LAB 7 ASSIGNMENT

Ch. 5 Repetition Structures

Start: 03/27/2024 4:34 PM
Name: Imani Hollie

LAB 7.1 – ALGORITHMS

Algorithm 1: Design a `While` loop that lets the user enter a number. The number should be multiplied by 10, and the result stored in a variable named `product`. The loop should iterate as long as `product` contains a value less than 100.

```
//Declare counter variable and initialize it to 0
Declare Integer product = 0
While product < 100
   Display "Enter a number."
   Input number
   product = number * 10
   Display number, "times 10 equals ", product
End While
```

Algorithm 2: Design a `Do-While` loop that asks the user to enter two numbers. The numbers should be added and the sum displayed. The loop should be added and the sum displayed. The loop should ask the user whether they wish to perform the operation again. If so, the loop should repeat; Otherwise it should terminate.

```
Declare Integer num1, num2
Declare String check = "y"
Do
   Display "Enter Number:"
   Input num1
   Display "Enter Number:"
   Input num2
   Set total = num1 + num2
   Display "The total," total, "equals", num1, "plus", num2
   Display "Do you want to add again?"
   Input check
While check == "y"
End While
```

Algorithm 3: Design a `For` loop that displays the following set of numbers: 0, 10, 20, 30, 40, 50, … 1000

```
//Declare counter variable and initialize it to 0
Declare Integer count = 0
Constant Integer MAX_VALUE = 1000
//For Loop to display 10s up to 1000
For count = 0 To MAX_VALUE Step 10
   Display count
End For
```

Algorithm 4: Design a loop that asks the user to enter a number. The loop should iterate 10 times and keep a running total of the numbers entered.

```
//Declare num to hold user input and count as the counter variable
```

```
Declare Integer num, count
//Declare total as accumulator variable
Declare Integer total = 0
//For loop prompts user for a number 10 times and adds them
For count = 1 To 10
   Display "Enter a number."
   Input num
   Set total = total + num
   Display "The total is ", total
End For
```

Algorithm 5: Design a `For` loop that calculates the total of the following series of numbers: $\frac{1}{30} + \frac{2}{29} + \frac{3}{28} + \cdots + \frac{30}{1}$

```
//Same start as A-4, just different named variables
Declare quotient, count
Declare Integer total = 0
//For this part I had to use a range Function with the For Loop
For count in range(1, 31)
   //Setting count to top (numerator)
   Set top = count
   //Setting bottom (denominator) to 31 - count
   Set bottom = 31 - count
   quotient = top / bottom
   total = total + quotient
   Display "The total is ", total
End For
```

Algorithm 6: Design a nested loop that displays 10 rows of '#' characters. There should be 15 '#' characters in each row.

```
//Utilize the range embedded in the nested loop
Declare String line, characters
//The outermost loop (line) will iterate 10 times
For line in range(1, 10)
   //The innermost loop (characters) will iterate 15 times
   For characters in range(1, 15)
        Display "#"
   End For
   //This is how to move to the next line without having to print it
   Display newline
End For
```

Algorithm 7: Convert the `While` loop in the following code to a `Do-While` loop:

```
Declare Integer x = 1
While x > 0
   Display "Enter a number."
   Input x
End While
```

```
Declare Integer x = 1
//The while loop iterates once prior to checking the condition
Do
```

```
   Display "Enter a number."
   Input x
While x > 0
End While
```

Algorithm 8: Convert the following `Do-While` loop to a `For` loop:

- ```
  Declare String sure
  Do
     Display "Are you sure you want to quit?"
     Input sure
  While sure != "Y" AND sure != "y"
  //The Do-While loop begins with Do and ends with While (condition)
  //The For loop begins with For (condition) and ends with End For
  ```
- ```
  Declare String sure
  For sure != "Y" AND sure != "y"
     Display "Are you sure you want to quit?"
     Input sure
  End For
  ```

Algorithm 9: Convert the following `While` loop to a `For` loop:

- ```
  Declare Integer count = 0
  While count < 50
     Display "The count is ", count
     Set count = count + 1
  End While
  ```
- ```
  Declare Integer count = 0
  //Utilize a range function instead of a counter
  For count in range(0, 49)
     Display "The count is ", count
  End For
  ```

