**Lab 3: Decisions and Boolean Logic**

This lab accompanies Chapter 3 of *Starting Out with Programming Logic & Design*.

Name: Ulric Aird

# Lab 3.1 – Evaluating Conditions

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Critical Review    A relational operator determines whether a specific relationship exists between two values.    Relational operators | | | | |
|  | **Operator** | **Meaning** | **Boolean Expression** |  |
| > | Greater than | X > Y |
| < | Less than | X < Y |
| >= | Greater than or equal to | X >= Y |
| <= | Less than or equal to | X <= Y |
| = = | Equal to | X = = Y |
| != | Not equal to | X != Y |
|  | | | | |
|  | | | | |

This lab requires you to think about possible true and false conditions using if statements.

**Step 1:** Consider the following values set to variables.

* myAge = 32
* yourAge = 18
* myNumber = 81
* yourNumber = 17
* votingAge = 18
* myName = "Katie"
* yourName = "Bob"

**Step 2:** Based on the values to the variables in Step 1, do the following conditions result in a true or false statement? (Reference: Boolean Expressions, page 111).

|  |  |
| --- | --- |
| **The condition** | **True or False** |
| myAge >= yourAge | True |
| yourAge > myAge | False |
| myAge == 45 | False |
| yourAge == votingAge | True |
| votingAge <= yourAge | True |
| myAge <= votingAge | True |
| myName != yourName | True |
| myNumber <= myAge | False |
| yourNumber >= myAge | False |
| yourNumber != 17 | False |

**Step 3:** Based on the values to the variables in Step 1, what is the expected output? Hint: The output will be either what is printed to the screen, or nothing. (Reference: Boolean Expressions, page 119).

|  |  |
| --- | --- |
| **The condition** | **Expected Output** |
| If myName == yourName Then print "We have the same name" End If | Nothing |
| If myAge >= yourAge Then  print "I am older or equal to your age" End If | I am older or equal to your age |
| If myName != "Katie" Then print "That is not my name" End If | Nothing |
| If myName == "Katie" Then print "That is my name" End If | That is my name |
| If myNumber == 17 Then print "My number is 17" End If | Nothing |
| If myNumber >=80 Then  print "My number is 80 or more" End If | My number is 80 or more |
| If yourNumber <= yourAge Then print "Your number is less than or  equal to your age" End If | Your number is less than or equal to your age |
| If myNumber < yourNumber Then print "My number is less" End If | Nothing |
| If yourAge >= votingAge Then print "You can vote" End If | You can vote |
| If myAge < yourAge Then print "I am younger" End If | Nothing |

# Lab 3.2 – Pseudocode and Decisions

|  |
| --- |
| Critical Review    Questions are often asked using an if statement such as if X > Y, whereas the question asked is "is X greater than Y"?    The general structure of an if statement is  If condition Then  Statement  Statement Etc.  End If |

This lab requires you to think about the steps that take place in a program by writing pseudocode. Read the following program prior to completing the lab.

A retail company assigns a $5000 store bonus if monthly sales are

$100,000 or more. Additionally, if their sales exceed 125% or more of their monthly goal of $90,000, then all employees will receive a message stating that they will get a day off.

**Step 1**: This program is easiest when solved using just one variable. Declare the variables that you will need in the program, using the proper data type and documenting the purpose. Depending on your programming style, you may find additional variables are useful. If that is the case, adjust your program as necessary.

|  |  |
| --- | --- |
| **Variable Name** | **Purpose** |
| monthlySales | Stores the monthly sales |
|  |  |

**Step 2:** Given the major task involved in this program, what modules might you consider including? Also describe the purpose of the module.

|  |  |
| --- | --- |
| **Module Name** | **Purpose** |
| Module getSales () | Allows the user to enter the monthly sales. |
| Module awardBonus () | This module will determine if a bonus should be awarded. |
| Module dayOff () | This module will determine if a day off should be awarded. |

**Step 3:** Complete the pseudocode by writing the missing lines. When writing your modules and making calls, be sure to pass necessary variables as arguments and accept them as reference parameters if they need to be modified in the module. (Reference: Writing a Decision Structure in Pseudocode, page 118).

Module main ()

//Declare local variables

Declare Real monthlySales

//Function calls

Call getSales(monthlySales)

Call awardBonus(monthlySales)

Call dayOff(monthlySales)

End Module

//this module takes in the required user input

Module getSales(Real Ref monthlySales)

Display "Enter the total sales for the month."

Input monthlySales

End Module

//this module will determine if a bonus is awarded

Module awardBonus(Real Ref monthlySales)

If monthlySales >=100000 Then

Print "You get a bonus of $5,000!!!"

End If

End Module

//this module will determine if all employees get a day

//off. If sales are greater than or equal to 112500, then

//they get a day off.

Module dayOff(Real Ref monthlySales)

If monthlySales >= 112500 Then

Print “You Got a Day Off”

End If

End Module

# Lab 3.3 – Flowcharts

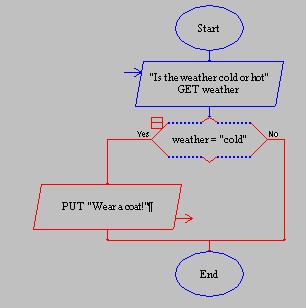
Critical Review

The flowchart symbol used to indicate some condition is a diamond. An

if

statement is called a single alternative decision structure. The code will only process

if the decision is true.

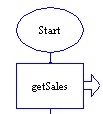


This lab requires you to convert your pseudocode in Lab 3.2 to a flowchart. Use an application such as Raptor or Visio.

**Step 1:** Start Raptor and save your document as *Lab 3-3*. The *.rap* file extension will be added automatically. Start by adding a Comment box that declares your variables. Here is how your Comment box should look.



