

# SODV1101 – Programming Fundamentals

## ILA 4.2

### Lab 4.2 –Logical Operators and Dual Alternative Decisions

#### Critical Review

The logical AND operator and the logical OR operator allow you to connect multiple Boolean expressions to create a compound expression.

The logical NOT operator reverses the truth of a Boolean expression.

When using the AND operator, both conditions must be true in order for the statements within an if to process.

When using the OR operator, either condition must be true in order for the statements within an if to process.

A dual alternative decision structure will execute one group of statements if its Boolean expression is true, or another group if its Boolean expression is false.

The general structure of an if then else statement is

```
If condition Then
    Statement
    Statement
    Etc.
Else
    Statement
    Statement
    Etc.
End If
```

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

**Step 2:** 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: Logical Operators, page 147).

## SODV1101 – Programming Fundamentals

### ILA 4.2

The condition	Expected Output
<pre>If myAge == 31 AND yourAge &lt; myAge Then     Display "My age is 31 and your age is less than that" End If</pre>	
<pre>If myAge &lt;= 35 AND myAge &gt;= 32 Then     Display "My age is between 32 and 35" End If</pre>	
<pre>If yourAge == votingAge OR yourAge &gt; votingAge Then     Display "You can vote" End If</pre>	
<pre>If myNumber == 83 OR yourNumber == 83 Then     Display "One of our numbers is 83" End If</pre>	

**Step 3:** Based on the values to the variables in Step 1, what is the expected output?  
(Reference: Dual Alternative Decision Structures, page 125).

The condition	Expected Output
<pre>If myAge == 31 AND yourAge &lt; myAge Then     Display "My age is 31 and your age is less than that" Else     Display "Our ages do not qualify" End If</pre>	
<pre>If myAge &lt;= 35 AND myAge &gt;= 32 Then     Display "My age is between 32 and 35" Else     Display "My age is not within that range" End If</pre>	
<pre>If yourAge == votingAge OR yourAge &gt; votingAge Then     Display "You can vote" Else     Display "You cannot vote" End If</pre>	
<pre>If myNumber == 83 OR yourNumber == 83 Then     Display "One of our numbers is 83" Else     Display "83 is not our numbers" End If</pre>	

**Lab 4.2 – Pseudocode: Dual Alternative Decisions****Critical Review**

A dual alternative decision structure will execute one group of statements if its Boolean expression is true, or another group if its Boolean expression is false.

The general structure of an if then else statement is

```
If condition Then
    Statement
    Statement
    Etc.
Else
    Statement
    Statement
    Etc.
End If
```

**Module Review**

Recall the difference between a reference variable and a value variable. Reference variables are used in the following lab when the value of the variable is modified in the module. You'll notice some parameter lists include the keyword Ref before the variable that is going to change within the module.

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

Recall the retail company program from Lab 3.2. The company now wants to modify their bonus portion to include different levels and types and eliminate the day off program. The new program is as follows:

```
A retail company assigns a $5000 store bonus if monthly
sales are more than $100,000; otherwise a $500 store bonus
is awarded. Additionally, they are doing away with the
previous day off program and now using a percent of sales
increase to determine if employees get individual bonuses.
If sales increased by at least 4% then all employees get a
$50 bonus. If they do not, then individual bonuses are 0.
```

**Step 1:** To accommodate the changes to the program, create the additional variables needed.

- Create a variable named `storeAmount` to hold the store bonus amount.
- Create a variable named `empAmount` to hold the individual bonus amount.
- Create a variable named `salesIncrease` to hold the percent of increase.

## SODV1101 – Programming Fundamentals

### ILA 4.2

```
//Declare local variables  
Declare Real monthlySales
```

---

---

---

**Step 2:** The first module in the program is getSales(). Since this is still required, leave this module as is. This module should be written as follows:

```
//MODULE 1  
//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
```

**Step 3:** The second module in the program was isBonus(). Since there are two types of bonuses now, rename this module and the module call to storeBonus(). Write an if then else statement within this module that will set the bonus amount to either 5000 or 500. Also, pass the variable storeAmount to the module as a reference. Complete the missing lines. (Reference: Dual Alternative Decision Structures, page 125).

```
//MODULE 2  
//this module will determine what the bonus levels are  
Module _____(Real monthlySales, Real Ref _____)  
    If monthlySales >=100000 Then  
        Set _____ = 5000  
        _____  
        Set _____ = 500  
    End If  
End Module
```

