BIRZEIT UNIVERSITY

Algorithm- 2

Computer Science Department

Iterative

- **Control structure that repeats a set of actions (loop body) while some condition remains true or until some condition becomes false**

(1)
```
WHILE condition
   actions
END WHILE
```

(3)
```
FOR iteration bounds
   actions
END FOR
```

(2)
```
REPEAT
   actions
UNTILE  condition
```

Definite and Indefinite Loops

Looping may be achieved using either a definite loop or an indefinite loop:

- **A definite loop is also referred to as a counter-controlled loop**
  - **→ The *loop body* will be executed a specific number of times.**
- **An indefinite loop is also referred to as a sentinel-controlled loop**
  - **→ The number of times the *body of the loop* should be executed can be different for each run of a program**

Understanding the Loop in a Program's Mainline Logic

- **Three steps that should occur in every properly functioning loop**
  - **Initialize the variable that will control the loop (sentinel or counter value)**
  - **Test the condition to determine whether the loop body executes**
  - **Update (aka alter) the loop control variable**
    - **increment/decrement the lcv**
    - **get a new input value to compare to the sentinel value**

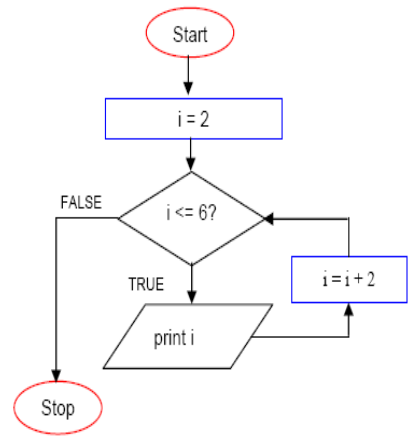
Using a Counter-Controlled while loop

- As long as a boolean expression (the condition) remains true, the while loop's body executes.
- Essential Steps:
  - Create and initialize a loop control variable (lcv)
  - Determine the upper or lower limit for the lcv
  - Determine the step (increment or decrement) for the lcv
  - Determine the boolean expression (condition) that will control the loop
  - Each iteration of the loop, update the lcv

Counter Controlled **While** Loop

```
Set i equal to two
While i less than or equal six
    print i
    add two to i
end while
```

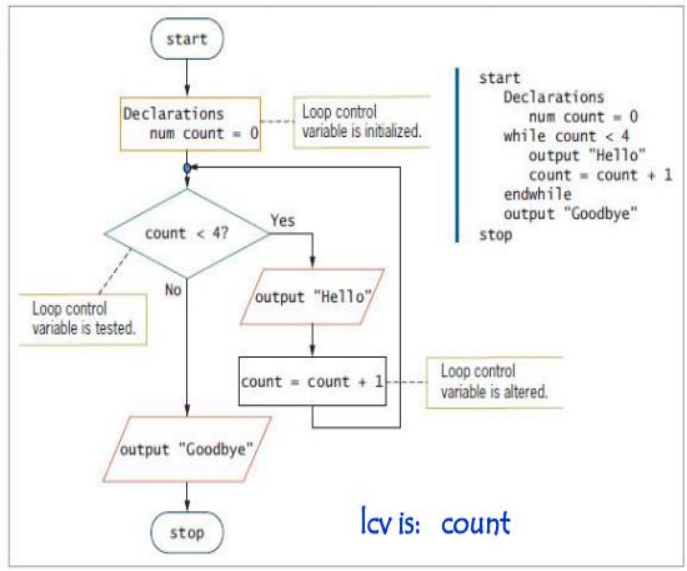
Output:
2 4 6



Using a **Definite (Counter-Controlled) Loop**

Must the condition be true or false for the loop body to be executed?

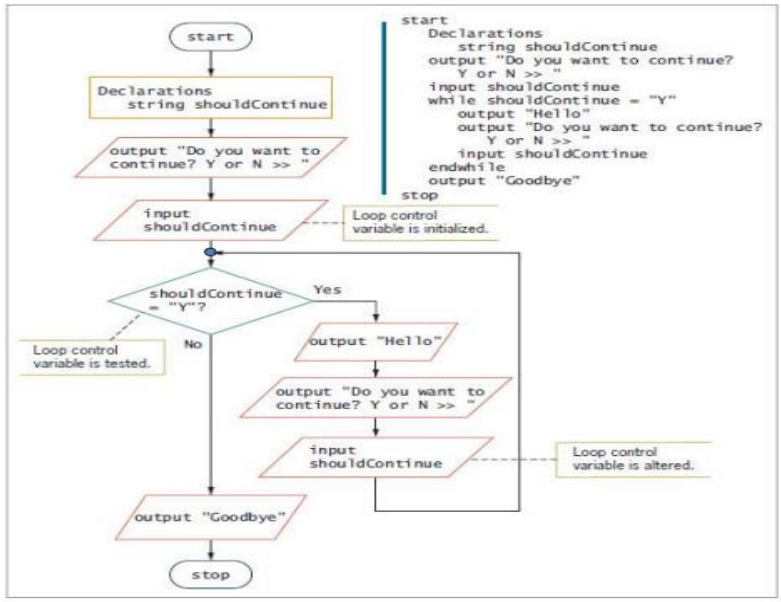
How many times is "Hello" displayed?



Using an **Indefinite Loop (Sentinel-Controlled)** with a **Sentinel Value**

- **Indefinite loop**
  - May be performed a different number of times each time the program executes
- **Essential steps:**
  - Identify a sentinel value (value outside the range of valid input data) that will be used as the loop exit condition
    - name = QUIT //where QUIT is a named constant sentinel value
    - End_of_input //RAPTOR sentinel module expression
    - Java → hasNext( ) == false //Java method to evaluate end of input condition
  - Each iteration of the loop get a new input value to compare to the sentinel value.

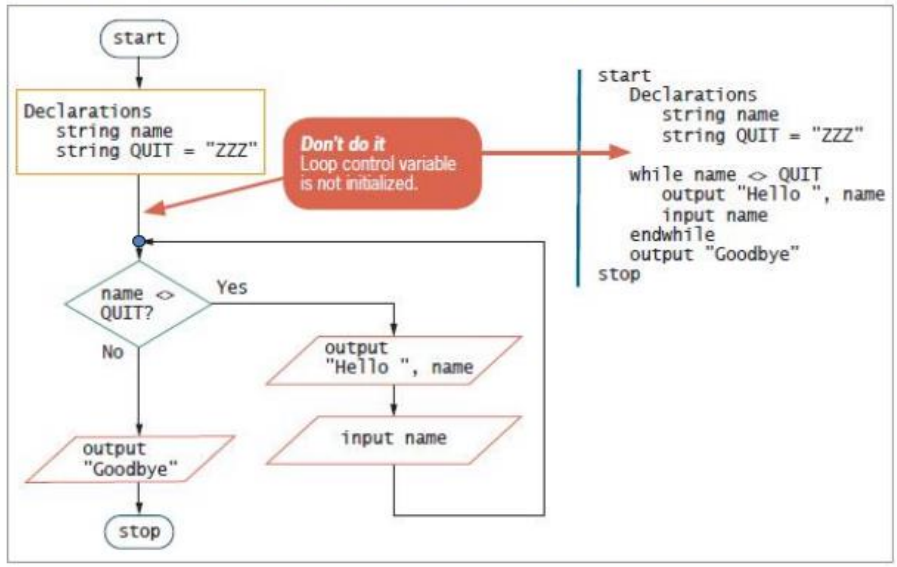
Example: Sentinel controlled indefinite loop