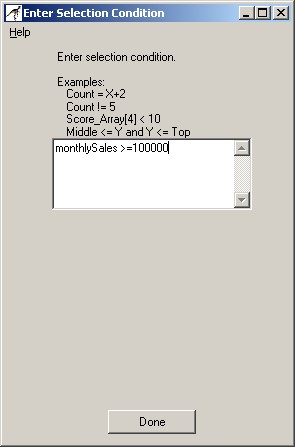
**Step 2:** The next step in your flowchart should be to call your methods. Below is a start of how your main should look.



**Step 3:** Continue this process to add your additional methods you defined in Lab 3.2, Step 3.

**Step 4:** Click on the getSales tab and add the necessary code to enter the monthly sales. Your getSales method might look like the following:

**Step 5:** Click on the second module which determines if a bonus of $5000 is awarded. Click the Selection symbol and add it between the start and the end of the module. Double click on the diamond symbol and add the code to determine if monthlySales is greater than or equal to 100000. The enter selection condition should be written as follows:



**Step 6:** Drag an output symbol and drop it on the True line. Double click on the output box and add text that prints "You earned a $5000 bonus!". Your module should like as follows:

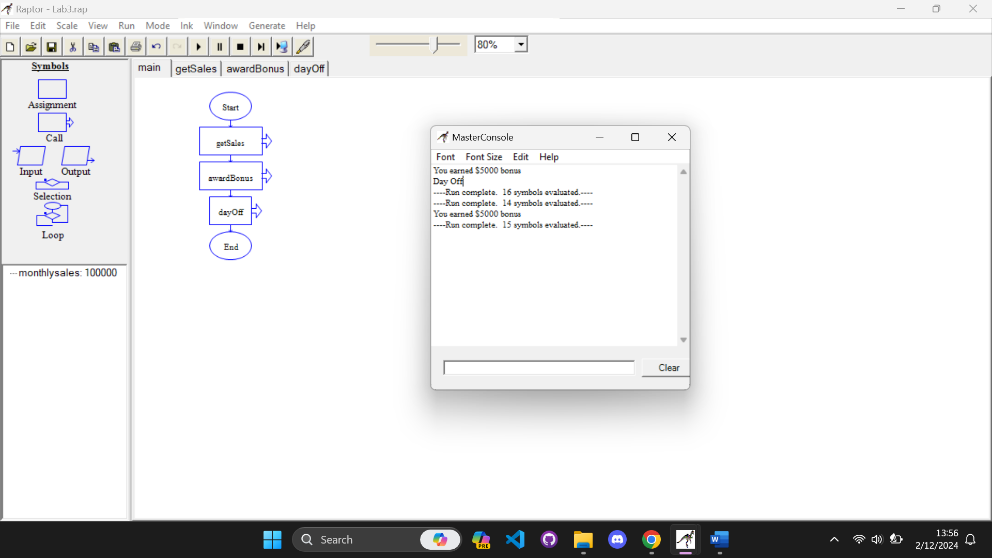
**Step 7:** Repeat the process in Step 6 to code your next module.

**Step 8:** When your program is complete, test the following monthly sales and ensure that the output matches the following. If your output is different, then review your decision statements.

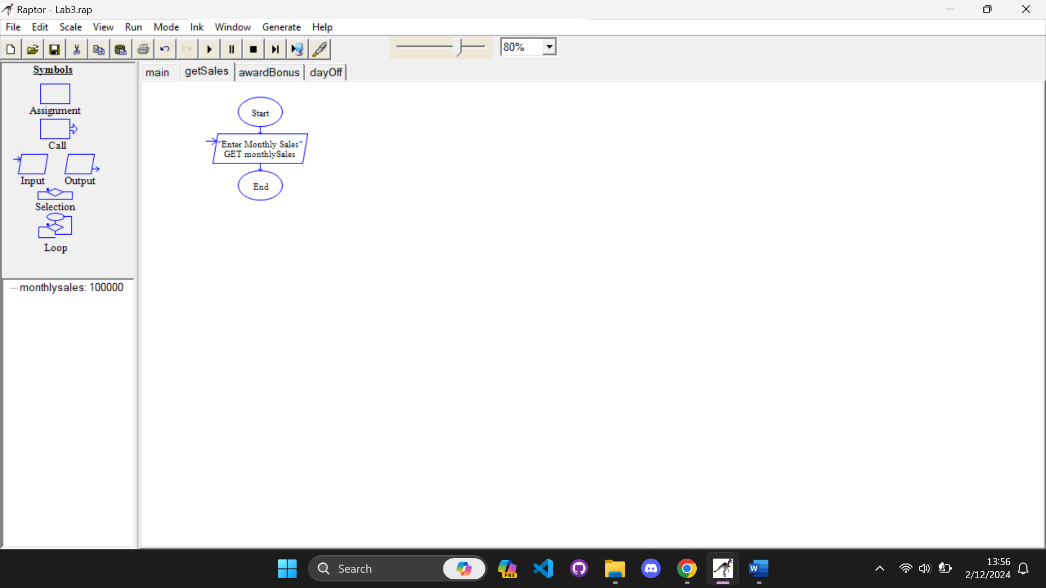
|  |  |
| --- | --- |
| **Monthly Sales** | **Expected Output** |
| monthlySales = 102500 | You earned a $5000 bonus! |
| monthlySales = 90000 | <nothing> |
| monthlySales= 112500 | You earned a $5000 bonus!  All employees get one day off!!! |

**Step 9:** The final step is to insert your finished flowchart in the space below. Inside Raptor, select File and the Print to Clipboard from the menu. Inside Word in the space below, select Edit and Paste. You will have to do this for each module you created.

**PASTE FLOWCHART HERE**



A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated