SAMPLE PROGRAMS TO HELP WITH ASSIGNMENTS FOR THE IF SYMBOLS & STATEMENT

RESOURCES NEEDED TO COMPLETE ASSIGNMENT: **<u>Read Chapter 3</u> – of the textbook;** **See LINKS & VIDEOS under CONTENT LINK ON SELECTION / IF STATEMENTS.**	**Score**	**Your Score**
<u>OBJECTIVE: FOR FLOWGORITHM</u>- Illustrate Control Structures using the if statement by entering a program to calculate gross pay. If the hours worked is greater than 40, determine the overtime hours, overtime pay and regular pay. You will display all input and output results as illustrated.		

<u>Description for IF STATEMENT ASSIGNMENTS</u>

The IF statement is used to create a decision structure or construct. It alters or changes the control or flow of a flowchart and an actual program. Thus, **the IF statement** is considered to be an example of a control structure. "A control structure is a logical design that controls the order in which a set of statement execute" (Tony Gaddis, 2018).

In the previous assignments the flowchart and program will execute the symbols and statements according to the sequential order we entered them. They are called sequential statements or the sequential construct. The if statement will be used in the gross pay example to control the order or make a decision to calculate the gross pay based on the number of hours worked.

The objective is to calculate the gross pay of an employee.

1. If the employee works 40 hours or less the gross pay is obtained by multiplying the hours worked by the rate of pay.

2. On the other hand, if an employee works more than 40 hours per week, then the overtime hours, and overtime pay must be calculated.

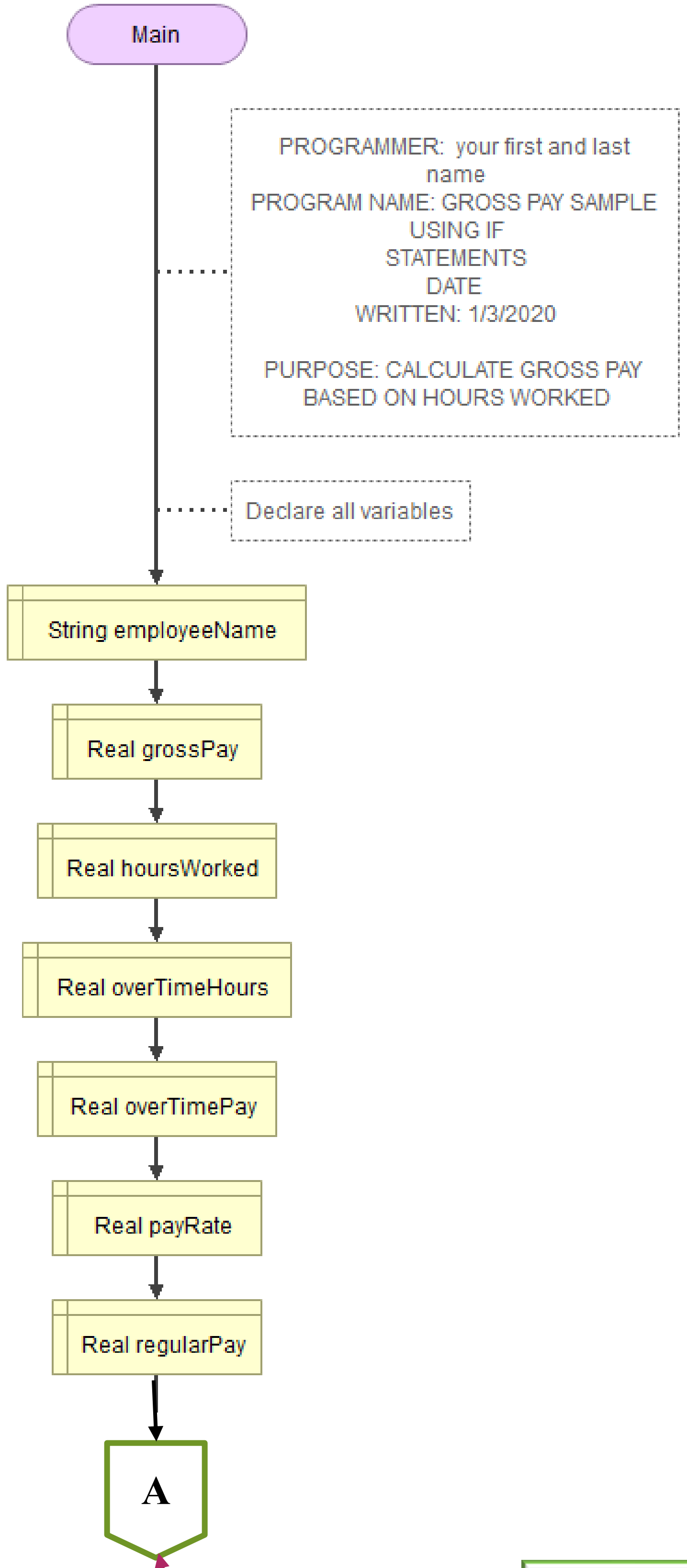
One approach would be to determine the overtime hours, overtime pay, and regular pay. Next, determine the gross pay based on the previous calculations.

<u>Be sure to copy the attached practice programs:</u>

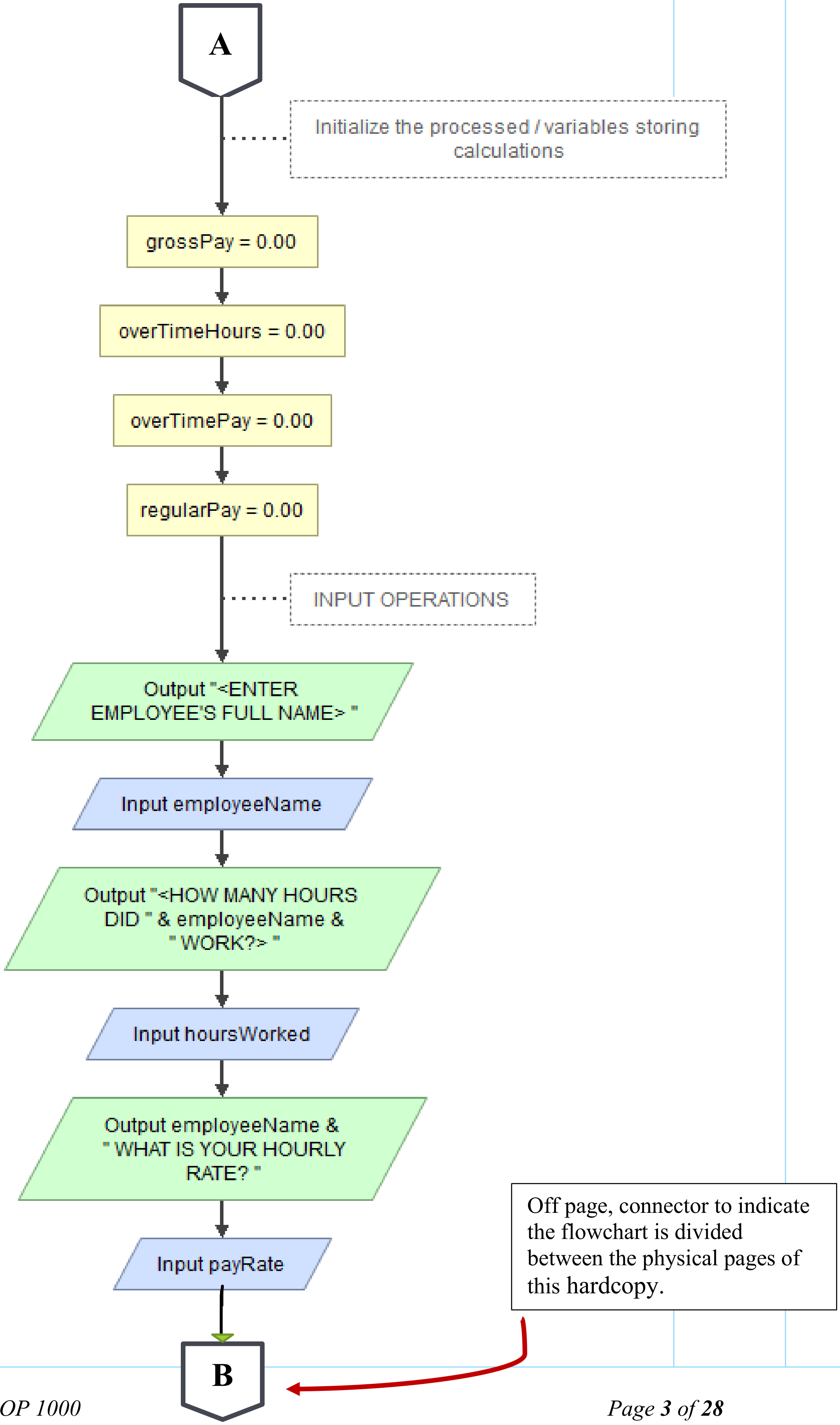
Lastname_firstname_A3_GROSS_PAY_IF_STATEMENT.fprg

Lastname_firstname_A3_GROSS_PAY_IF_STATEMENT.py

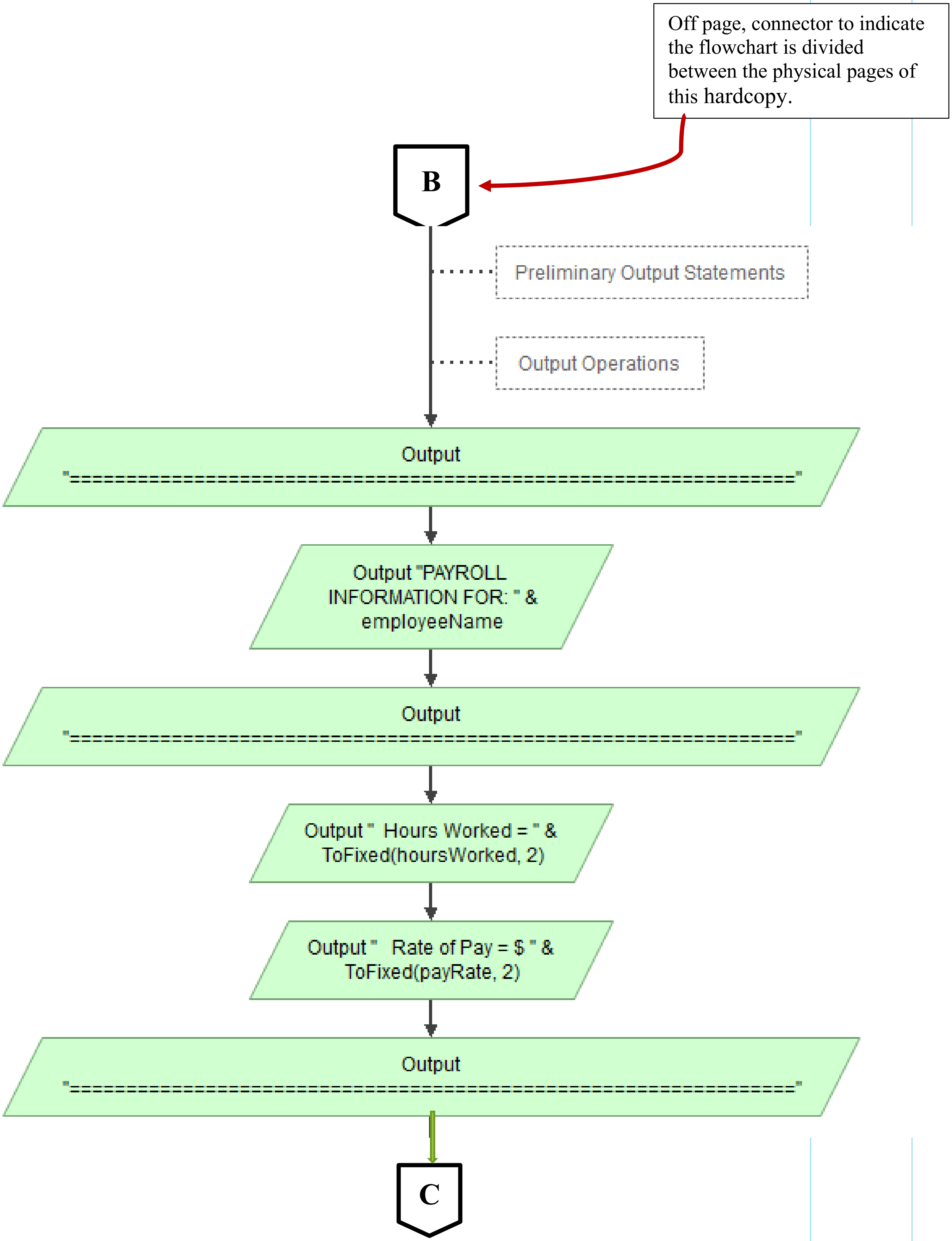
<u>Look at the FOLLOWING FLOWGORITHM ILLUSTRATION:</u>



