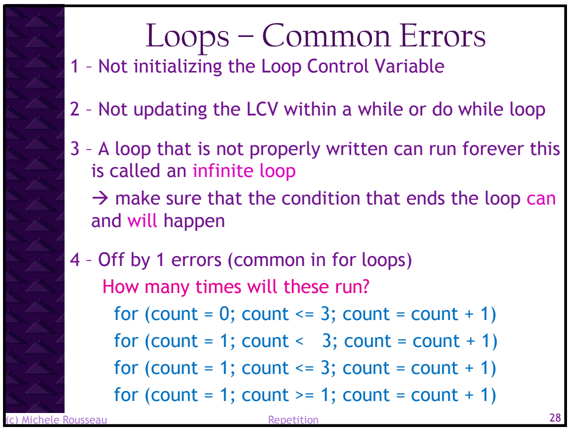
Syntax:

```
do
{
    initialize/ change the LCV;
    loop statements;
}
while (condition);
```



When would we use them

- When we know we have to run the code at least 1 time
- Great for checking for valid inputs



Loops – Common Errors

1 - Not initializing the Loop Control Variable

2 - Not updating the LCV within a while or do while loop

3 - A loop that is not properly written can run forever this is called an infinite loop

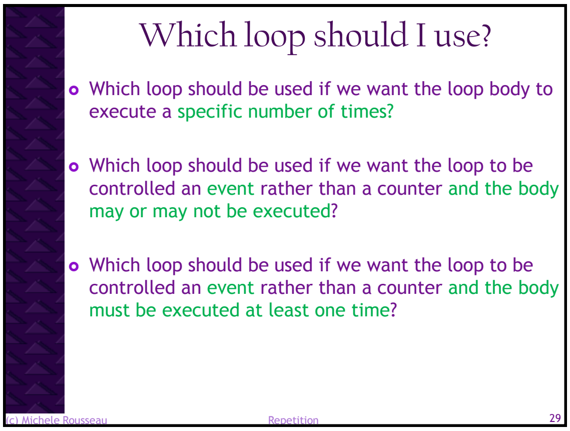
→ make sure that the condition that ends the loop can and will happen

4 - Off by 1 errors (common in for loops)

How many times will these run?

```
for (count = 0; count <= 3; count = count + 1)
for (count = 1; count <   3; count = count + 1)
for (count = 1; count <= 3; count = count + 1)
for (count = 1; count >= 1; count = count + 1)
```

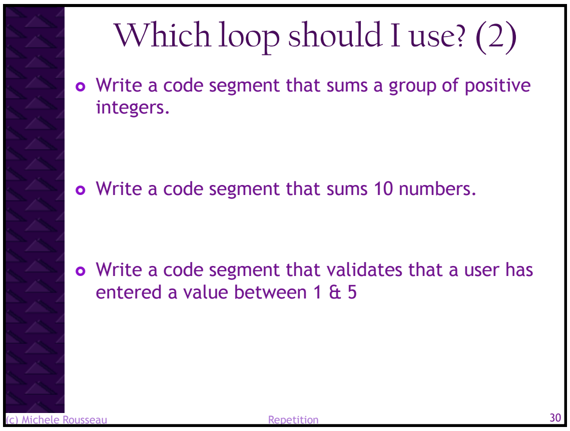
(c) Michele Rousseau Repetition 28



Which loop should I use?

- Which loop should be used if we want the loop body to execute a specific number of times?
- Which loop should be used if we want the loop to be controlled an event rather than a counter and the body may or may not be executed?
- Which loop should be used if we want the loop to be controlled an event rather than a counter and the body must be executed at least one time?

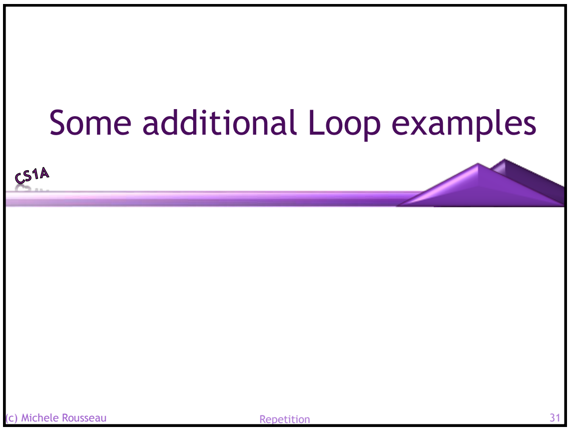
(c) Michele Rousseau Repetition 29



Which loop should I use? (2)

- Write a code segment that sums a group of positive integers.
- Write a code segment that sums 10 numbers.
- Write a code segment that validates that a user has entered a value between 1 & 5