**Step 4:** Write a module that will ask the user to enter the percent of sales increase in decimal format. This module will have to accept salesIncrease as a reference. Complete the missing lines.

```
//MODULE 3  
//this module takes in percent of increase in decimal  
//format such as .02 for 2 percent.  
Module getIncrease(Real Ref _____)  
    Display "_____. "  
    Input _____  
End Module
```

**Step 5:** Write a module that will determine individual bonuses. If the sales increase percent was 4% or more, then all employees get a \$50 bonus. If the sales increase was not reached, then the bonus amount should be set to zero. This module should be called empBonus and accept salesIncrease as a normal variable and empAmount as a reference.

## SODV1101 – Programming Fundamentals

### ILA 4.2

```
//MODULE 4
//this module will determine what the bonus levels are
Module _____ (Real _____, Real Ref _____)
    If salesIncrease >= _____ Then
        Set _____ = 50
    Else
        Set _____ = 0
    End If
End Module
```

**Step 6:** Write a module that will print the store bonus and the employee bonus amount. Name this module printBonus() and pass the two necessary variables.

```
//MODULE 5
//this module will display store and employee bonus info.
Module _____ (Real _____, Real _____)
    Display "The store bonus is $", _____
    Display "The employee bonus is $", _____
End Module
```

**Step 7:** The final step in completing the pseudocode is to call all the modules with the proper arguments. Complete the missing lines.

```
Module main ()
    //Declare local variables
    Declare Real monthlySales
    Declare Real storeAmount
    Declare Real empAmount
    Declare Real salesIncrease

    //Function calls
    Call getSales(monthlySales)
    Call getIncrease(salesIncrease)
    Call _____ (_____, _____)
    Call _____ (_____, _____)
    Call _____ (_____, _____)

End Module
```

**Lab 4.3 – Pseudocode: Nested Decision Structures****Critical Review**

To test more than one condition, a decision structure can be nested inside another decision structure. This structure can become very complex, and often an if-then-else-if statement is used instead.

The general structure of the if-then-else-if statement is

```
If condition_1 Then
    Statement
    Statement
    Etc.
Else If condition_2 Then
    Statement
    Statement
    Etc.
Insert as many Else If clauses as necessary
Else
    Statement
    Statement
    Etc.
End If
```

A case structure lets the value of a variable or an expression determine which path of execution the program will take. This is often used as an alternative to a nested if else decision.

The company now wants to add additional levels to their store and employee bonuses. The new levels are as follows:

Store bonuses:

```
If store sales are $80,000 or more, store bonus is $3000
If store sales are $90,000 or more, store bonus is $4000
If store sales are $100,000 or more, store bonus is $5000
If store sales are $110,000 or more, store bonus is $6000
```

Employee bonuses:

```
If percent of increase is 3% or more, employee bonus is $40
If percent of increase is 4% or more, employee bonus is $50
If percent of increase is 5% or more, employee bonus is $75
```

**Step 1:** Modify the storeBonus module to write a nested if else statement to set the new bonus levels. Complete the missing lines. (Reference: The if-then-else Statement, page 140).

## SODV1101 – Programming Fundamentals

### ILA 4.2

```
//MODULE 2
//this module will determine what the bonus levels are
Module storeBonus (Real monthlySales, Real Ref storeAmount)
    If monthlySales >= 110000 Then
        Set storeAmount = 6000
    Else If monthlySales >= _____ Then
        Set storeAmount = _____
    Else if monthlySales >= _____ Then
        Set storeAmount = _____
    Else if monthlySales >= _____ Then
        Set storeAmount = _____
    Else
        Set storeAmount = 0
    End If
End Module
```

**Step 2:** Modify the empBonus module to write a nested if else statement to set the new bonus levels. Complete the missing lines. (Reference: The if-then-else Statement, page 140).

```
//MODULE 4
//this module will determine what the bonus levels are
Module empBonus (Real salesIncrease, Real Ref empAmount)
    If salesIncrease >= .05 Then
        Set empAmount = 75
    Else If salesIncrease >= _____ Then
        Set empAmount = _____
    Else if salesIncrease >= _____ Then
        Set empAmount = _____
    Else
        Set empAmount = 0
    End If
End Module
```