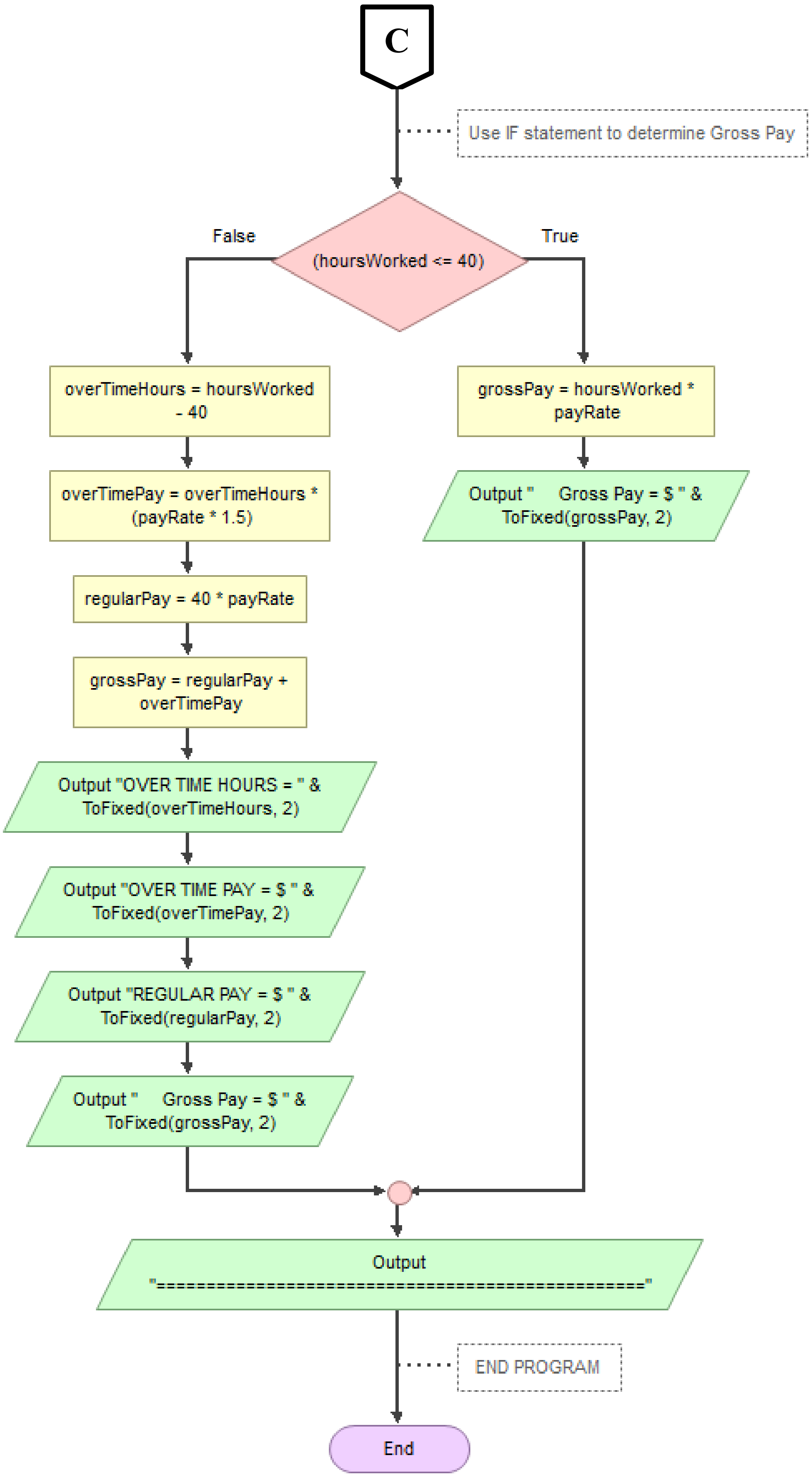
Off page connector to indicate that the flowchart is divided between the physical pages / (**you will not find this symbol in FLOWGORITHM**)

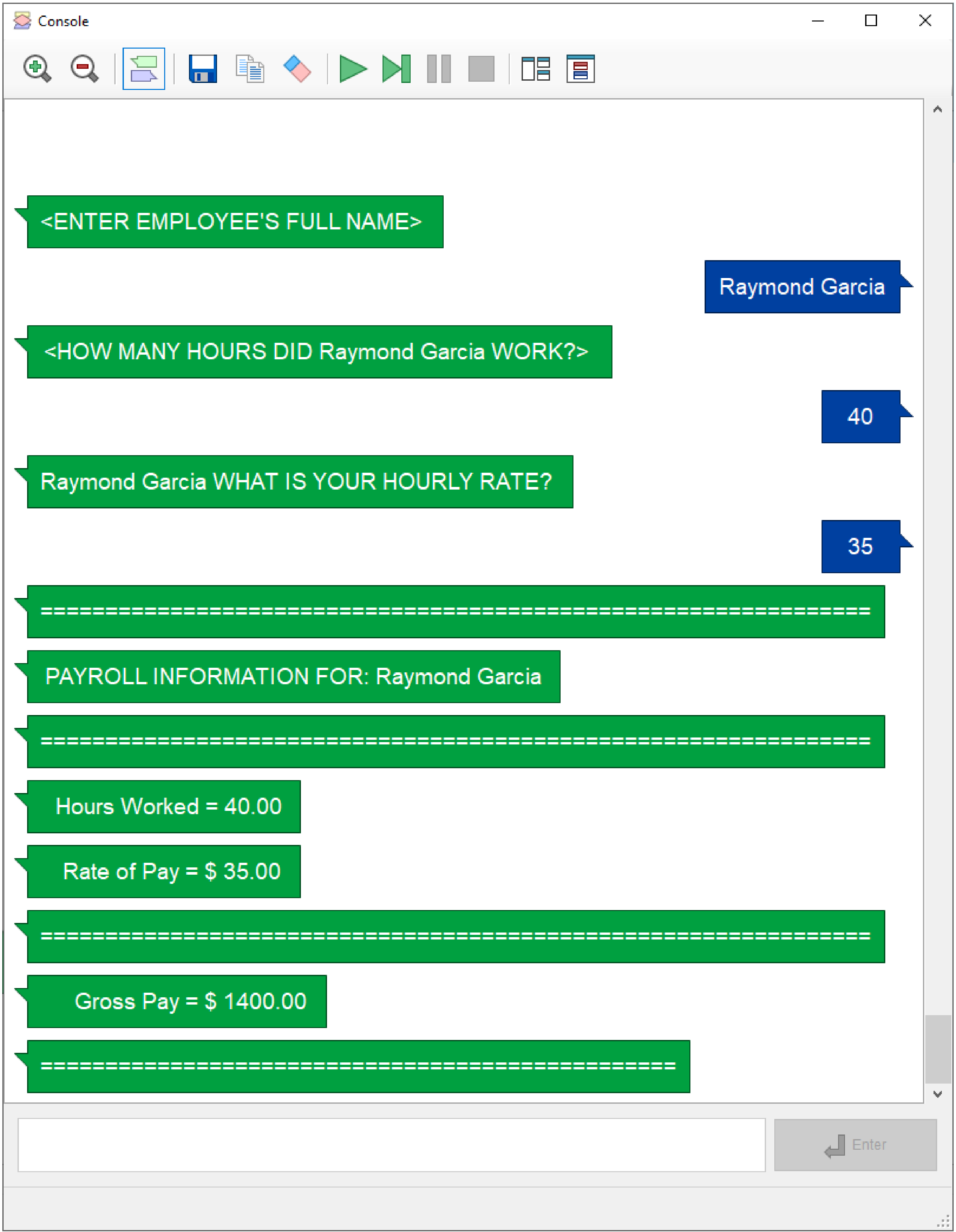

A
Initialize the processed / variables storing calculations
grossPay = 0.00
overTimeHours = 0.00
overTimePay = 0.00
regularPay = 0.00
INPUT OPERATIONS
Output "<ENTER EMPLOYEE'S FULL NAME> "
Input employeeName
Output "<HOW MANY HOURS DID " & employeeName & " WORK?> "
Input hoursWorked
Output employeeName & " WHAT IS YOUR HOURLY RATE? "
Input payRate
B

Off page, connector to indicate the flowchart is divided between the physical pages of this hardcopy.

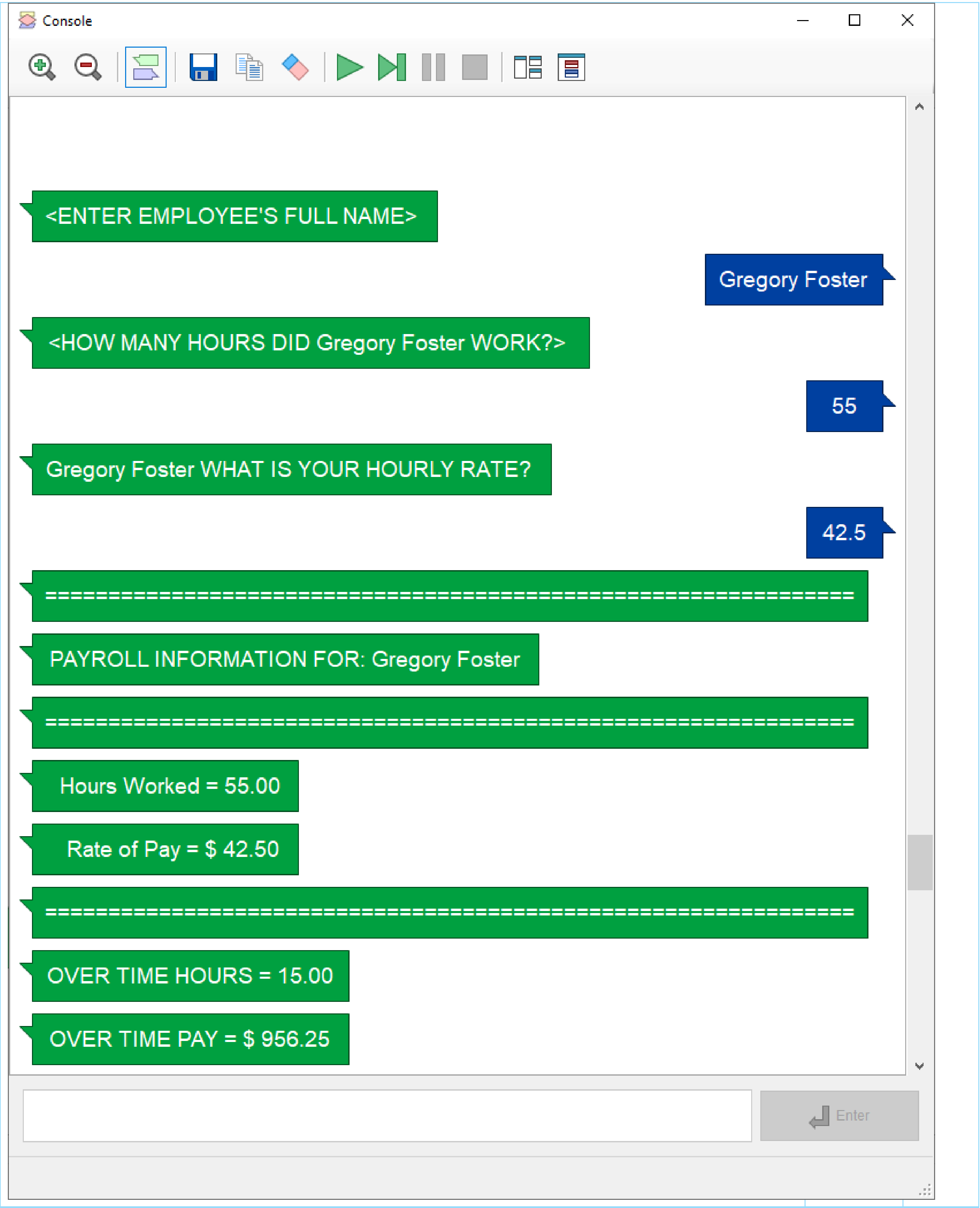

B
Preliminary Output Statements
Output Operations
Output "==="
Output "PAYROLL INFORMATION FOR: " & employeeName
Output "==="
Output " Hours Worked = " & ToFixed(hoursWorked, 2)
Output "   Rate of Pay = $ " & ToFixed(payRate, 2)
Output "==="
C

Off page, connector to indicate the flowchart is divided between the physical pages of this hardcopy.


C
Use IF statement to determine Gross Pay
False
(hoursWorked <= 40)
True
overTimeHours = hoursWorked - 40
grossPay = hoursWorked * payRate
overTimePay = overTimeHours * (payRate * 1.5)
Output "   Gross Pay = $ " & ToFixed(grossPay, 2)
regularPay = 40 * payRate
grossPay = regularPay + overTimePay
Output "OVER TIME HOURS = " & ToFixed(overTimeHours, 2)
Output "OVER TIME PAY = $ " & ToFixed(overTimePay, 2)
Output "REGULAR PAY = $ " & ToFixed(regularPay, 2)
Output "   Gross Pay = $ " & ToFixed(grossPay, 2)
Output "=="
END PROGRAM
End



```
<ENTER EMPLOYEE'S FULL NAME>
                                                  Raymond Garcia
<HOW MANY HOURS DID Raymond Garcia WORK?>
                                                              40
Raymond Garcia WHAT IS YOUR HOURLY RATE?
                                                              35
=================================================================
PAYROLL INFORMATION FOR: Raymond Garcia
=================================================================
  Hours Worked = 40.00
   Rate of Pay = $ 35.00
=================================================================
     Gross Pay = $ 1400.00
==================================================
```



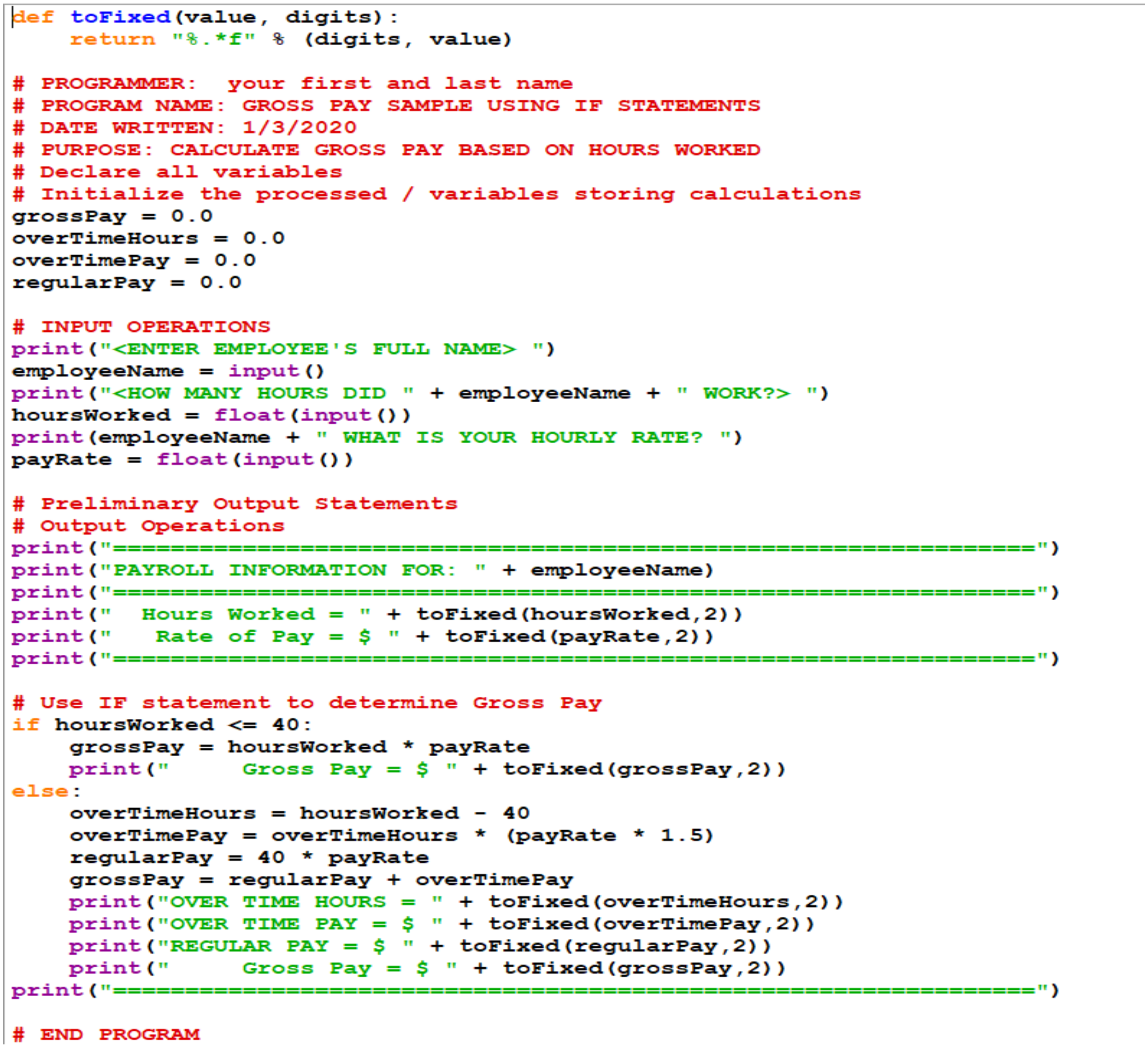
```
<ENTER EMPLOYEE'S FULL NAME>
Gregory Foster
<HOW MANY HOURS DID Gregory Foster WORK?>
55
Gregory Foster WHAT IS YOUR HOURLY RATE?
42.5
=====================================================================
PAYROLL INFORMATION FOR: Gregory Foster
=====================================================================
  Hours Worked = 55.00
   Rate of Pay = $ 42.50
=====================================================================
OVER TIME HOURS = 15.00
OVER TIME PAY = $ 956.25
```

1. □ Save the above practice FLOWGORITHM program as: a. Lastname_firstname_A3_GROSS_PAY_IF_STATEMENT.fprg		
2. □ Add appropriate comments as outlined in the illustration.		
3. □ Enter all appropriate calculations with IF statement(s).		
4. □ Enter all the required input statements illustrated		
5. □ Enter all required output statements as illustrated		

PART 2–Expressing the Flowgorithm program in Python

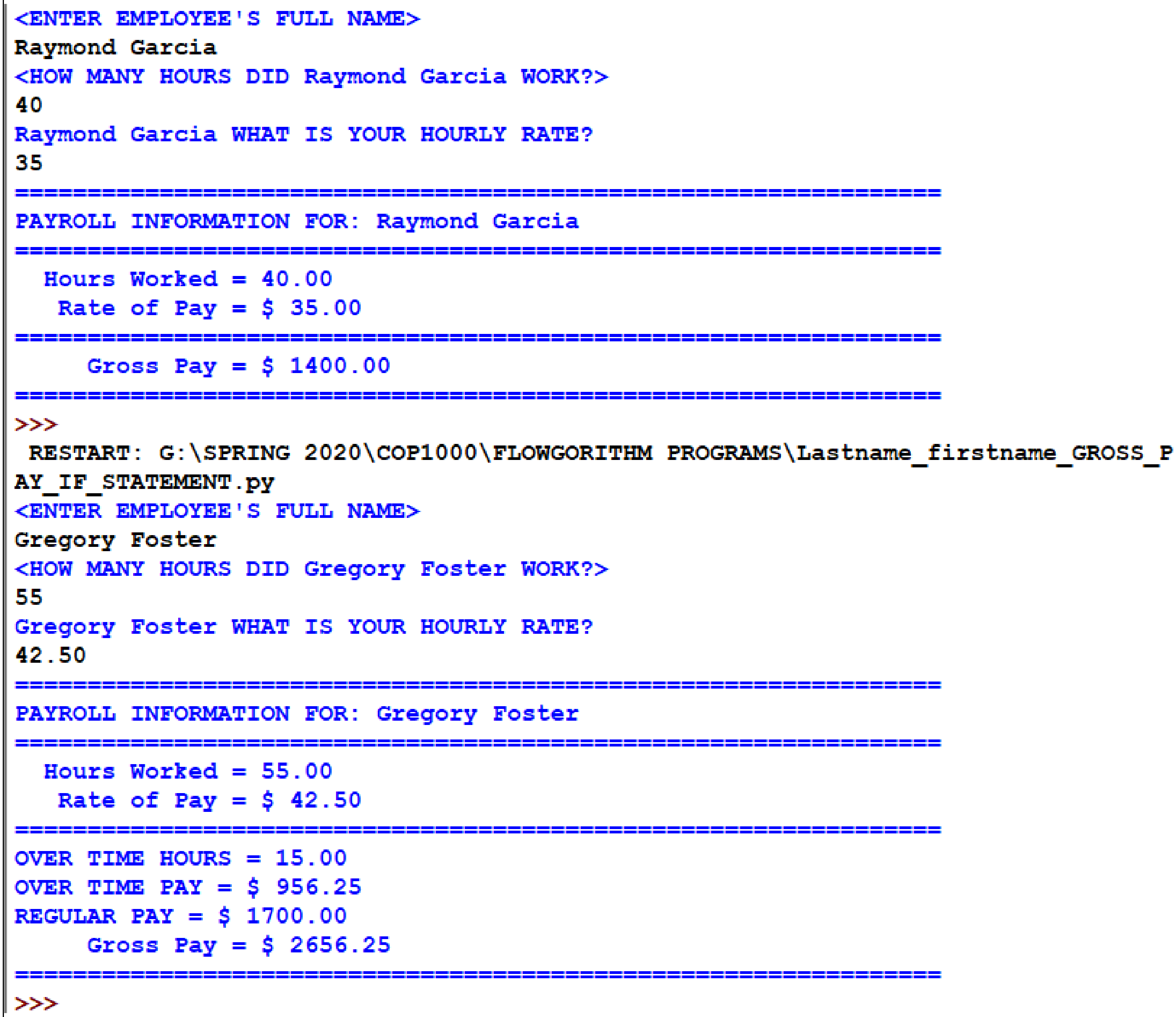
1. □ Click the Source Code Viewer and choose Python to convert to actual python code.			
2. □ Save the program. It will automatically save with the same name used in Flowgorithm but with the .py extension.			
3. □ Open the file.			

YOUR PYTHON ASSIGNMENT WILL RESEMBLE THE FOLLOWING:



```
def toFixed(value, digits):
    return "%.*f" % (digits, value)

# PROGRAMMER:  your first and last name
# PROGRAM NAME: GROSS PAY SAMPLE USING IF STATEMENTS
# DATE WRITTEN: 1/3/2020
# PURPOSE: CALCULATE GROSS PAY BASED ON HOURS WORKED
# Declare all variables
# Initialize the processed / variables storing calculations
grossPay = 0.0
overTimeHours = 0.0
overTimePay = 0.0
regularPay = 0.0

# INPUT OPERATIONS
print("<ENTER EMPLOYEE'S FULL NAME> ")
employeeName = input()
print("<HOW MANY HOURS DID " + employeeName + " WORK?> ")
hoursWorked = float(input())
print(employeeName + " WHAT IS YOUR HOURLY RATE? ")
payRate = float(input())

# Preliminary Output Statements
# Output Operations
print("===============================================================")
print("PAYROLL INFORMATION FOR:  " + employeeName)
print("===============================================================")
print("  Hours Worked = " + toFixed(hoursWorked,2))
print("   Rate of Pay = $ " + toFixed(payRate,2))
print("===============================================================")

# Use IF statement to determine Gross Pay
if hoursWorked <= 40:
    grossPay = hoursWorked * payRate
    print("     Gross Pay = $ " + toFixed(grossPay,2))
else:
    overTimeHours = hoursWorked - 40
    overTimePay = overTimeHours * (payRate * 1.5)
    regularPay = 40 * payRate
    grossPay = regularPay + overTimePay
    print("OVER TIME HOURS = " + toFixed(overTimeHours,2))
    print("OVER TIME PAY = $ " + toFixed(overTimePay,2))
    print("REGULAR PAY = $ " + toFixed(regularPay,2))
    print("     Gross Pay = $ " + toFixed(grossPay,2))
print("===============================================================")

# END PROGRAM
```

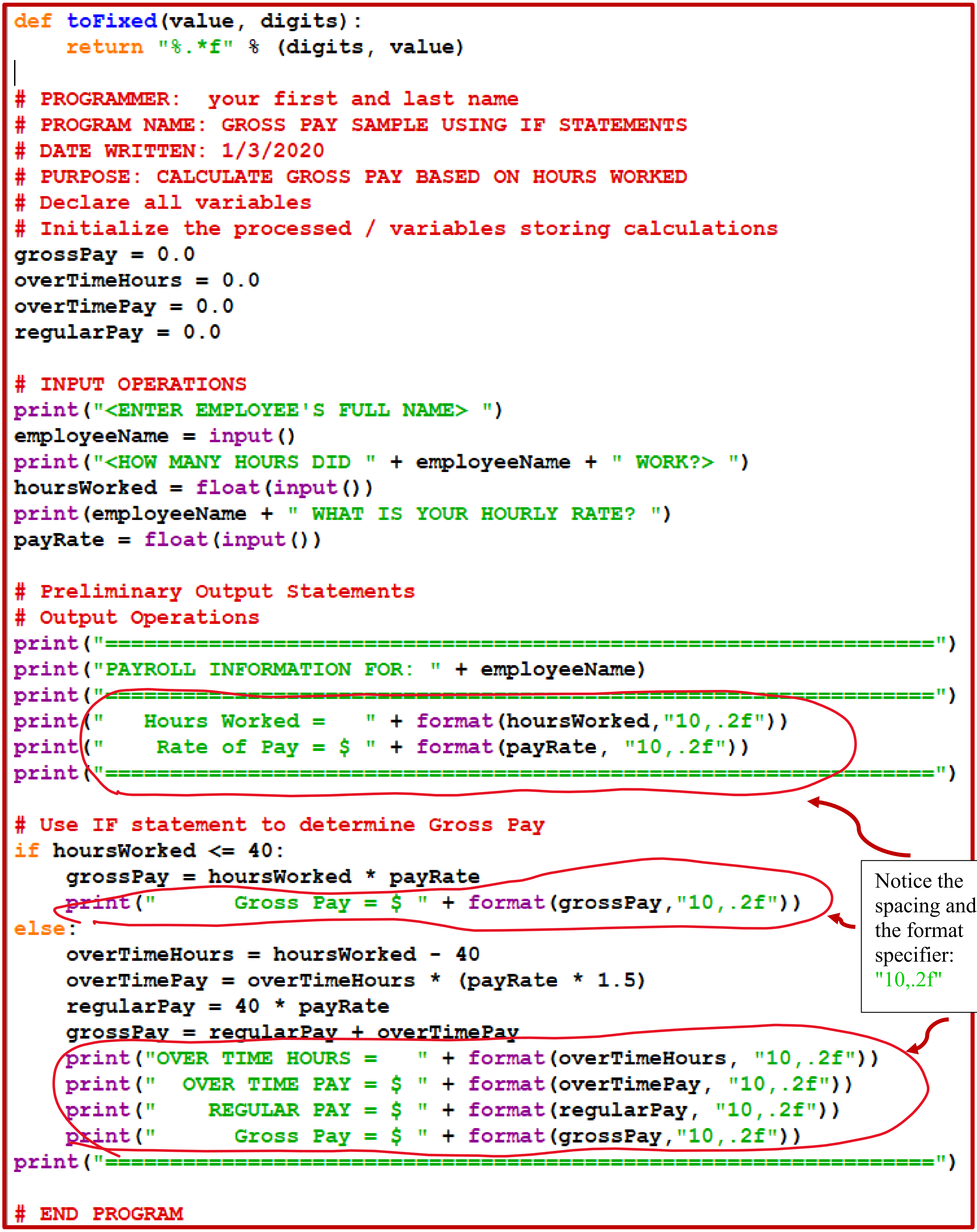
OUTPUT WILL RESEMBLE THE FOLLOWING:



```
<ENTER EMPLOYEE'S FULL NAME>
Raymond Garcia
<HOW MANY HOURS DID Raymond Garcia WORK?>
40
Raymond Garcia WHAT IS YOUR HOURLY RATE?
35
================================================================
PAYROLL INFORMATION FOR: Raymond Garcia
================================================================
  Hours Worked = 40.00
   Rate of Pay = $ 35.00
================================================================
     Gross Pay = $ 1400.00
================================================================
>>> 
 RESTART: G:\SPRING 2020\COP1000\FLOWGORITHM PROGRAMS\Lastname_firstname_GROSS_P
AY_IF_STATEMENT.py
<ENTER EMPLOYEE'S FULL NAME>
Gregory Foster
<HOW MANY HOURS DID Gregory Foster WORK?>
55
Gregory Foster WHAT IS YOUR HOURLY RATE?
42.50
================================================================
PAYROLL INFORMATION FOR: Gregory Foster
================================================================
  Hours Worked = 55.00
   Rate of Pay = $ 42.50
================================================================
OVER TIME HOURS = 15.00
OVER TIME PAY = $ 956.25
REGULAR PAY = $ 1700.00
     Gross Pay = $ 2656.25
================================================================
>>> 
```