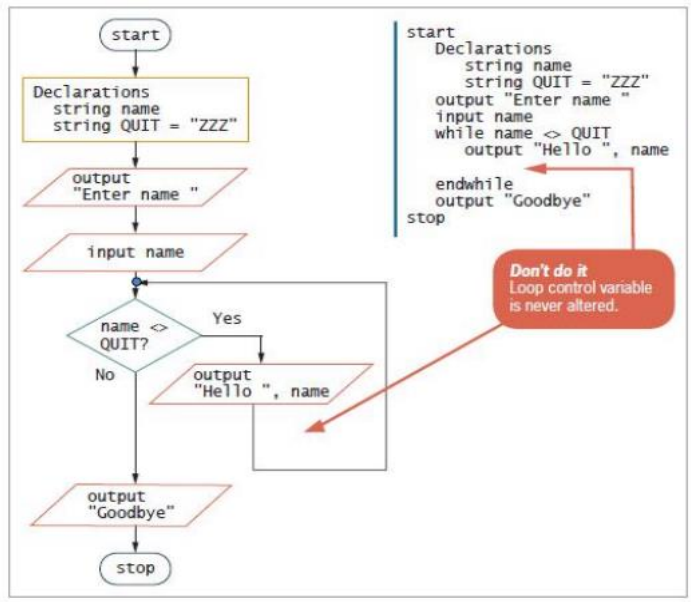
Common Loop Mistakes

- Neglecting to initialize the loop control variable
- Neglecting to update the loop control variable
- Loop executes one too many or one too few times
  - **operator: < or <= OR > or >=**
  - **number of iterations = Last used – First used + 1**
- Including statements inside the loop that belong outside the loop

Incorrect logic: lcv initialization is missing



Incorrect logic: lcv is not altered



Counter Controlled - avg.

- Write an algorithm to calculate the average of a set of 10 students.

Solution 1

1. Set counter to **zero**
2. Set total to zero
3. While counter is *less than* ten
   Ask user to enter grade
   Read grade and save as gd
   Add gd into the total
   increment counter
   end while
4. Set the average to the total divided by counter
5. Print "the average is " average

Solution 2

1. Set counter to **one**
2. Set total to zero
3. While counter is *less than or* ***equal*** ten
   Ask user to enter grade
   Read grade and save as gd
   Add the gd into the total
   increment counter
   end while
4. Set the average to the total divided by **10**
5. Print "the average is " average

Please solve q1 page 12 – sentinel controlled avg

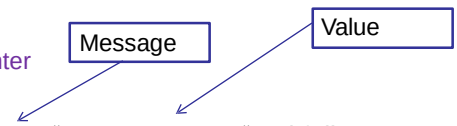
Example 4

Write an algorithm that will count the number of student pass in a class and the amount failed. The pass mark is more than or equal to 65. Suppose the number of students are 52 . The algorithm should **output** the *amount fail* and *passed*.

Example 4 – cont.

1. Set counter to zero
2. Set numberOfStudents to **52**
3. Set passCounter to zero
4. Set failureCounter to zero
5. While counter **less than** numberOfStudents
   Ask user to enter student mark
   Read mark and save as mk
   if mk greater than or equal sixty five then
   increment passCounter
   else
   increment failureCounter

   end if
   increment counter
   end while
6. Print "pass counter =" passCounter "and failure counter =" failureCounter



Please solve q2 page 12 - sentinel

Using a `for` Loop

- **`for` statement** or **`for` loop** is a definite loop
  - specifically, it is a pre-test loop
- Puts all of the loop control expressions in the for loop header:
  1. Initialize
  2. Test
  3. Update
- Takes the form: [initial, final]

```
for loopControlVariable = initialValue to finalValue [step
 stepValue]  do something
endfor
```

Using a for Loop – cont.