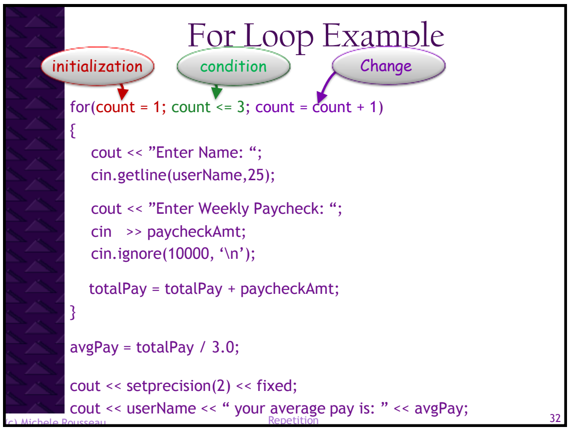
(c) Michele Rousseau Repetition 30



Some additional Loop examples

CS1A

(c) Michele Rousseau Repetition 31



For Loop Example

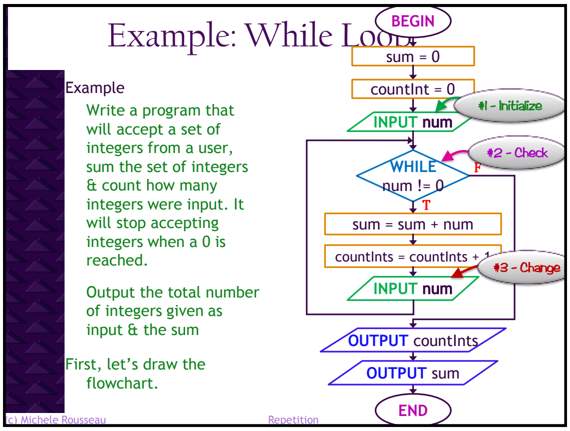
initialization

condition

Change

```
for(count = 1; count <= 3; count = count + 1)
{
    cout << "Enter Name: ";
    cin.getline(userName,25);

    cout << "Enter Weekly Paycheck: ";
    cin   >> paycheckAmt;
    cin.ignore(10000, '\n');

   totalPay = totalPay + paycheckAmt;
}
avgPay = totalPay / 3.0;

cout << setprecision(2) << fixed;
cout << userName << " your average pay is: " << avgPay;
```

(c) Michele Rousseau Repetition 32



Example: While Loop

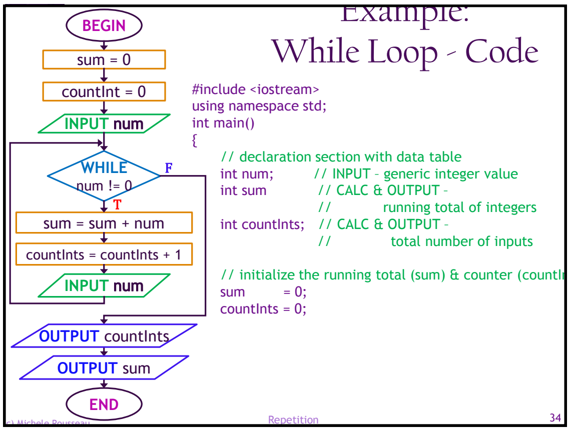
Example

Write a program that will accept a set of integers from a user, sum the set of integers & count how many integers were input. It will stop accepting integers when a 0 is reached.

Output the total number of integers given as input & the sum

First, let's draw the flowchart.

(c) Michele Rousseau Repetition

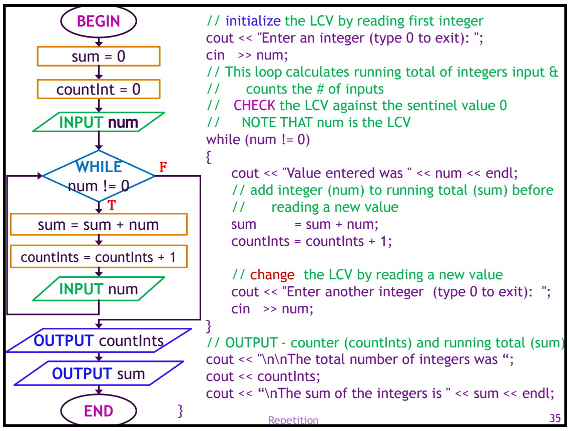


Example: While Loop - Code

```
#include <iostream>
using namespace std;
int main()
{
   // declaration section with data table
   int num;        // INPUT - generic integer value
   int sum          // CALC & OUTPUT -
                    //           running total of integers
   int countInts;   // CALC & OUTPUT -
                    //            total number of inputs

   // initialize the running total (sum) & counter (countI
   sum       = 0;
   countInts = 0;
```

Repetition

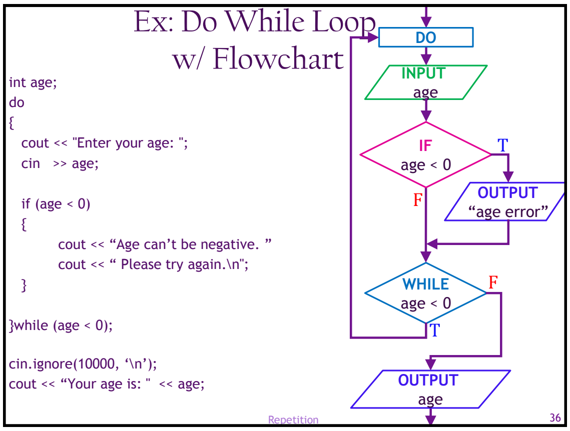
34



```
   // initialize the LCV by reading first integer
   cout << "Enter an integer (type 0 to exit): ";
   cin   >> num;
   // This loop calculates running total of integers input &
   //      counts the # of inputs
   //   CHECK the LCV against the sentinel value 0
   //     NOTE THAT num is the LCV
   while (num != 0)
   {
      cout << "Value entered was " << num << endl;
      // add integer (num) to running total (sum) before
      //      reading a new value
      sum         = sum + num;
      countInts = countInts + 1;

      // change  the LCV by reading a new value
      cout << "Enter another integer  (type 0 to exit):  ";
      cin   >> num;
   }
   // OUTPUT - counter (countInts) and running total (sum)
   cout << "\n\nThe total number of integers was “;
   cout << countInts;
   cout << “\nThe sum of the integers is " << sum << endl;
}
```

Repetition

35



Ex: Do While Loop w/ Flowchart

```
int age;
do
{
   cout << "Enter your age: ";
   cin   >> age;

   if (age < 0)
   {
         cout << “Age can’t be negative. ”
         cout << “ Please try again.\n";
   }

}while (age < 0);

cin.ignore(10000, ‘\n’);
cout << “Your age is: "  << age;
```

Repetition

36