The results look good but are not aligned vertically by the decimal point.

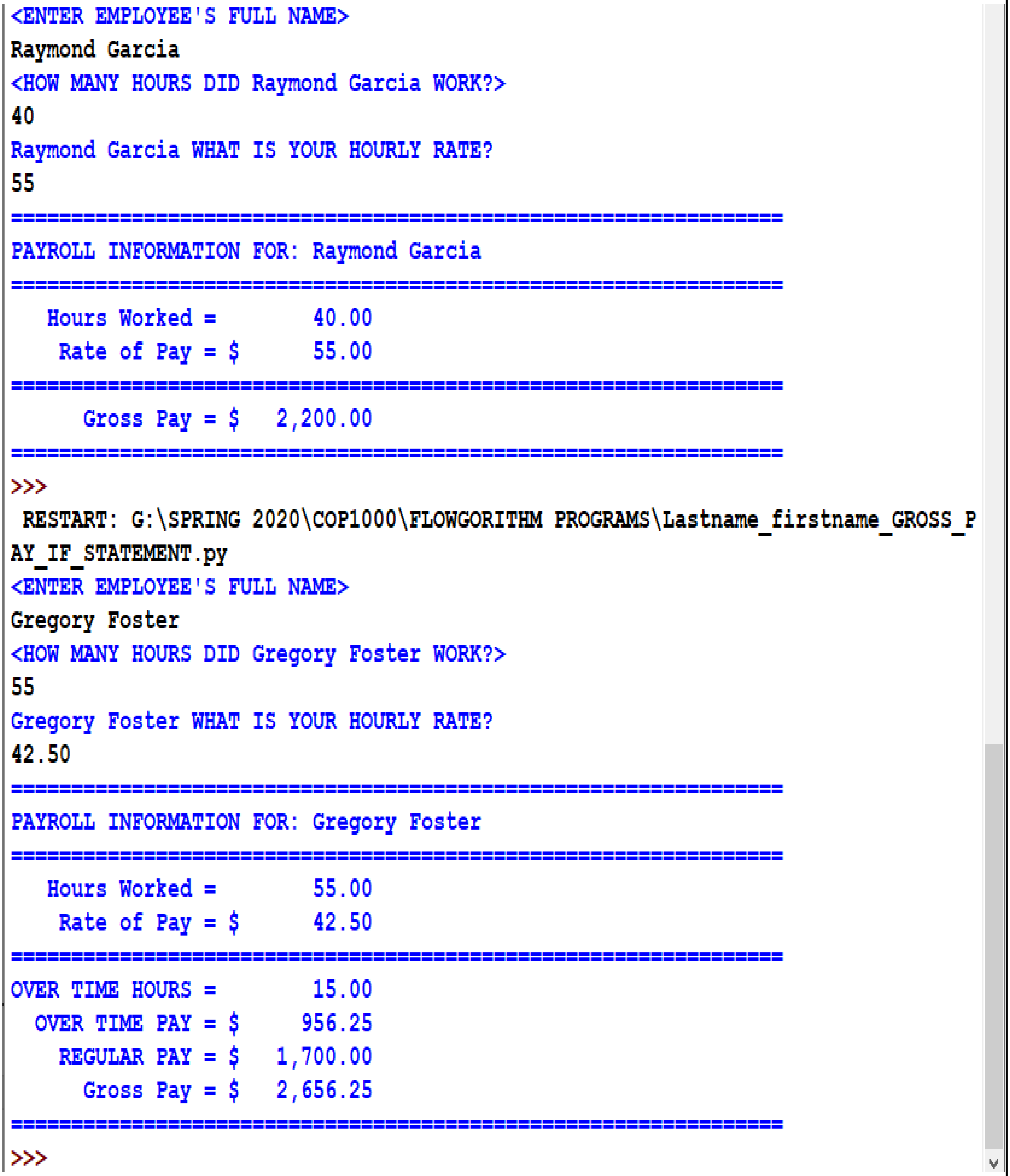
1. **□ To accomplish this task the string literal labels, need to be adjusted by inserting blank spaces along with replacing the ToFixed function from flowgorithm to the format function in python as in the following illustration. Change the toFixed function to format in all the print statements only.**
2. **□ Additionally, add the format specifier "10,.2f" to all the numerical values which are to be printed as in the following illustration.**



```python
def toFixed(value, digits):
    return "%.*f" % (digits, value)

# PROGRAMMER:  your first and last name
# PROGRAM NAME: GROSS PAY SAMPLE USING IF STATEMENTS
# DATE WRITTEN: 1/3/2020
# PURPOSE: CALCULATE GROSS PAY BASED ON HOURS WORKED
# Declare all variables
# Initialize the processed / variables storing calculations
grossPay = 0.0
overTimeHours = 0.0
overTimePay = 0.0
regularPay = 0.0

# INPUT OPERATIONS
print("<ENTER EMPLOYEE'S FULL NAME> ")
employeeName = input()
print("<HOW MANY HOURS DID " + employeeName + " WORK?> ")
hoursWorked = float(input())
print(employeeName + " WHAT IS YOUR HOURLY RATE? ")
payRate = float(input())

# Preliminary Output Statements
# Output Operations
print("================================================================")
print("PAYROLL INFORMATION FOR:  " + employeeName)
print("================================================================")
print("   Hours Worked =   " + format(hoursWorked,"10,.2f"))
print("    Rate of Pay = $ " + format(payRate, "10,.2f"))
print("================================================================")

# Use IF statement to determine Gross Pay
if hoursWorked <= 40:
    grossPay = hoursWorked * payRate
    print("      Gross Pay = $ " + format(grossPay,"10,.2f"))
else:
    overTimeHours = hoursWorked - 40
    overTimePay = overTimeHours * (payRate * 1.5)
    regularPay = 40 * payRate
    grossPay = regularPay + overTimePay
    print("OVER TIME HOURS =   " + format(overTimeHours, "10,.2f"))
    print("  OVER TIME PAY = $ " + format(overTimePay, "10,.2f"))
    print("    REGULAR PAY = $ " + format(regularPay, "10,.2f"))
    print("      Gross Pay = $ " + format(grossPay,"10,.2f"))
print("================================================================")

# END PROGRAM
```

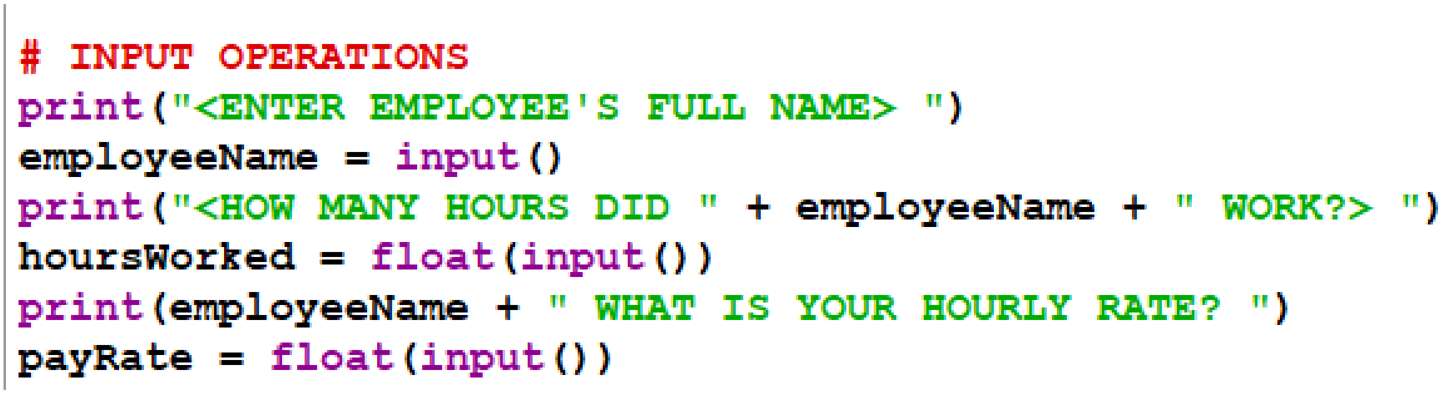
Notice the spacing and the format specifier: "10,.2f"

RUN THE PROGRAM USING THE FOLLOWING OUTPUT; NOTICE THE REULSTS:



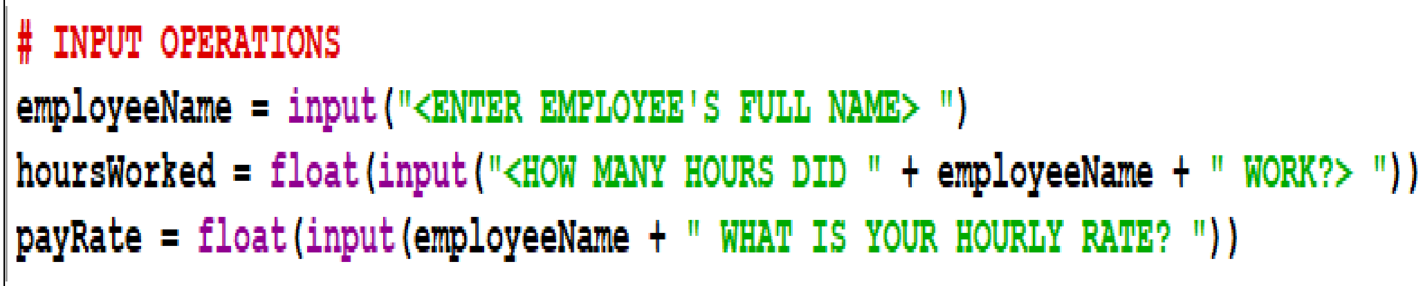
```
<ENTER EMPLOYEE'S FULL NAME>
Raymond Garcia
<HOW MANY HOURS DID Raymond Garcia WORK?>
40
Raymond Garcia WHAT IS YOUR HOURLY RATE?
55
================================================================
PAYROLL INFORMATION FOR: Raymond Garcia
================================================================
   Hours Worked =        40.00
    Rate of Pay = $      55.00
================================================================
      Gross Pay = $   2,200.00
================================================================
>>>
 RESTART: G:\SPRING 2020\COP1000\FLOWGORITHM PROGRAMS\Lastname_firstname_GROSS_P
AY_IF_STATEMENT.py
<ENTER EMPLOYEE'S FULL NAME>
Gregory Foster
<HOW MANY HOURS DID Gregory Foster WORK?>
55
Gregory Foster WHAT IS YOUR HOURLY RATE?
42.50
================================================================
PAYROLL INFORMATION FOR: Gregory Foster
================================================================
   Hours Worked =        55.00
    Rate of Pay = $      42.50
================================================================
OVER TIME HOURS =        15.00
  OVER TIME PAY = $     956.25
    REGULAR PAY = $   1,700.00
      Gross Pay = $   2,656.25
================================================================
>>>
```

COMBINING THE PRINT/INPUT STATEMENTS:

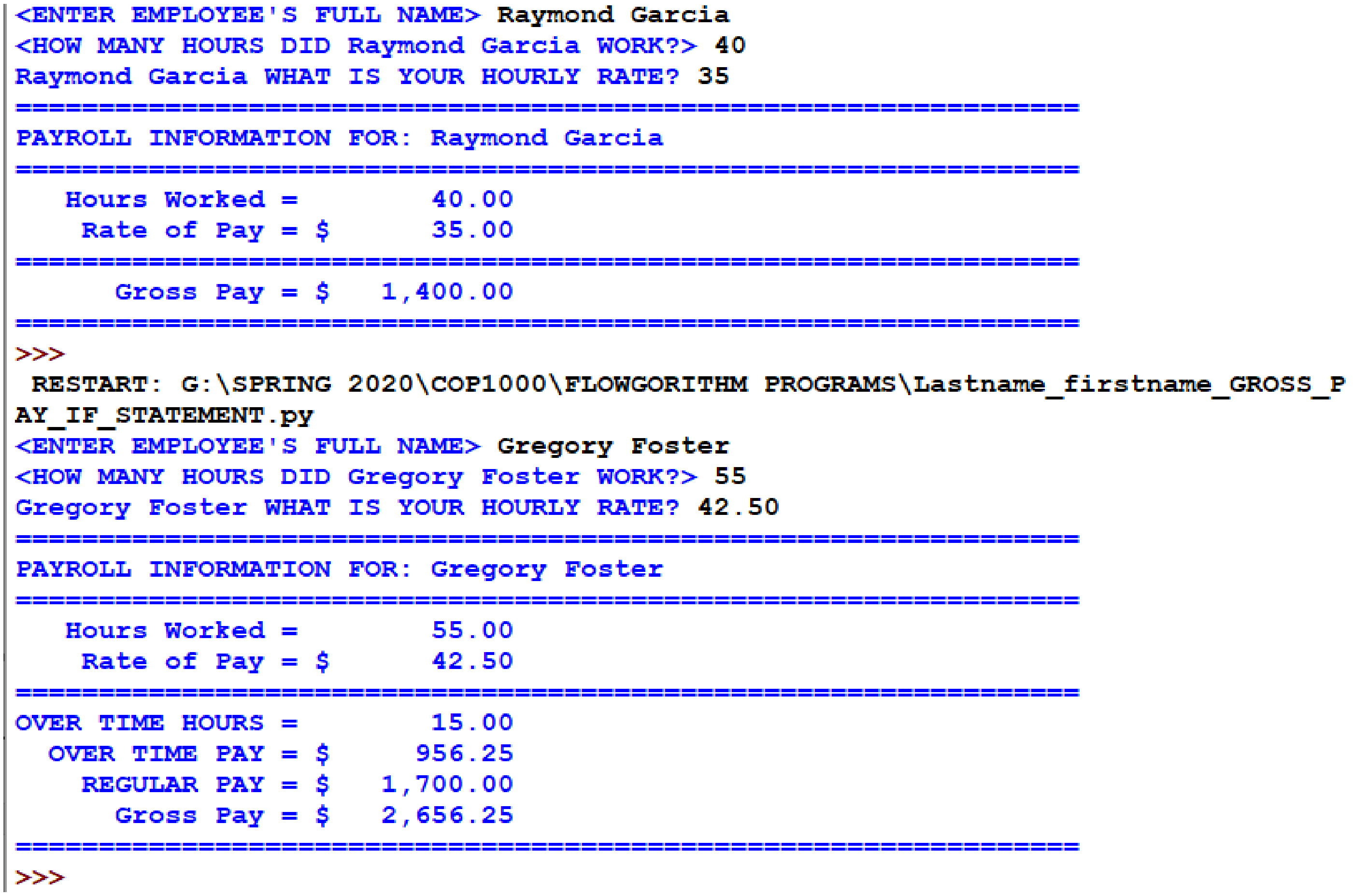


```
# INPUT OPERATIONS
print("<ENTER EMPLOYEE'S FULL NAME> ")
employeeName = input()
print("<HOW MANY HOURS DID " + employeeName + " WORK?> ")
hoursWorked = float(input())
print(employeeName + " WHAT IS YOUR HOURLY RATE? ")
payRate = float(input())
```

Chapter 2 illustrates the print/input statements combined as one.

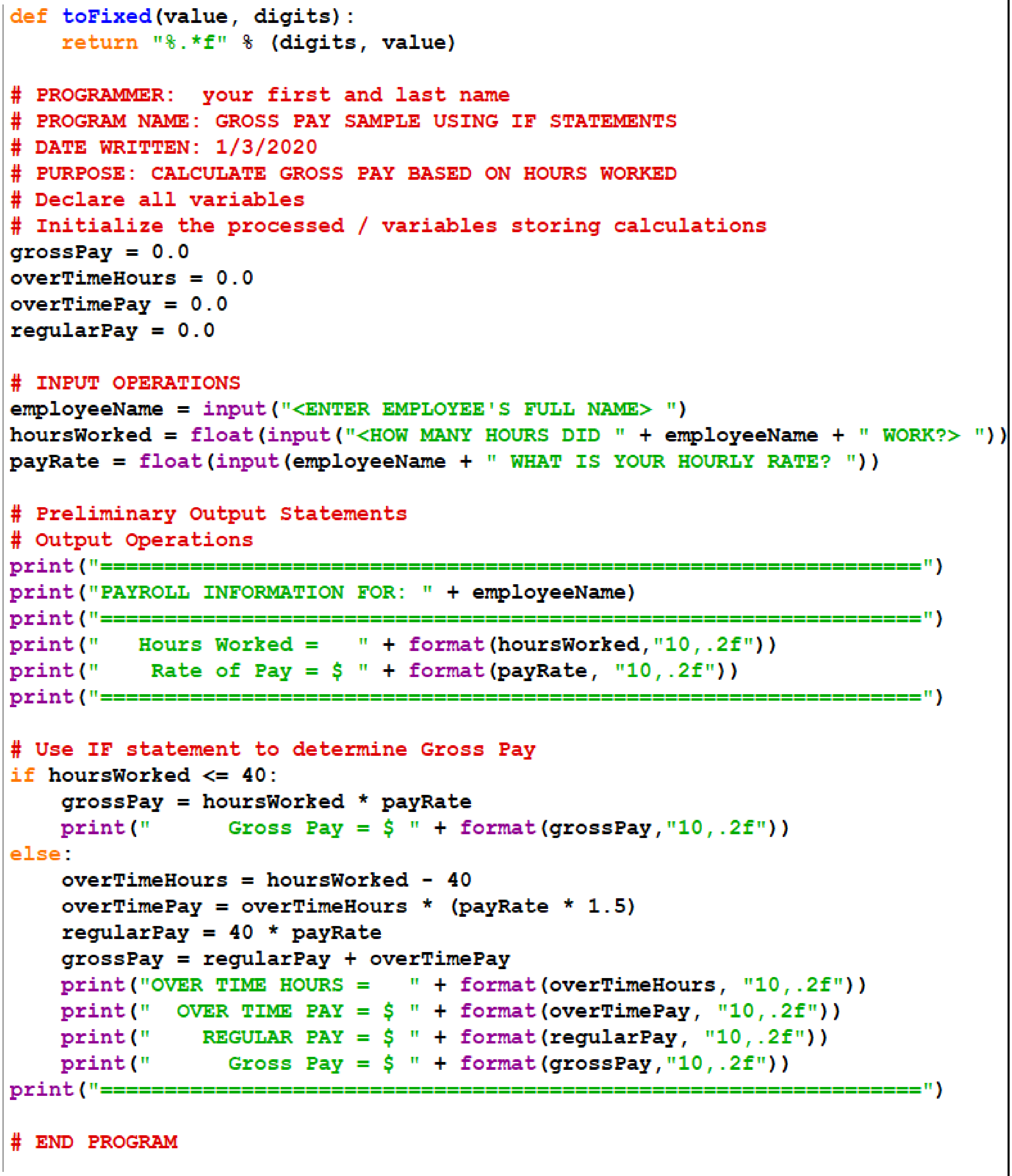


```
# INPUT OPERATIONS
employeeName = input("<ENTER EMPLOYEE'S FULL NAME> ")
hoursWorked = float(input("<HOW MANY HOURS DID " + employeeName + " WORK?> "))
payRate = float(input(employeeName + " WHAT IS YOUR HOURLY RATE? "))
```



```
<ENTER EMPLOYEE'S FULL NAME> Raymond Garcia
<HOW MANY HOURS DID Raymond Garcia WORK?> 40
Raymond Garcia WHAT IS YOUR HOURLY RATE? 35
================================================================
PAYROLL INFORMATION FOR: Raymond Garcia
================================================================
   Hours Worked =        40.00
    Rate of Pay = $      35.00
================================================================
      Gross Pay = $   1,400.00
================================================================
>>> 
 RESTART: G:\SPRING 2020\COP1000\FLOWGORITHM PROGRAMS\Lastname_firstname_GROSS_P
AY_IF_STATEMENT.py
<ENTER EMPLOYEE'S FULL NAME> Gregory Foster
<HOW MANY HOURS DID Gregory Foster WORK?> 55
Gregory Foster WHAT IS YOUR HOURLY RATE? 42.50
================================================================
PAYROLL INFORMATION FOR: Gregory Foster
================================================================
   Hours Worked =        55.00
    Rate of Pay = $      42.50
================================================================
OVER TIME HOURS =        15.00
  OVER TIME PAY = $     956.25
    REGULAR PAY = $   1,700.00
      Gross Pay = $   2,656.25
================================================================
>>> 
```

FINAL RESULTS OF THE ENTIRE PYTHON PROGRAM WILL CLOSELY RESEMBLE THE FOLLOWING:



```
def toFixed(value, digits):
    return "%.*f" % (digits, value)

# PROGRAMMER:  your first and last name
# PROGRAM NAME: GROSS PAY SAMPLE USING IF STATEMENTS
# DATE WRITTEN: 1/3/2020
# PURPOSE: CALCULATE GROSS PAY BASED ON HOURS WORKED
# Declare all variables
# Initialize the processed / variables storing calculations
grossPay = 0.0
overTimeHours = 0.0
overTimePay = 0.0
regularPay = 0.0

# INPUT OPERATIONS
employeeName = input("<ENTER EMPLOYEE'S FULL NAME> ")
hoursWorked = float(input("<HOW MANY HOURS DID " + employeeName + " WORK?> "))
payRate = float(input(employeeName + " WHAT IS YOUR HOURLY RATE? "))

# Preliminary Output Statements
# Output Operations
print("===============================================================")
print("PAYROLL INFORMATION FOR:  " + employeeName)
print("===============================================================")
print("   Hours Worked =    " + format(hoursWorked,"10,.2f"))
print("    Rate of Pay = $  " + format(payRate, "10,.2f"))
print("===============================================================")

# Use IF statement to determine Gross Pay
if hoursWorked <= 40:
    grossPay = hoursWorked * payRate
    print("      Gross Pay = $  " + format(grossPay,"10,.2f"))
else:
    overTimeHours = hoursWorked - 40
    overTimePay = overTimeHours * (payRate * 1.5)
    regularPay = 40 * payRate
    grossPay = regularPay + overTimePay
    print("OVER TIME HOURS =    " + format(overTimeHours, "10,.2f"))
    print("  OVER TIME PAY = $  " + format(overTimePay, "10,.2f"))
    print("    REGULAR PAY = $  " + format(regularPay, "10,.2f"))
    print("      Gross Pay = $  " + format(grossPay,"10,.2f"))
print("===============================================================")

# END PROGRAM
```

Do not forget to resave your final version using the same name:

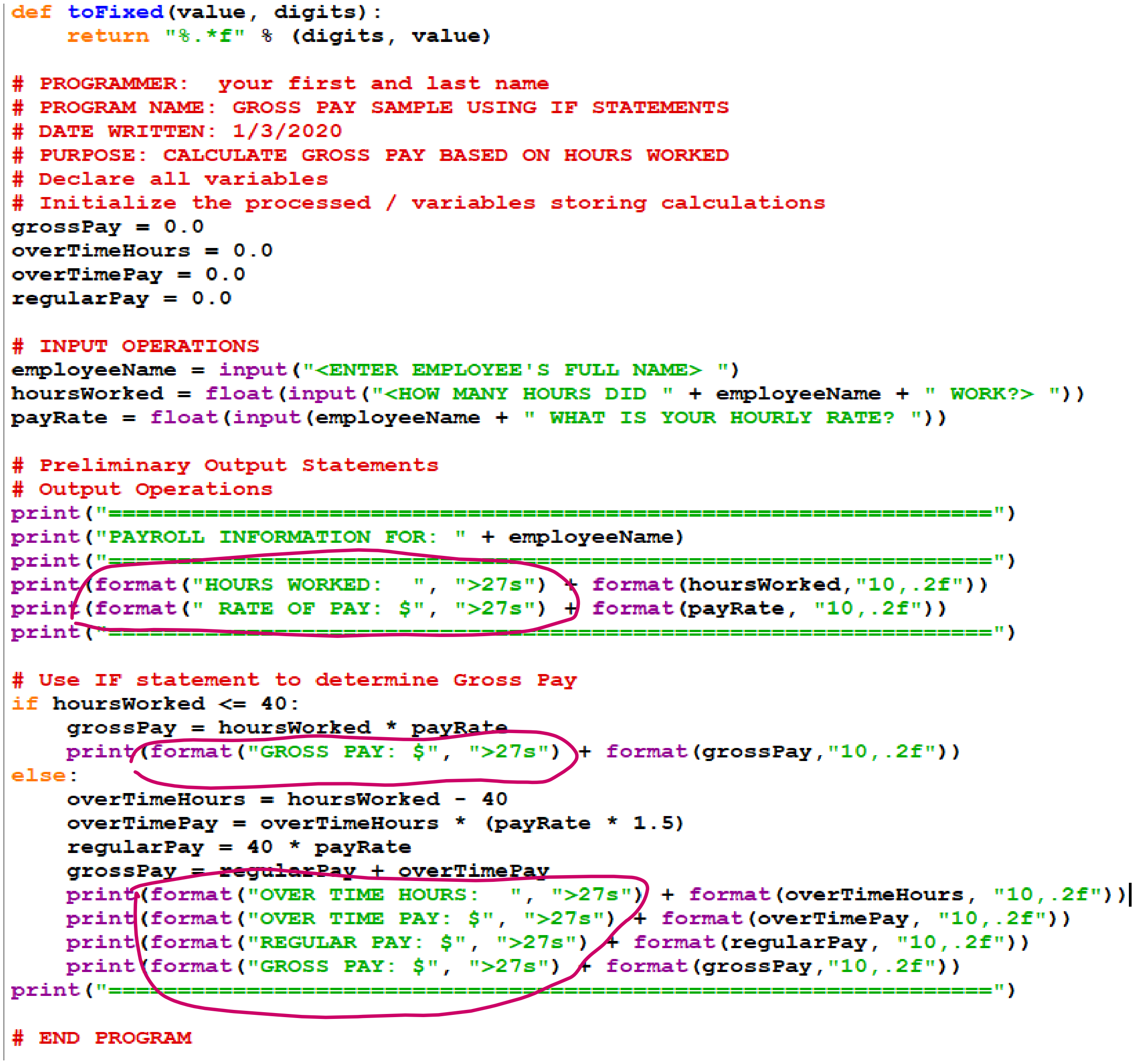
Lastname_firstname_A3_GROSS_PAY_IF_STATEMENT.py

FORMATTING LABELS

The format function can be used on labels or string data type.

1. In the format specifier the “s” for string will be used as the data type rather than “f” or float.
2. Additionally, there are special symbols to align the labels:
    a. > is used to right align the labels
    b. < is used to left align the labels; this is the default setting
    c. ^ or the caret symbol will center the labels.

ILLUSTRATION FOLLOWS:



```
def toFixed(value, digits):
    return "%.*f" % (digits, value)

# PROGRAMMER:  your first and last name
# PROGRAM NAME: GROSS PAY SAMPLE USING IF STATEMENTS
# DATE WRITTEN: 1/3/2020
# PURPOSE: CALCULATE GROSS PAY BASED ON HOURS WORKED
# Declare all variables
# Initialize the processed / variables storing calculations
grossPay = 0.0
overTimeHours = 0.0
overTimePay = 0.0
regularPay = 0.0

# INPUT OPERATIONS
employeeName = input("<ENTER EMPLOYEE'S FULL NAME> ")
hoursWorked = float(input("<HOW MANY HOURS DID " + employeeName + " WORK?> "))
payRate = float(input(employeeName + " WHAT IS YOUR HOURLY RATE? "))

# Preliminary Output Statements
# Output Operations
print("===============================================================")
print("PAYROLL INFORMATION FOR: " + employeeName)
print("===============================================================")
print(format("HOURS WORKED:  ", ">27s") + format(hoursWorked,"10,.2f"))
print(format(" RATE OF PAY: $", ">27s") + format(payRate, "10,.2f"))
print("===============================================================")

# Use IF statement to determine Gross Pay
if hoursWorked <= 40:
    grossPay = hoursWorked * payRate
    print(format("GROSS PAY: $", ">27s") + format(grossPay,"10,.2f"))
else:
    overTimeHours = hoursWorked - 40
    overTimePay = overTimeHours * (payRate * 1.5)
    regularPay = 40 * payRate
    grossPay = regularPay + overTimePay
    print(format("OVER TIME HOURS:  ", ">27s") + format(overTimeHours, "10,.2f"))
    print(format("OVER TIME PAY: $", ">27s") + format(overTimePay, "10,.2f"))
    print(format("REGULAR PAY: $", ">27s") + format(regularPay, "10,.2f"))
    print(format("GROSS PAY: $", ">27s") + format(grossPay,"10,.2f"))
print("===============================================================")

# END PROGRAM
```

SHORTENING THE PRINT STATEMENT USED TO DISPLAY DOUBLE LINES:
i.e.

```
print("=================================================================")
```

change to:

```
print("=" * 65)
```

```
def toFixed(value, digits):
    return "%.*f" % (digits, value)

# PROGRAMMER:  your first and last name
# PROGRAM NAME: GROSS PAY SAMPLE USING IF STATEMENTS
# DATE WRITTEN: 1/3/2020
# PURPOSE: CALCULATE GROSS PAY BASED ON HOURS WORKED
# Declare all variables
# Initialize the processed / variables storing calculations
grossPay = 0.0
overTimeHours = 0.0
overTimePay = 0.0
regularPay = 0.0

# INPUT OPERATIONS
employeeName = input("<ENTER EMPLOYEE'S FULL NAME> ")
hoursWorked = float(input("<HOW MANY HOURS DID " + employeeName + " WORK?> "))
payRate = float(input(employeeName + " WHAT IS YOUR HOURLY RATE? "))

# Preliminary Output Statements
# Output Operations
print("=" * 65)
print("PAYROLL INFORMATION FOR:  " + employeeName)
print("=" * 65)
print(format("HOURS WORKED:   ", ">27s") + format(hoursWorked,"10,.2f"))
print(format(" RATE OF PAY: $", ">27s") + format(payRate, "10,.2f"))
print("=" * 65)

# Use IF statement to determine Gross Pay
if hoursWorked <= 40:
    grossPay = hoursWorked * payRate
    print(format("GROSS PAY: $", ">27s") + format(grossPay,"10,.2f"))
else:
    overTimeHours = hoursWorked - 40
    overTimePay = overTimeHours * (payRate * 1.5)
    regularPay = 40 * payRate
    grossPay = regularPay + overTimePay
    print(format("OVER TIME HOURS:   ", ">27s") + format(overTimeHours, "10,.2f"))
    print(format("OVER TIME PAY: $", ">27s") + format(overTimePay, "10,.2f"))
    print(format("REGULAR PAY: $", ">27s") + format(regularPay, "10,.2f"))
    print(format("GROSS PAY: $", ">27s") + format(grossPay,"10,.2f"))
print("=" * 65)

# END PROGRAM
```

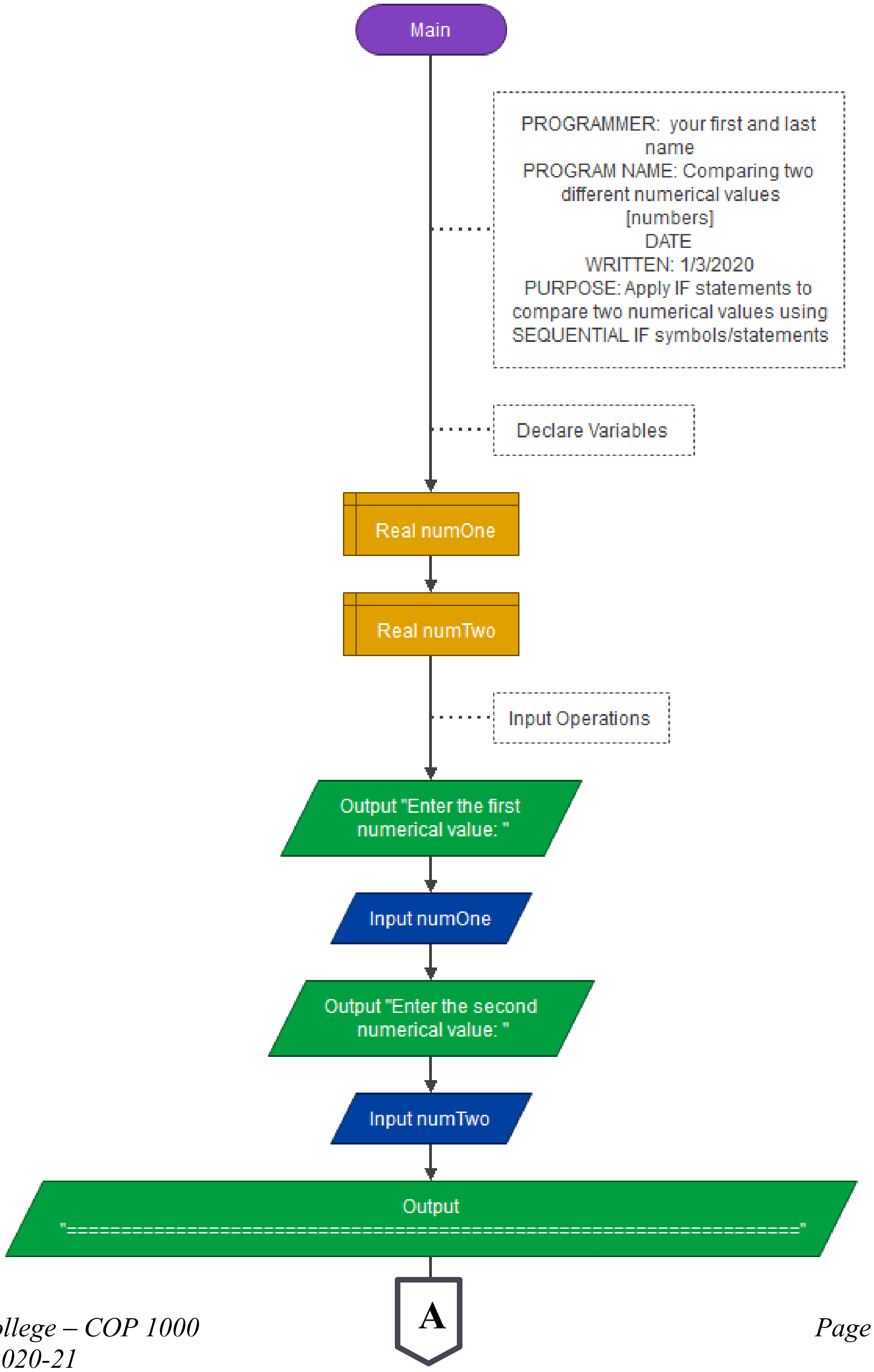
PART 3 – DETERMINING THE LARGEST OF TWO VALUES

PURPOSE: A decision statement / IF statement will be used to decide if two values are equal, which one is smaller or larger.

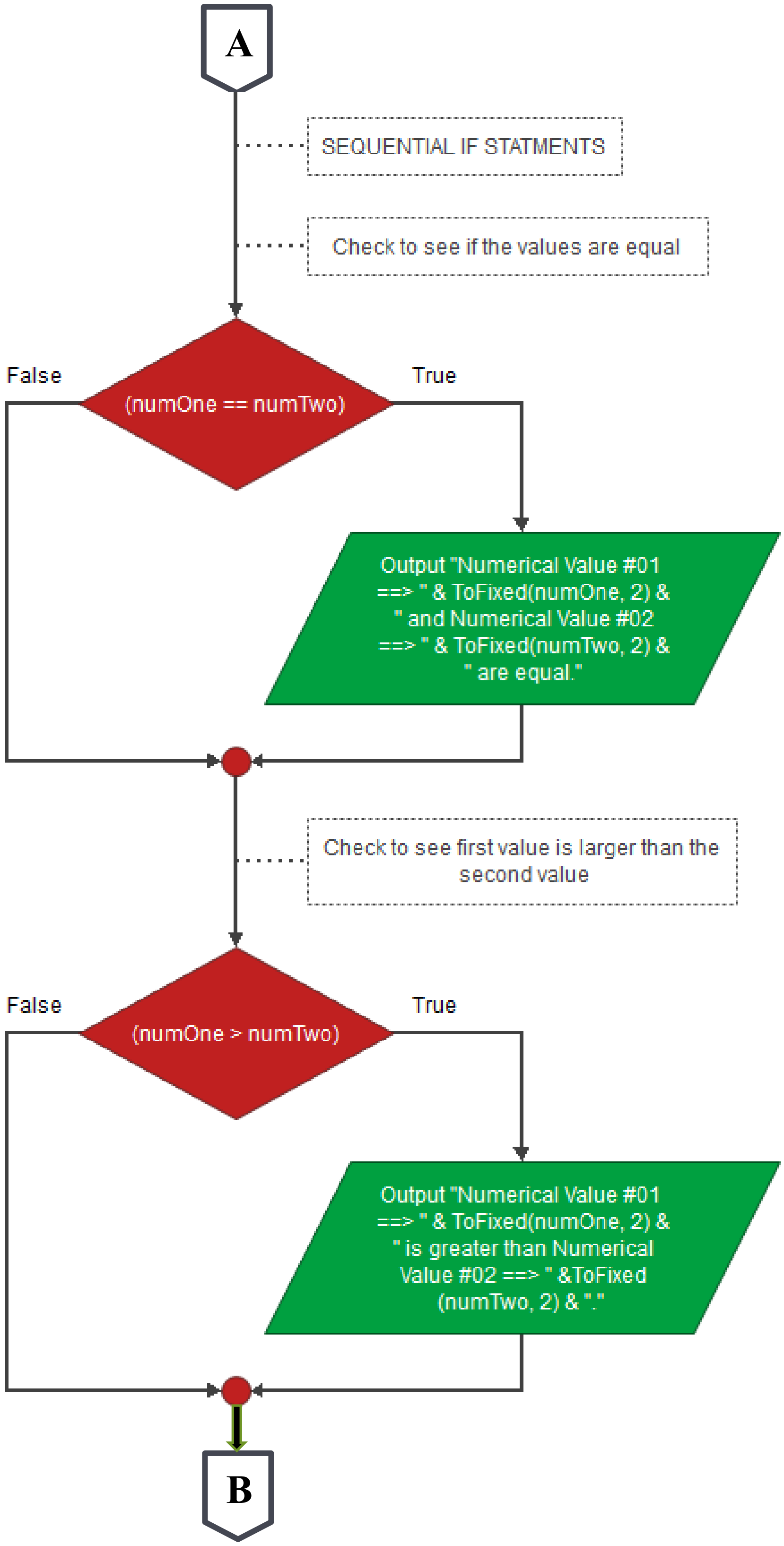
STUDY THE FOLLOWING EXAMPLES:

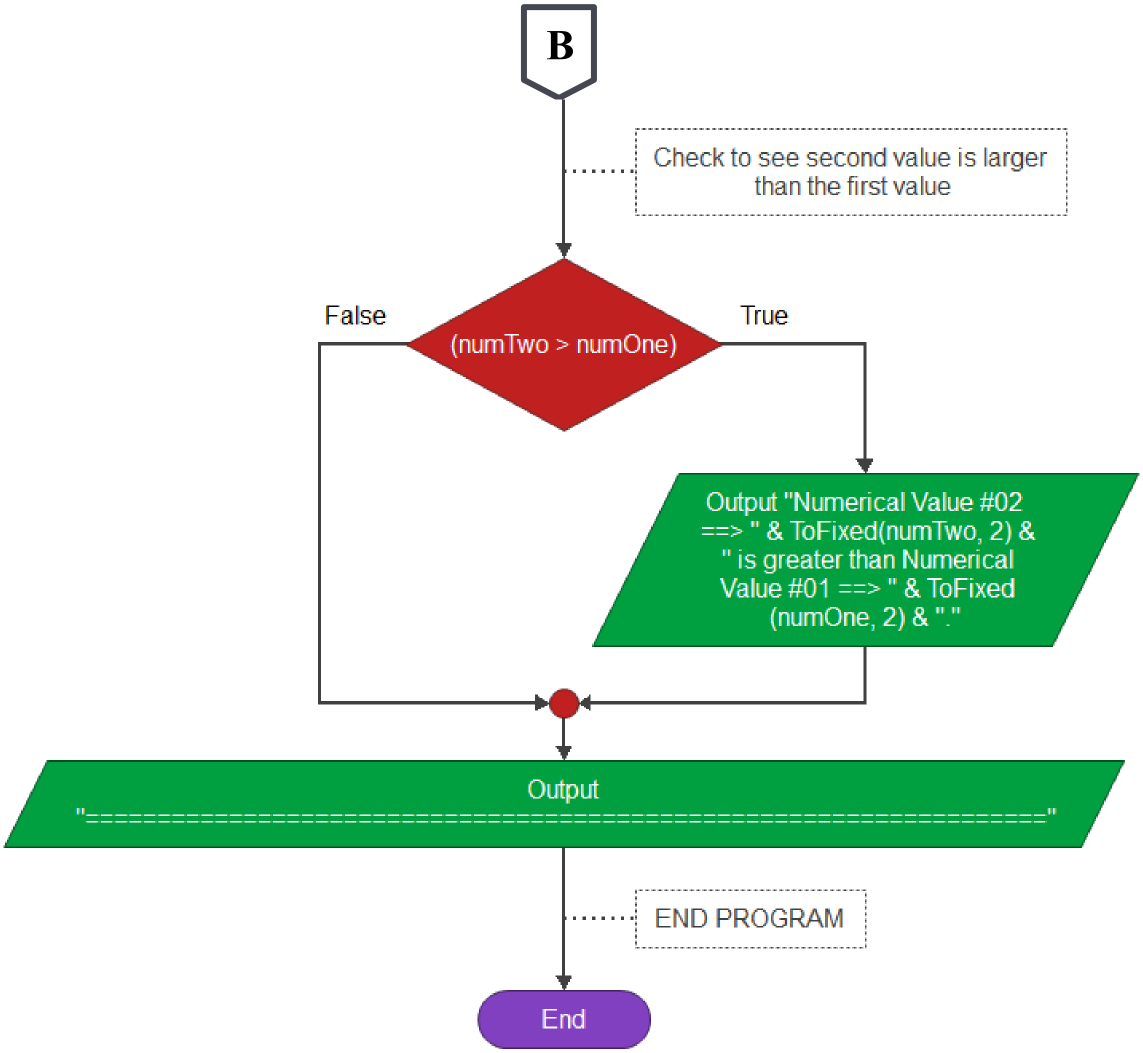
FLOWGORITHM FLOWCHART:

USING A SEQUENTIAL IF SYMBOL / STATEMENT

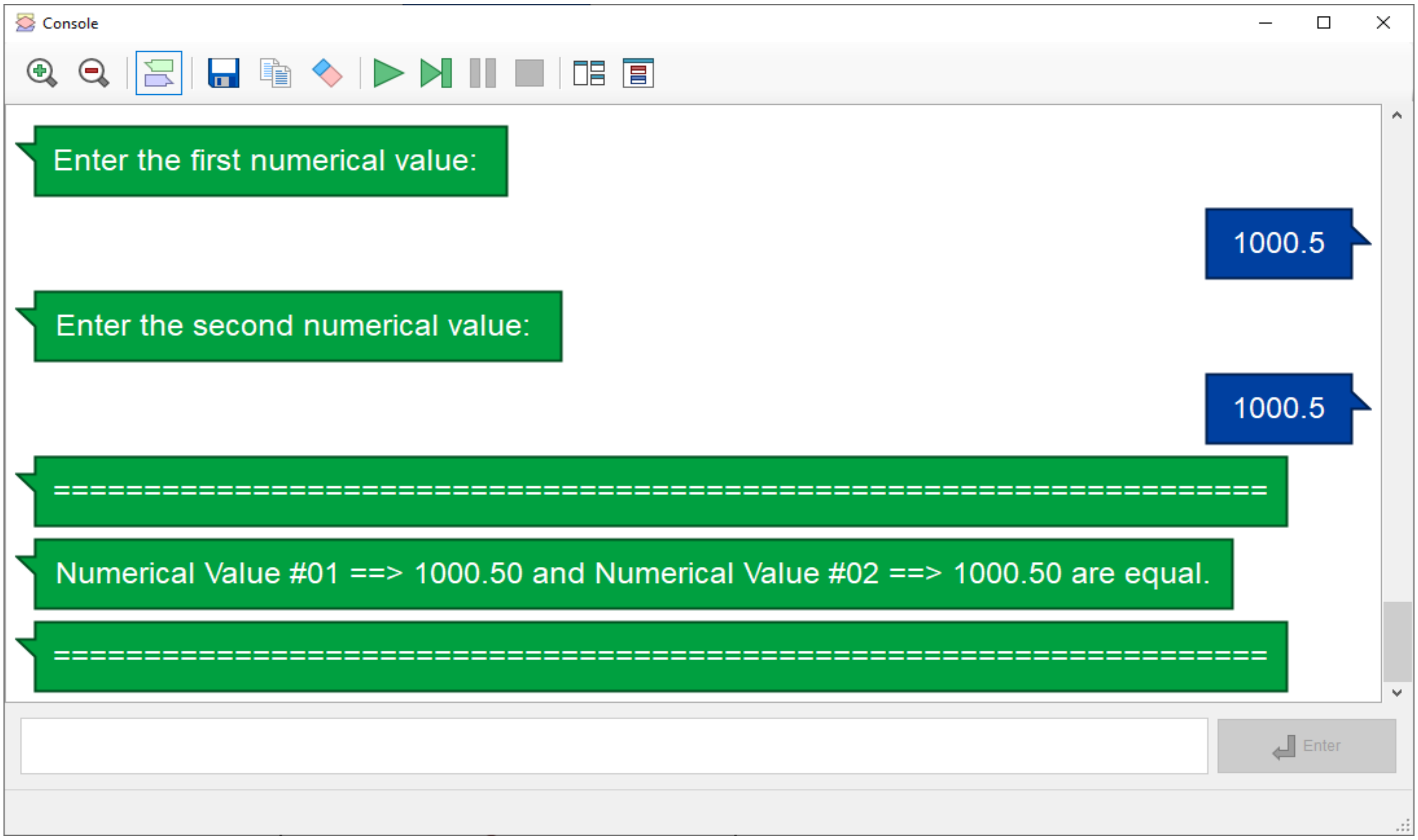

Main
PROGRAMMER: your first and last name
PROGRAM NAME: Comparing two different numerical values [numbers]
DATE WRITTEN: 1/3/2020
PURPOSE: Apply IF statements to compare two numerical values using SEQUENTIAL IF symbols/statements
Declare Variables
Real numOne
Real numTwo
Input Operations
Output "Enter the first numerical value: "
Input numOne
Output "Enter the second numerical value: "
Input numTwo
Output "=="
A

USING THE SEQUENTIAL APPROACH:

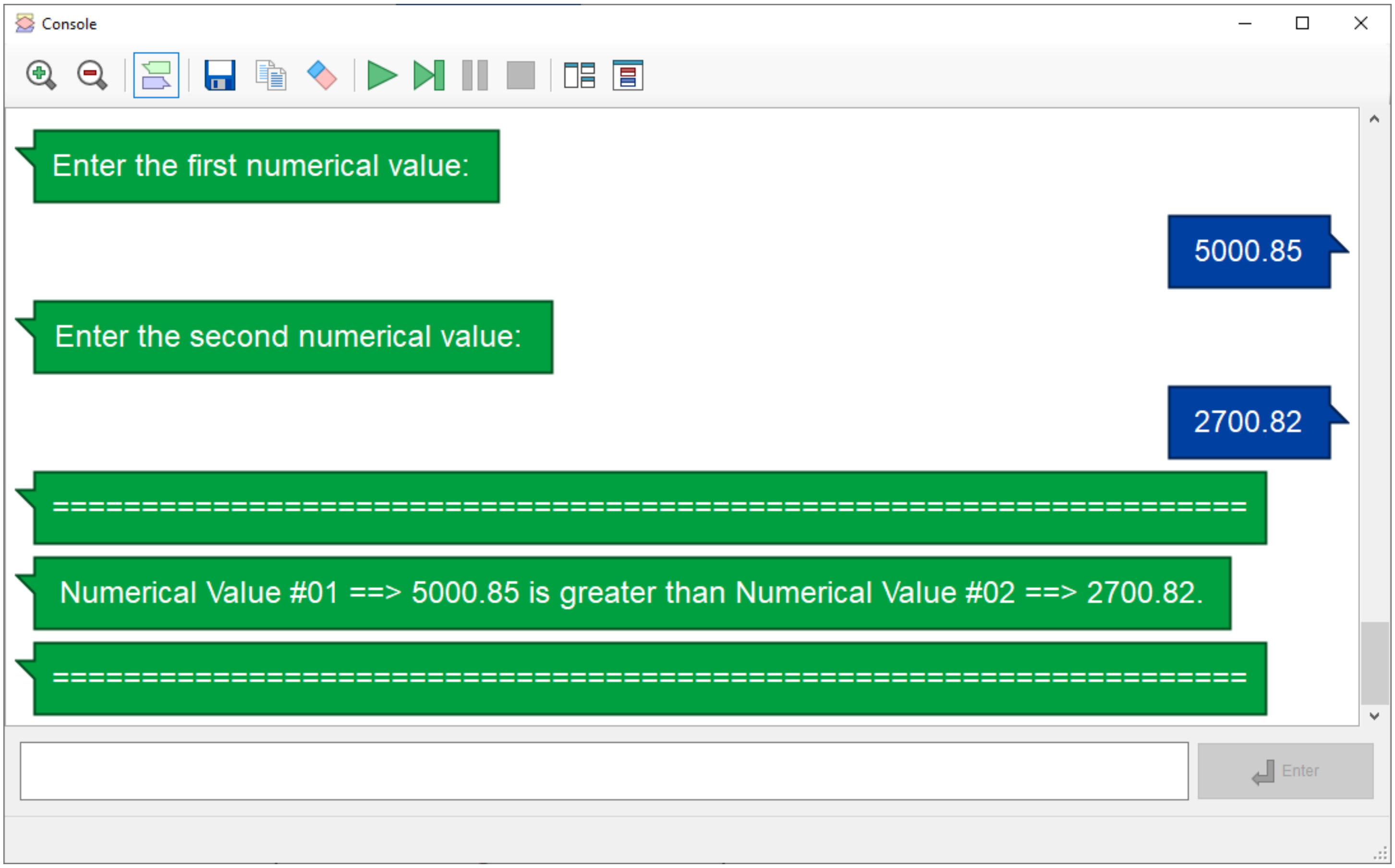

A
SEQUENTIAL IF STATMENTS
Check to see if the values are equal
False
(numOne == numTwo)
True
Output "Numerical Value #01 ==> " & ToFixed(numOne, 2) & " and Numerical Value #02 ==> " & ToFixed(numTwo, 2) & " are equal."
Check to see first value is larger than the second value
False
(numOne > numTwo)
True
Output "Numerical Value #01 ==> " & ToFixed(numOne, 2) & " is greater than Numerical Value #02 ==> " &ToFixed (numTwo, 2) & "."
B