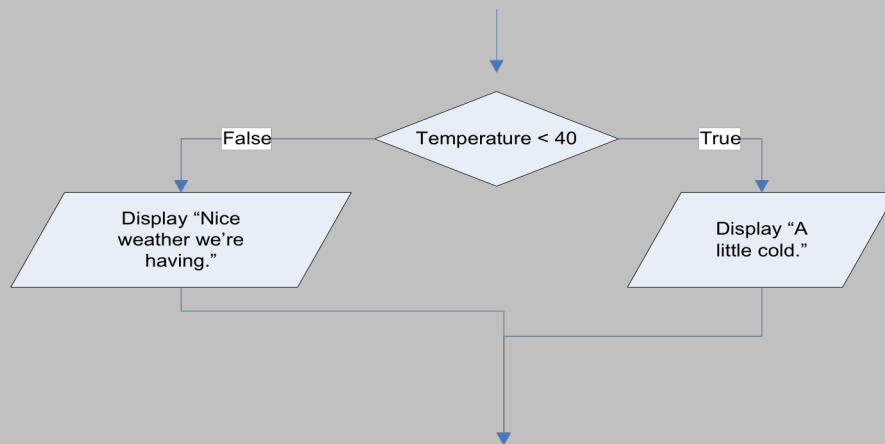
**Step 3:** Modify Module 5 by adding an if statement that will print a message if both the store bonus and the employee bonus are the highest amounts possible. (Reference: Logical Operators, page 147).

```
//MODULE 5
//this module will display store and employee bonus info.
Module printBonus(Real storeAmount, Real empAmount)
    Display "The store bonus is $", storeAmount
    Display "The employee bonus is $", empAmount
    If storeAmount == _____ AND empAmount == _____ Then
        Display "Congrats! You have reached the highest
        bonus amounts possible!"
    End If
End Module
```

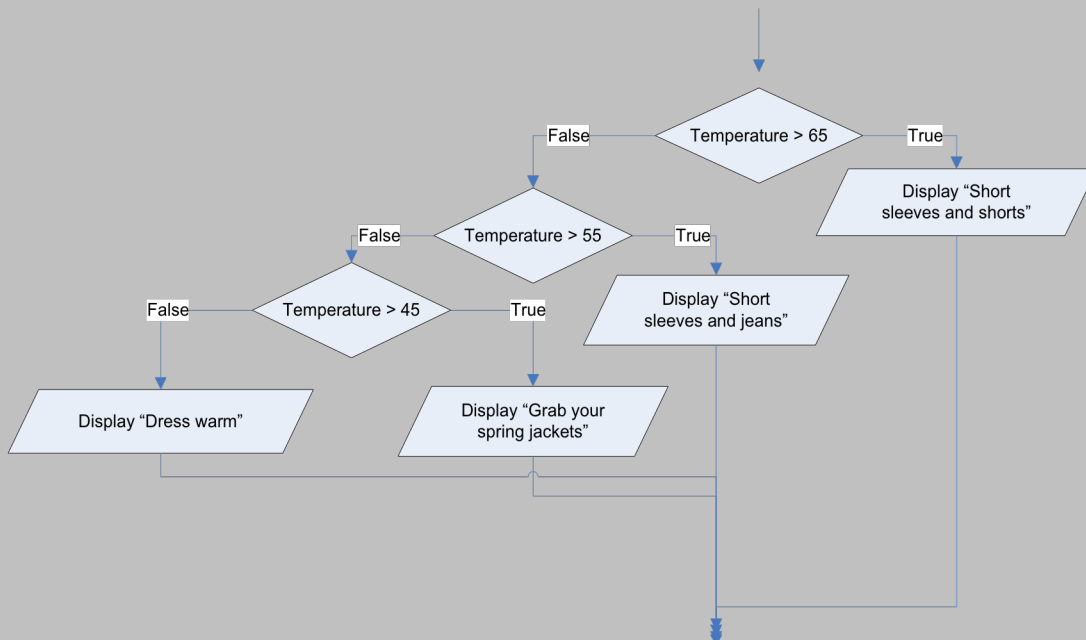
**Lab 4.4 – Flowcharts****Critical Review**

A dual alternative decision structure has two possible paths of execution – one path is taken if a condition is true, and the other path is taken if the condition is false.

A diamond with a true and false value is used in flowcharting a dual alternative decision structure.



Nested if-then-else flowcharts look as follows:





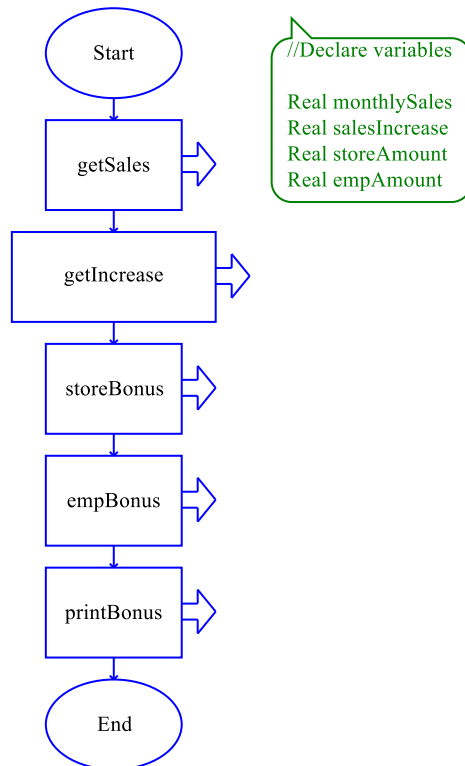
## SODV1101 – Programming Fundamentals

### ILA 4.2

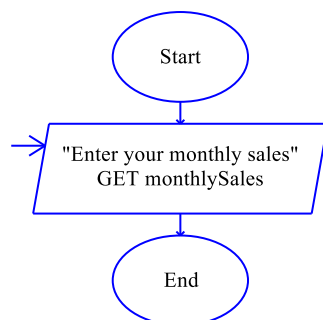
This lab requires you to convert your pseudocode in Lab 4.3 to a flowchart. Use an application such as Raptor or Visio.

**Step 1:** Start Raptor and save your document as *Lab 4-4*. The *.rap* file extension will be added automatically. Start by adding a Comment box that declares your variables.

**Step 2:** The next step in your flowchart should be to call your methods. Main should look as follows. Be sure to click yes to add new tabs for each module.



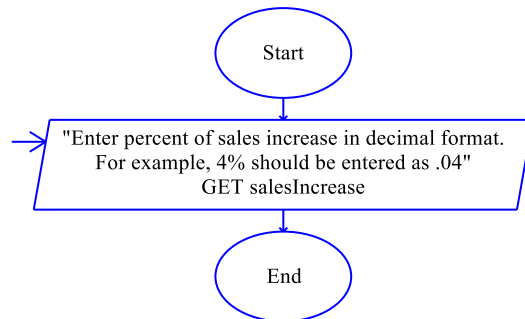
**Step 3:** Click on the getSales tab and add the necessary code to enter the monthly sales. Your getSales method might look like the following:



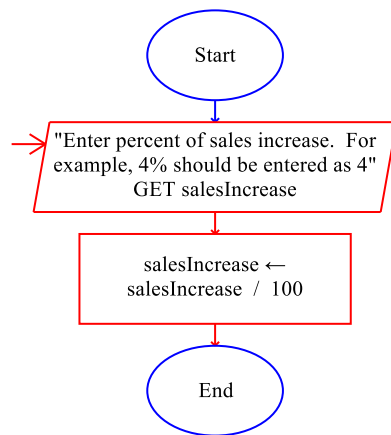
## SODV1101 – Programming Fundamentals

### ILA 4.2

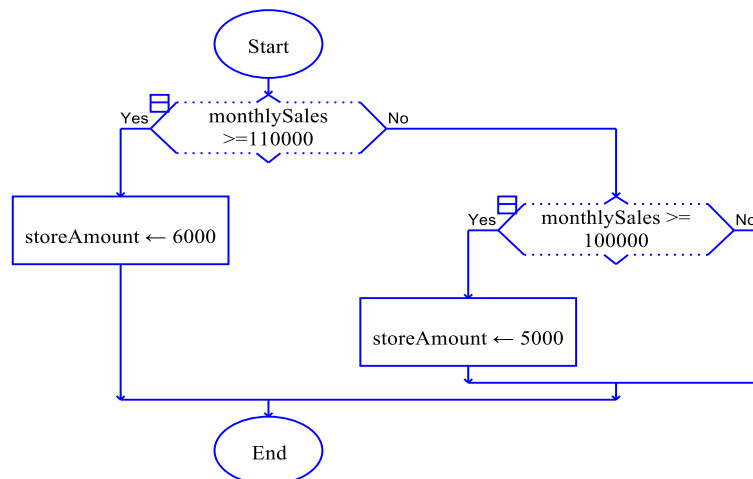
**Step 4:** Click on the `getIncrease` tab and add the necessary code to enter the percent of increase in sales. Since percentages can be entered differently, you should specify the method to the user. Either method is fine, as long as you specify. One method is as follows:



