# **Lab 3: Decisions and Boolean Logic**

This lab accompanies Chapter 4 of *Starting Out with Programming Logic & Design*.

Name: \_\_\_\_\_\_\_\_David White\_\_\_\_\_\_\_\_\_

**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 160).

|  |  |
| --- | --- |
| **The condition** | **True or False** |
| myAge >= yourAge | True |
| yourAge > myAge | false |
| myAge == 45 | false |
| yourAge == votingAge | true |
| votingAge <= yourAge | true |
| myAge <= votingAge | false |
| 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 160).

|  |  |
| --- | --- |
| **The condition** | **Expected Output** |
| If myName == yourName Then  print "We have the same name"  End If |  |
| 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 |  |
| 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 |  |
| If myNumber >=80 Then  print "My number is 80 or more"  End If | My number is 80 or moure |
| 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 |  |
| If yourAge >= votingAge Then  print "You can vote"  End If | You can vote |
| If myAge < yourAge Then  print "I am younger"  End If |  |

**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 checkBonus() | This module will determine if a bonus should be awarded. |
| Module checkDay() | 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 160).

Module main ()

//Declare local variables

Declare Real monthlySales

//Function calls

Call getSales(monthlySales)

Call checkBonus(monthlySales)

Call checkDay(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 checkBonus(Real Ref monthlySales)

If monthlySales >=100000 Then

Display "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 checkDay (Real Ref monthlySales)

If monthlySales >=112500

Display “You get a day off!!!”

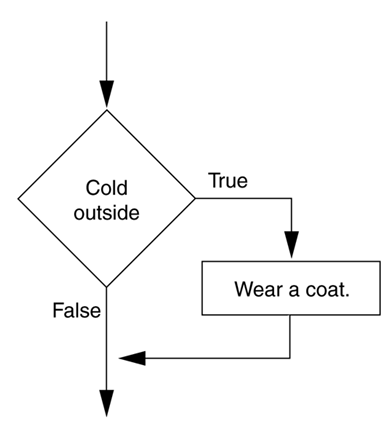
End If

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. Please use Visio or Draw.io to create your diagram.

When your program design is complete, test the following monthly sales and ensure that the output matches the following. If your output is different, then review your decision structure.

|  |  |
| --- | --- |
| **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!!! |

**Insert** your finished flowchart in the space below for later reference. You will also upload this flowchart in Blackboard. If you are using Draw.io, please convert your flowchart to a pdf file before uploading it.

**PASTE FLOWCHART HERE**



**Lab 3.4 – Python Code**

Critical Review

In Python we use the if statement to write a single alternative decision structure. Here is the general format of the if statement:

*if condition:*

*statement*

*statement*

*etc.*

For simplicity, we will refer to the first line as the if clause. The if clause begins with the word if, followed by a condition, which is an expression that will be evaluated as either true or false. A colon appears after the condition. Beginning at the next line is a block of statements.

When the if statement executes, the condition is tested. If the condition is true, the statements that appear in the block following the if clause are executed. If the condition is false, the statements in the block are skipped.

When strings are evaluated in a condition, single quotation marks are used. For example:

*name1 = 'Mary'*

*name2 = 'Mark'*

*if name1 != name2:*

*print 'The names are not the same.'*

*if name1 == ‘Mary’*

*print ‘The name is Mary.’*

**Step 1:** Start the IDLE Environment for Python. Prior to entering code, save your file by clicking on File and then Save. Select your location and save this file as *Lab3-4.py*. Be sure to include the .py extension.

**Step 2:** Document the first few lines of your program to include your name, the date, and a brief description of what the program does. Description of the program should be:

# This program will demonstrate how to use decision

# statements in Python.

**Step** **3:** Start your program with the following code:

# This program determines if a bonus should be awarded

# The main function

def main():

print('Welcome to the program')

monthlySales = getSales() # gets sales

# This function gets the monthly sales

def getSales():

monthlySales = float(input('Enter the monthly sales $'))

return monthlySales

# calls main

main()

**Step 4:** Add a function call to the method that determines whether a bonus is awarded. The call should be in main and process after monthlySales = getSales(). Be sure to pass monthlySales to the function as an argument since that will be needed to determine if a bonus is awarded. Your code might look as follows:

# Function call to determine bonus

isBonus(monthlySales)

**Step 5:** Under the getSales() function, code the function that will determine whether a bonus is awarded. Be sure to accept monthlySales in the parameter list. Also, note that the if statement is followed by a colon, and the print statement inside must be tabbed over.

def isBonus(monthlySales):

if monthlySales >= 100000:

print "You have earned a $5,000 bonus!!!"

**Step 6:** Repeat the process in Step 4 to make a function call to the method that determines whether all employees get a day off.

**Step 7:** Repeat the process in Step 5 to code the function that will determine whether all employees should get a day off.