Algorithm 10: Convert the following `For` loop to a `While` loop:

- ```
  Declare Integer count
  For count = 1 To 50
     Display count
  End For
  ```
- ```
  Declare Integer count
  While count <= 50
     Display count
     Set count = count + 1
  End For
  ```

LAB 7.2 – DEBUGGING EXERCISES

Exercise 1: Find the error in the following pseudocode:

```
1. Declare Boolean finished = False
2. Declare Integer value, cube
3. While NOT finished
   4. Display "Enter a value to be cubed."
   5. Input value;
   6. Set cube = value ^ 3
   7. Display value, " cubed is ", cube
8. End While
```

An error occurs between Lines 7 and 8, because there isn't an update for the finished variable, making this an infinite loop. To fix it, add another line between them and set finished to equal True (finished = True) to close the loop as follows:

```
1. Declare Boolean finished = False
2. Declare Integer value, cube
3. While NOT finished
   4. Display "Enter a value to be cubed."
   5. Input value;
   6. Set cube = value ^ 3
   7. Display value, " cubed is ", cube
   8. Set finished = True
9. End While
```

Exercise 2: The programmer intended the following pseudocode to display the numbers 1 – 60, and then display the message "Time's up!" It will not function as intended, however. Find the error.

```
1. Declare Integer counter = 1
2. Const Integer TIME_LIMIT = 60
3. While counter < TIME_LIMIT
   4. Display counter
   5. Set counter = counter + 1
6. End While
7. Display "Time's up!"
```

An error occurs on Lines 2 and 3: For Line 2, the 'Const' keyword used should be replaced with 'Constant'; And for Line 3, the condition less than (<) should be changed to less than or equal to (<=) to ensure the loop iterates, as follows:

```
1. Declare Integer counter = 1
2. Constant Integer TIME_LIMIT = 60
3. While counter <= TIME_LIMIT
   4. Display counter
   5. Set counter = counter + 1
6. End While
7. Display "Time's up!"
```

Exercise 3: The programmer intended the following pseudocode to get five sets of two numbers each, calculate the sum of each set, and calculate the sum of all the numbers entered. It will not function as intended. however. Find the error.

```
1. //This program calculates the sum of five sets of two numbers.
2. Declare Integer number, sum, total
3. Declare Integer sets, numbers
4. Constant Integer MAX_SETS = 5
5. Constant Integer MAX_NUMBERS = 2
6. Set sum = 0;
7. Set total = 0;
8. For sets = 1 To MAX_NUMBERS
   9. For numbers = 1 To MAX_SETS
      10. Display "Enter number ", numbers, " of set ", sets, "."
      11. Input number;
      12. Set sum = sum + number
   13. End For
   14. Display "The sum of set ", sets, " is ", sum, "."
   15. Set total = total + sum
   16. Set sum = 0
17. End For
18. Display "The total of all the sets is ", total, "."
```

An error occurs on Lines 8 and 9: For Line 8, the outer loop should be changed from MAX_NUMBERS to MAX_SETS since it iterates over the groups of data (meaning it logically works with data groups rather than individual pieces of data); For Line 9, the inner loop should be changed from MAX_SETS to MAX_NUMBERS since it iterates over the individual pieces of data (meaning it logically does the opposite of outer loops, as it works with the data within the groups that are managed by them). To fix it, switch the variables between Lines 8 and 9 as follows:

```
1. //This program calculates the sum of five sets of two numbers.
2. Declare Integer number, sum, total
3. Declare Integer sets, numbers
4. Constant Integer MAX_SETS = 5
5. Constant Integer MAX_NUMBERS = 2
6. Set sum = 0;
7. Set total = 0;
8. For sets = 1 To MAX_SETS
   9. For numbers = 1 To MAX_NUMBERS
      10. Display "Enter number ", numbers, " of set ", sets, "."
      11. Input number;
      12. Set sum = sum + number
   13. End For
   14. Display "The sum of set ", sets, " is ", sum, "."
   15. Set total = total + sum
   16. Set sum = 0
17. End For
18. Display "The total of all the sets is ", total, "."
```