Here is a different method whereas the number entered is divided by 100:



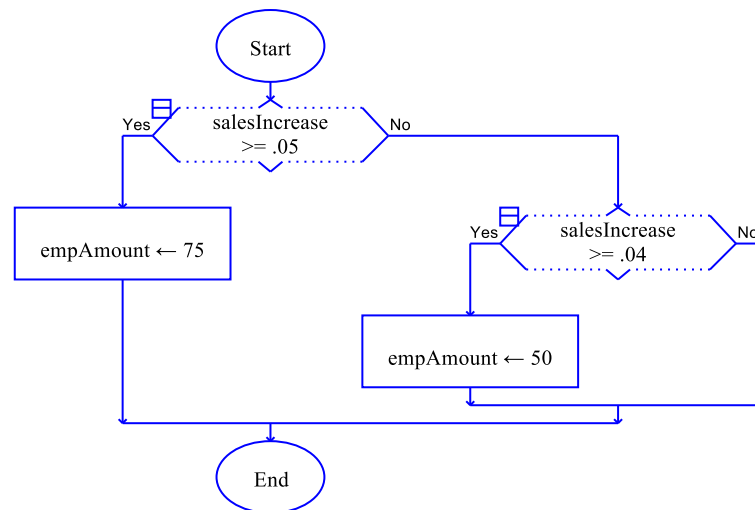
**Step 5:** Click on the `storeBonus` tab and add the nested if-then-else statement from Lab 4.3, Step 1. Do not forget the final else of setting `storeAmount` to 0. The start of your module should look as follows, and you should have a total of four decisions:



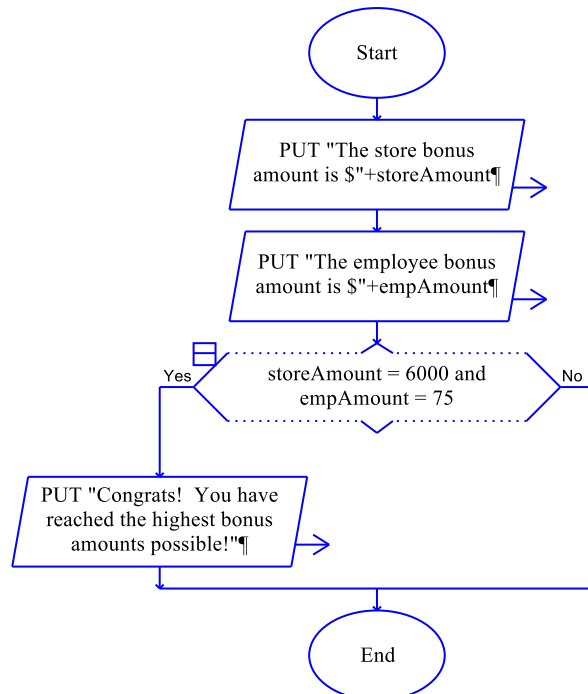
## SODV1101 – Programming Fundamentals

### ILA 4.2

**Step 6:** Click on the empBonus tab and add the nested if-then-else statement from Lab 4.3, Step 2. Do not forget the final else of setting empAmount to 0. The start of your module should look as follows, and you should have a total of three decisions:



**Step 7:** Click the printBonus tab and add the necessary code from Lab 4.3, Step 3. The module should look as follows:



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

## SODV1101 – Programming Fundamentals

### ILA 4.2

Input Values	Expected Output
monthlySales = 120500 salesIncrease = 5	The store bonus amount is \$6000 The employee bonus amount is \$75 Congrats! You have reached the highest bonus amounts possible!
monthlySales = 93400 salesIncrease = 5	The store bonus amount is \$4000 The employee bonus amount is \$75
monthlySales = 75000 salesIncrease = 1.5	The store bonus amount is \$0 The employee bonus amount is \$0
monthlySales = 82000 salesIncrease = 3.6	The store bonus amount is \$3000 The employee bonus amount is \$40
monthlySales = 125000 salesIncrease = 4.5	The store bonus amount is \$6000 The employee bonus amount is \$50

