LAB 5 ASSIGNMENT

Ch. 4 Decision Structures and Boolean Logic

Start: 01/31/2024 6:11 PM
Name: Imani Hollie

LAB 5.1 – ALGORITHMS

Algorithm 1: Design an `If-Then` statement (or a flowchart with a single alternative decision structure) that assigns 20 to the variable `y` and assigns 40 to the variable `z` if the variable `x` is greater than 100.

```
If x > 100 Then
   Set y == 20
   Set z == 40
End If
```

Algorithm 2: Design an `If-Then` statement (or a flowchart with a single alternative decision structure) that assigns 0 to the variable `b` and assigns 1 to the variable `c` if the variable `a` is less than 10.

```
If a < 10 Then
   Set b == 0
   Set c == 1
End If
```

Algorithm 3: Design an `If-Then-Else` statement (or a flowchart with a dual alternative decision structure) that assigns 0 to the variable `b` if the variable `a` is less than 10. Otherwise, it should assign 99 to the variable `b`.

```
If a < 10 Then
   Set b == 0
Else
   Set b == 99
End If
```

Algorithm 4: The following pseudocode contains several nested `If-Then-Else` statements. Unfortunately, it was written without proper alignment and indentation. Rewrite the code and use the proper conventions of alignment and indentation.

```
If score < 60 Then
Display “Your grade is F.”
Else
If score < 70 Then
Display "Your grade is D.”
Else
If score < 80 Then
Display "Your grade is C.”
Else
If score < 90 Then
Display “Your grade is B.”
Else
Display ‘Your grade is A.”
End If
End If
End If
```

```
   End If
```

- ```
  If score < 60 Then
     Display "Your grade is F."
  Else
     If score < 70 Then
        Display "Your grade is D."
     Else
        If score < 80 Then
           Display "Your grade is C."
        Else
           If score < 90 Then
              Display "Your grade is B."
           Else
              Display "Your grade is A."
           End If
        End If
     End If
  End If
  ```

Algorithm 5: Design nested decision structures that performs the following: If `amount1` is greater than 10 and `amount2` is less than 100, display the greater of `amount1` and `amount2`.

- ```
  Display "Enter an amount."
  Input amount1
  Display "Enter an amount."
  Input amount2
  If amount1 < amount2 Then
     Display amount1
  Else
     Display amount2
  End If
  ```

Algorithm 6: Rewrite the following `If-Then-Else` If statement as a `Select Case` statement.

- ```
  If selection == 1 Then
       Display "You selected A."
  Else If selection == 2 Then
       Display "You selected 2."
  Else If selection == 3 Then
       Display "You selected 3."
  Else If selection == 4 Then
       Display "You selected 4."
  Else
       Display "Not good with numbers, eh?"
  End If
  ```
- ```
  If selection == 1 Then
     Display "You selected A."
  Else
     If selection == 2 Then
        Display "You selected 2."
     Else
        If score == 3 Then
  ```

```
         Display "You selected 3."
      Else
         If score == 4 Then
            Display "You selected 4."
         Else
            Display "Not good with numbers, eh?"
         End If
      End If
   End If
End If
```

Algorithm 7: Design an `If-Then-Else` statement (or a flowchart with a dual alternative decision structure) that displays "Speed is normal" if the `speed` variable is within the range of 24 to 56. If speed holds a value outside this range, display "Speed is abnormal."

```
If speed < 24 AND speed > 54 Then
   Display "Speed is normal."
Else
   Display "Speed is abnormal."
End If
```

Algorithm 8: Design an `If-Then-Else` statement (or a flowchart with a dual alternative decision structure) that determines whether the `points` variable is outside the range of 9 to 51. If the variable holds a value outside this range it should display "Invalid points." Otherwise, it should display "Valid points."

```
If NOT(points < 9 AND points > 54) Then
   Display "Invalid points."
Else
   Display "Valid points."
End If
```

Algorithm 9: Design a case structure that tests the `month` variable and does the following:

- If the `month` variable is set to 1; It displays "January has 31 days."
- If the `month` variable is set to 2; It displays "February has 28 days."
- If the `month` variable is set to 3; It displays "March has 31 days."
- If the `month` variable is set to anything else, it displays "Invalid selection."

```
Select month
   Case 1:
      Display "January has 31 days."
   Case 2:
      Display "February has 28 days."
   Case 3:
      Display "March has 31 days."
   Default:
      Display "Invalid selection."
End If
```

Algorithm 10: Write an `If-Then` statement that sets the variable `hours` to 10 when the flag variable `minimum` is set.

```
If minimum Then
   Set hours == 10
```

```
   End If
```

LAB 5.2 – DEBUGGING EXERCISES

Exercise 1: Part of the following pseudocode is incompatible with Java, Python, C, and C++ languages. Identify the problem. How would you fix the problem if you were to translate this pseudocode into one of the languages?

```
1. Module checkEquality(Integer num1, Integer num2)
   2. If num1 = num2 Then
      3. Display "The values are equal."
   4. Else
      5. Display "The values are NOT equal."
   6. End If
7. End Module
```

An error occurs on Line 2, because the equal operator is two equal signs (==). To fix it, on Line 2 add another equal sign as follows:

```
1. Module checkEquality(Integer num1, Integer num2)
   2. If num1 == num2 Then
      3. Display "The values are equal."
   4. Else
      5. Display "The values are NOT equal."
   6. End If
7. End Module
```

Exercise 2: The intended purpose of the following module is to set the temp parameter to the value 32.0 if it is not already equal to 32.0. This will not work as the programmer expects, however. Find the problem.

```
1. Module resetTemperature(Real Ref temp)
   2. If NOT temp == 32.0 Then
      3. Set temp == 32.0
   4. End If
5. End Module
```

An error occurs on Lines 2 and 3, because (1) on Line 2 the expression `temp == 32.0` should be in parentheses [`()`] and (2) on Line 3 only needs an equal sign (=) to assign a value to a variable. To fix it, on Line 2 add parentheses and on Line 3 delete the first equal sign with an exclamation point as follows:

```
1. Module resetTemperature(Real Ref temp)
   2. If NOT(temp == 32.0) Then
      3. Set temp = 32.0
   4. End If
5. End Module
```

Exercise 3: The intended purpose of the following module is to determine whether the `value` parameter is within a specified range. The module will not work, however. Find the problem.

```
1. Module checkRange(Integer value, Integer lower, Integer upper)
   2. If value < lower AND value > upper Then
      3. Display "The value is outside the range."
   4. Else
      5. Display "The value is within the range."
   6. End If
7. End Module
```

An error occurs on Lines 3 and 5 as when the statements executes, if the expression is found to be true then the value parameter will be within the range, otherwise it will be outside of the range. To fix it, switch the strings on Lines 3 and 5 as follows:

```
1. Module checkRange(Integer value, Integer lower, Integer upper)
   2. If value < lower AND value > upper Then
      3. Display "The value is within the range."
   4. Else
      5. Display "The value is outside the range."
   6. End If
7. End Module
```