- Example pseudocode for loop

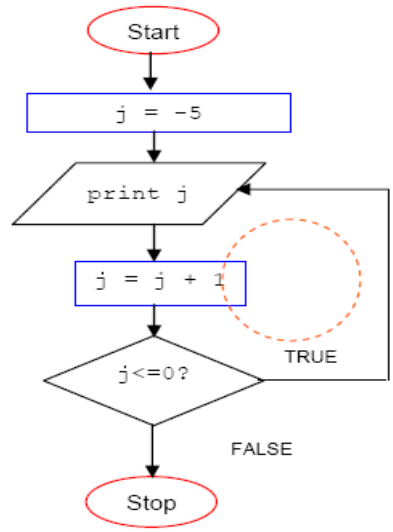
```
for count = 0 to 3 step 1          [0,3]
     output "Hello"
end for
```

- Initializes count to 0
- Checks count against the limit value 3 (test)
- If evaluation is true, for statement body prints the label
- Executes 4 times (last=3, first=0, 3-0+1 equals 4)
- Increases count by 1 (update)
- while loop:
  - count = 0
  - while count <= 3 or while count < 4
- Java for loop: for (count=0; count <= 3; count += 1)

Iterative - **Repeat**

1. Set j equal to negative five
2. Repeat
   print j
   increment j
   until j less than or equal to zero

Output:
-5 -4 3 -2 -1 0



Write an algorithm to print the **sum of the digits** of a given number

Input a Number
Initialize Sum to zero
While Number is not zero
    Get Remainder by Number Mod 10
    Add Remainder to Sum
    Divide Number by 10
End While
Print Sum

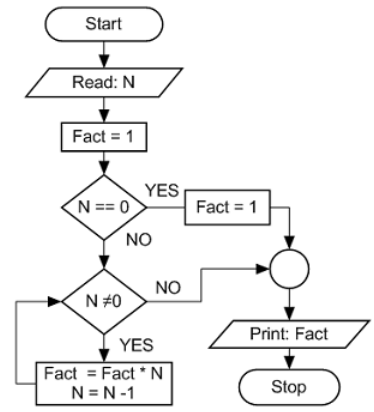
Lab 3-12: Factorial

Factorial function is defined as:

- If N = 0 then N! = 1
- If N > 0 then N! = N(N – 1)

1. Ask user to enter N
2. Read N
3. Set Fact = 1
4. If N equals 0 then
        Set Fact = 1
   Else
        While N not equal 0
            Set Fact = Fact * N
            N = N -1
        End While
   End If
5. Print Fact

0! = 1
1! = 1
2! = 1.2 = 2
3!= 1.2.3 = 6
4! = 1.2.3.4 = 24
5! = 1.2.3.4.5 = 120
6! = 1.2.3.4.5.6 = 720



Please solve q4 page 12 in lab

LAB: Write an algorithm to calculate and print the **nth power of a** number.

If the user enters the number= 8 and n=3, the algorithm should calculate the value of 8^3= 8*8*8 and print the result which is 512

Read base number as **base**
Read exponent as **exponent**
Set result to 1
while exponent not equal 0
    result = result * base
    decrement exponent
End While
Print result

Write an algorithm to print the sum of the following series , taking the first 7 terms. use only one loop. A= 1! +2!+3!+4!+5!+6!+7!

Set oldFact equal to one
Set counter equal to one
Set sum equal to zero
While counter in less than eight
    Set oldFact equal to oldFact multiply by counter
    Set sum equal to sum added by oldFact
    Increment counter
End while
Print sum

LAB: Write an algorithm to check if the number is **prime** or not

Input Any integer number (num)
Output Is it a prime number or Not

1 Set i to 2
2 While i less than or equal num/2
3 if num mod i = 0
4 print "Not a Prime number" and exit;
5 Increment i by 1
6 If (i is equal (num/2)+1)
7 print "Prime number"

Counter-Controlled while loop- Example 1

Write an Algorithm to **reverse** digits of an integer

Input: num
(1) Initialize rev = 0
(2) Loop while num > 0
(a) Multiply rev by 10 and add remainder of num to rev
(rev = rev*10 + num%10)
(b) Divide num by 10
(num = num/10)

(3) Return rev