**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**

**Lab 4.5 – Python Code****Critical Review**

In code we write a dual alternative decision structure as an `if-else` statement. Here is the general format of the `if-else` statement:

```
if condition:
    statement
    statement
    etc.
else:
    statement
    statement
    etc.
```

Here is the general format of the `if-elif-else` statement:

```
if condition_1:
    statement
    statement
    etc.
elif condition_2:
    statement
    statement
    etc.
```

*Insert as many `elif` clauses as necessary...*

```
else:
    statement
    statement
    etc.
```

The logical operators `and`, `or`, and `not` are used in Python to connect Boolean expressions.

**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 *Lab4-5.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.

## SODV1101 – Programming Fundamentals

### ILA 4.2

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

```
#Lab 4-5
#The main function
def main():
    monthlySales = getSales() #call to get sales

#This function gets the monthly sales
def getSales():
    monthlySales = input('Enter the monthly sales $')
    monthlySales = float(monthlySales)
    return monthlySales

#This function gets the percent of increase in sales
def getIncrease():

#This function determines the storeAmount bonus
def storeBonus():

#This function determines the empAmount bonus
def empBonus():

#This function prints the bonus information
def printBonus():

#calls main
main()
```

**Step 4:** Under the getIncrease function, add the necessary code to allow the user to input sales increase. Your code might look as follows:

```
#This function gets the percent of increase in sales
def getIncrease():
    salesIncrease = input('Enter percent of sales increase.
    For example 4% should be entered as 4: ')
    salesIncrease = float(salesIncrease)
    salesIncrease = salesIncrease / 100
    return salesIncrease
```

**Step 5:** Under the call to getSales(), add a function call to getIncrease().

```
salesIncrease = getIncrease() #call to get sales increase
```

**Step 6:** Under the storeBonus function, add the necessary code so that the program will determine what the proper storeAmount variable should have. This function might look as follows:

## SODV1101 – Programming Fundamentals

### ILA 4.2

```
#This function determines the storeAmount bonus
def storeBonus(monthlySales):
    if monthlySales >=110000:
        storeAmount = 6000
    elif monthlySales >=100000:
        storeAmount = 5000
    elif monthlySales >=90000:
        storeAmount = 4000
    elif monthlySales >=80000:
        storeAmount = 3000
    else:
        storeAmount = 0
    return storeAmount
```

**Step 7:** Under the call to `getIncrease()`, add a function call to `storeBonus()`.

```
#call to get the store bonus
storeAmount = storeBonus(monthlySales)
```

**Step 8:** Repeat the similar process in step 6 and 7 for writing the `empBonus()` function and making a call to it. Recall that this function uses `salesIncrease` to determine `empAmount`.

**Step 9:** Code the `printBonus()` function to print the `storeAmount` and `empAmount`. This function might look as follows:

```
#This function prints the bonus information
def printBonus(storeAmount, empAmount):
    print 'The store bonus amount is $', storeAmount
    print 'The employee bonus amount is $', empAmount
    if storeAmount == 6000 and empAmount == 75:
        print 'Congrats! You have reached the highest bonus
amounts possible!'
```

**Step 10:** Under the call to `empBonus()`, add a function call to `printBonus`. This call might look as follows:

```
printBonus(storeAmount, empAmount) #call to print amounts
```

**Step 11:** Click Run and Run Module to see how your program processes. Test the following values to verify the expected output.

Input Values	Expected Output
monthlySales = 120500 salesIncrease = 5	The store bonus amount is \$ 6000 The employee bonus amount is \$ 75 Congrats! You have reached the highest bonus amounts possible!
monthlySales = 93400 salesIncrease = 5	The store bonus amount is \$4000 The employee bonus amount is \$75

## SODV1101 – Programming Fundamentals

### ILA 4.2

monthlySales = 75000 salesIncrease = 1.5	The store bonus amount is \$0 The employee bonus amount is \$0
monthlySales = 82000 salesIncrease = 3.6	The store bonus amount is \$3000 The employee bonus amount is \$40
monthlySales = 125000 salesIncrease = 4.5	The store bonus amount is \$6000 The employee bonus amount is \$50

**Step 12:** Execute your program so that it works and paste the final code below

**PASTE CODE HERE**