**Step 8:** Click Run and Run Module to see how your program processes. Test the following monthlySales values to verify the expected output.

|  |  |
| --- | --- |
| **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:** Execute your program so that it works and paste the final code below

**PASTE CODE HERE**

#######################################################

# Name: David White

# Class: CIS-1400

# Assignment: Lab 3-4

# File: lab3-4.py

# Purpose: Demonstrate how to use decision statements

#######################################################

print('\n\*\*\*David White\*\*\*\n') # Display author's name

#main fuction

def main():

print('This program determines if a bonus should be awarded')

print()

monthlySales = getSales()

checkBonus(monthlySales)

checkDay(monthlySales)

return

#get the montly sales

def getSales():

monthlySales = float(input('Enter the monthly sales: $'))

return monthlySales

#check for bonus

def checkBonus(monthlySales):

if monthlySales >= 100000:

print('You have earned a $5,000 bonus!!!')

return

#check for day off

def checkDay(monthlySales):

if monthlySales >= 112500:

print('You have earned a day off!!!')

return

#call main

main()

**Lab 3.5 – Programming Challenge 1 – Guess the Secrets**

Write the Pseudocode, Flowchart, and Python code for the following programming problem.

**Guess the Secrets**

Write a program that will ask the user to enter a person’s age, their weight, and their birth month. Your program will compare the entered values to the following and print the appropriate responses. Be sure to use modules.

|  |  |  |
| --- | --- | --- |
| **The Secret Answers** | **The Comparison** | **The Printed Response** |
| age = 25 | If the guessed age is less than or equal to 25 | Congratulations, the age is 25 or less |
| weight = 128 | If the guessed weight is greater than or equal to 128 | Congratulations, the weight is 128 or more |
| birthMonth = ‘April’ | If the guessed birth month is equal to April | Congratulations, the birth month is April |

*Hint*: This program can be written a few different ways. It is suggested that only three variables are used to store the guessed answers to age, weight, and birthMonth. The pseudocode for main might look as follows:

GO BACK TO WEEK 3 HOMEWORK TO REVIEW MODULES! You are building on what you learned so far, and putting it all together.

Module main ()

//Declare local variables

Declare Integer age

Declare Integer weight

Declare String birthMonth

//Function calls

Call getAge(age)

Call getWeight(weight)

Call getMonth(birthMonth)

Call correctAnswers(age, weight, birthMonth)

End Module

If the program is written properly, sample output should look as follows:

Sample 1:

**//Input**

Enter your guess for age: 20

Enter your guess for weight: 140

Enter your guess for birth month: March

**//Output**

Congratulations, the age is 25 or less.

Congratulations, the weight is 128 or more.

Sample 2:

**//Input**

Enter your guess for age: 32

Enter your guess for weight: 119

Enter your guess for birth month: April

**//Output**

Congratulations, the birth month is April.

Sample 3:

**//Input**

Enter your guess for age: 58

Enter your guess for weight: 128

Enter your guess for birth month: January

**//Output**

Congratulations, the weight is 128 or more.

**The Pseudocode**

Module main()

//declare variables

Declare Integer age

Declare Integer weight

Declare String birthmonth

//function call

Call getAge(age)

Call getWeight(weight)

Call getMonth(birthMonth)

Call checkAnswers(age, weight, birthMonth)

End Module

//ask for age

Module getAge(Integer Ref age)

Display “Guess the age: ”

Input age

End Module

//ask for weight

Module getWeight(Integer Ref weight)

Display “Guess the weight: “

Input weight

End Module

//ask for month

Module getMonth(String Ref birthMonth)

Display “Guess the Birth Month: ”

Input birthMonth

End Module

//check answers

Module checkAnswers(Integer Ref age, Integer Real weight, String Real birthMonth)

If age <= 25

Display “Congratulations, the age is 25 or less”

End If

If weight >= 128

Display “Congratulations, the weight is 128 or more”

End If

If birthMonth = ‘April’

Display “Congratulations, the birth month is April”

End If

End Module

**The Flowchart**

**­­** 

**The Python Code**

#######################################################

# Name: David White

# Class: CIS-1400

# Assignment: Lab 3-5

# File: lab3-5.py

# Purpose: Guess the values

#######################################################

print('\n\*\*\*David White\*\*\*\n') # Display author's name

#main funcion

def main():

print('Program Start')

print()

age = getAge()

weight = getWeight()

birthMonth = getMonth()

checkAnswers(age, weight, birthMonth)

return

#ask for age

def getAge():

age = int(input('Guess the age: '))

return age

#ask for weight

def getWeight():

weight = int(input('Guess the Weight: '))

return weight

#ask for birth month

def getMonth():

birthMonth = str(input('Guess the Birth Month: '))

return birthMonth

#check answers

def checkAnswers(age, weight, birthMonth):

if age <= 25:

print('Congratulations, the age is 25 or less')

return

if weight >= 128:

print('Congratulations, the weight is 128 or more')

return

if birthMonth == 'april':

print('Congratulations, the birth month is April')

return

return

#call main

main()