B
Check to see second value is larger than the first value
False
(numTwo > numOne)
True
Output "Numerical Value #02 ==> " & ToFixed(numTwo, 2) & " is greater than Numerical Value #01 ==> " & ToFixed (numOne, 2) & "."
Output "=="
END PROGRAM
End

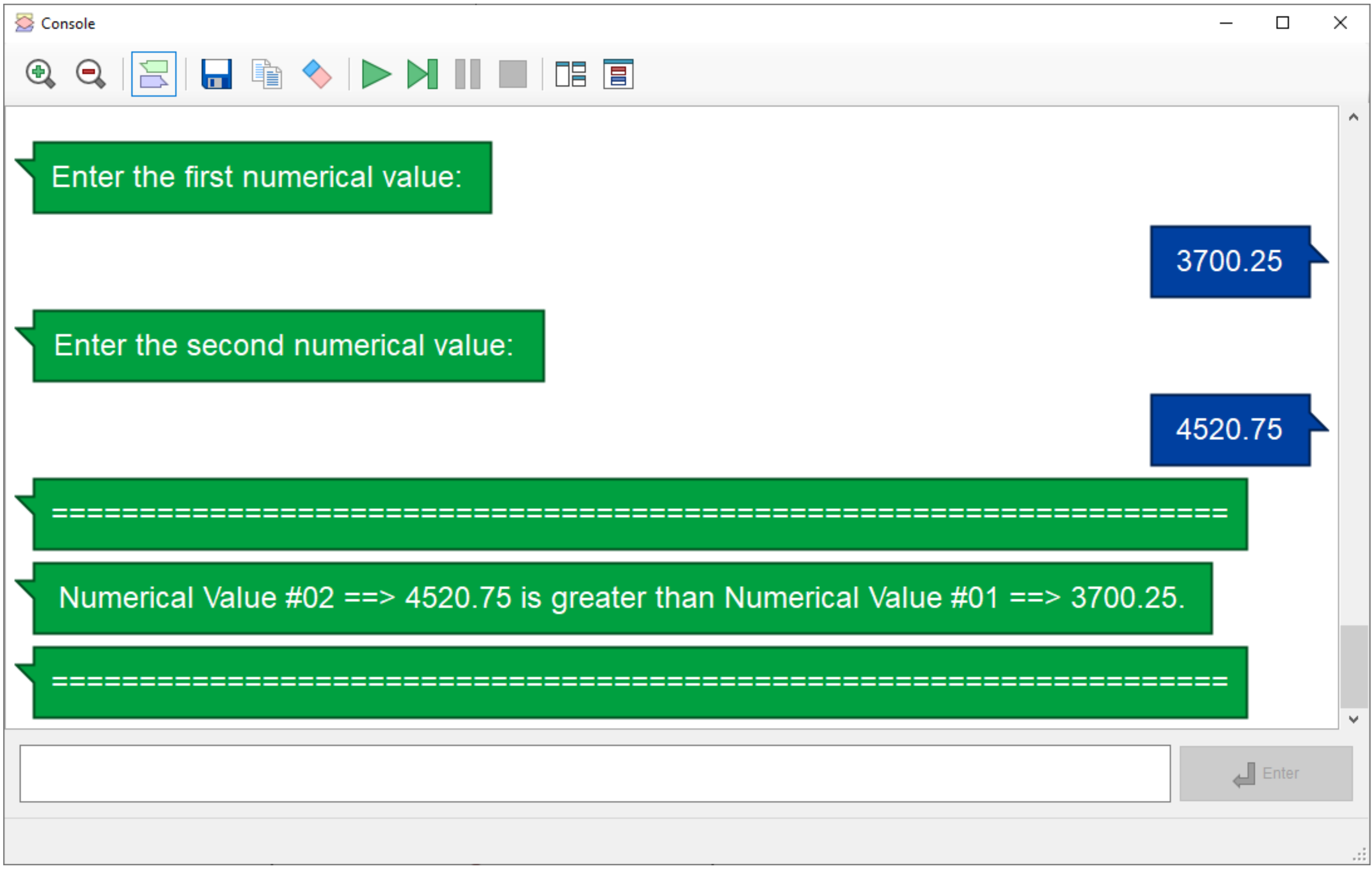
SAMPLE: EXECUTIONS / RUNS



```
Enter the first numerical value:
                                                        1000.5
Enter the second numerical value:
                                                        1000.5
========================================================================
Numerical Value #01 ==> 1000.50 and Numerical Value #02 ==> 1000.50 are equal.
========================================================================
```

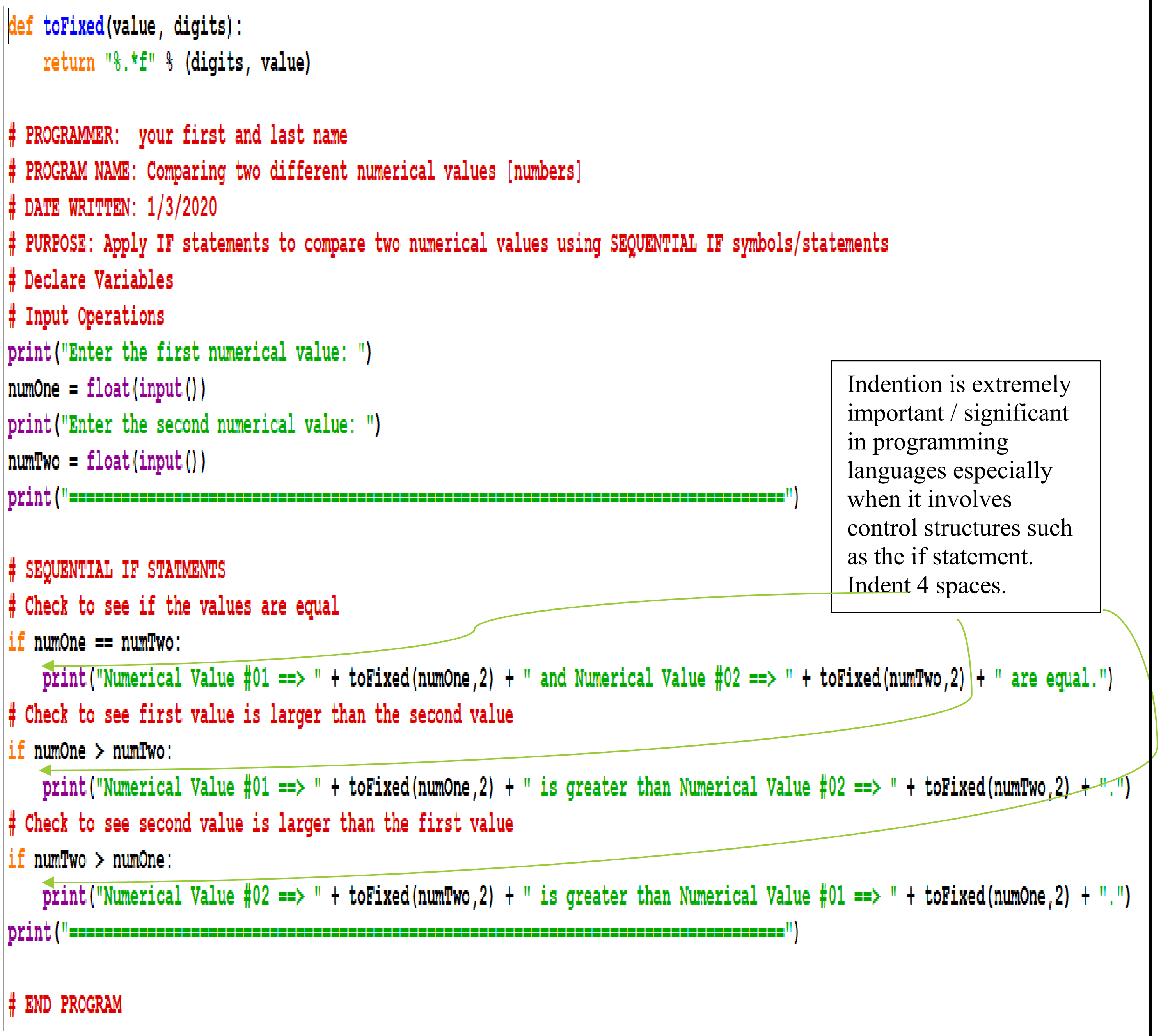


```
Enter the first numerical value:
                                                        5000.85
Enter the second numerical value:
                                                        2700.82
========================================================================
Numerical Value #01 ==> 5000.85 is greater than Numerical Value #02 ==> 2700.82.
========================================================================
```



```
Enter the first numerical value:
3700.25
Enter the second numerical value:
4520.75
================================================================================
Numerical Value #02 ==> 4520.75 is greater than Numerical Value #01 ==> 3700.25.
================================================================================
```

CONVERT TO PYTHON:

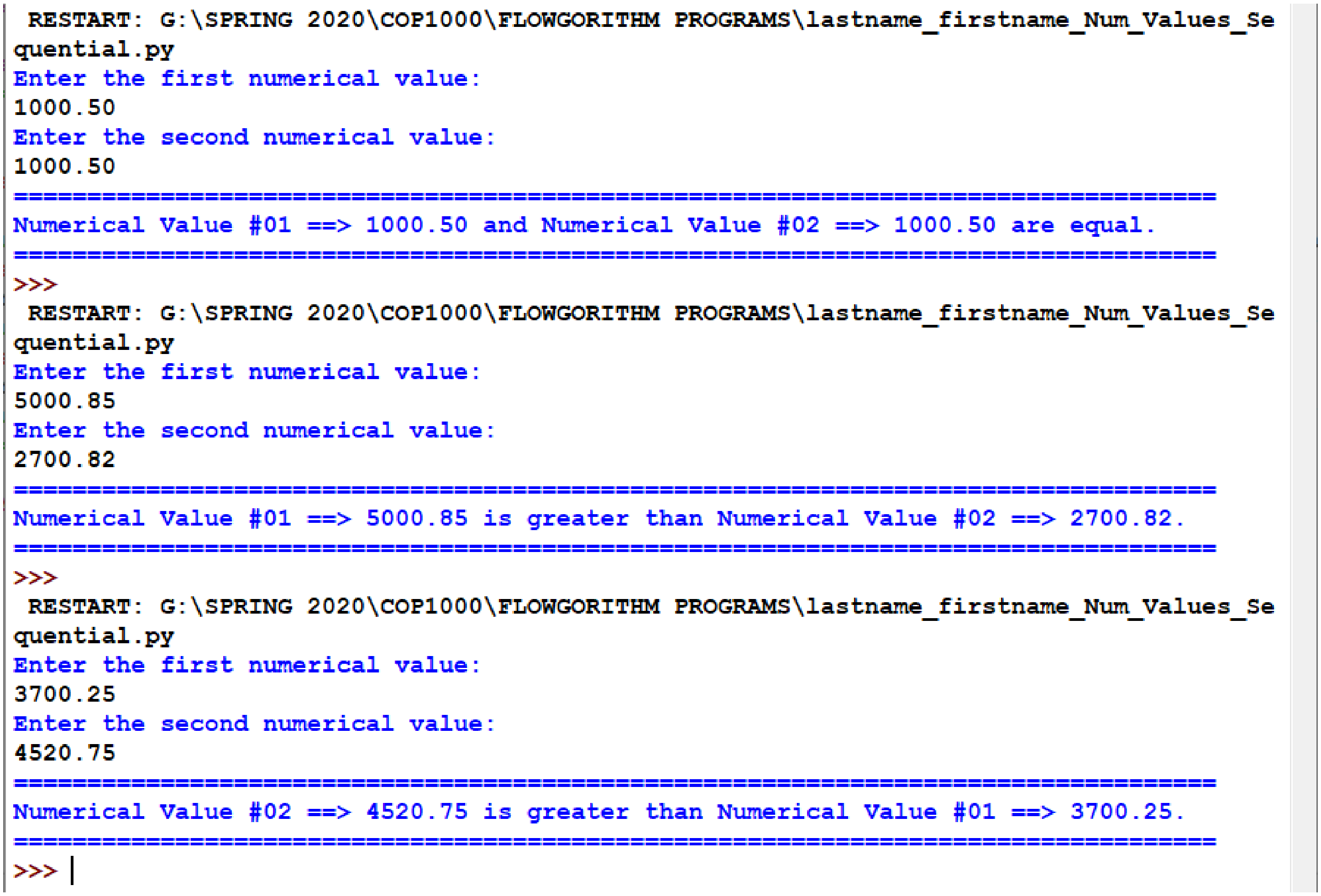


```python
def toFixed(value, digits):
    return "%.*f" % (digits, value)

# PROGRAMMER:  your first and last name
# PROGRAM NAME: Comparing two different numerical values [numbers]
# DATE WRITTEN: 1/3/2020
# PURPOSE: Apply IF statements to compare two numerical values using SEQUENTIAL IF symbols/statements
# Declare Variables
# Input Operations
print("Enter the first numerical value: ")
numOne = float(input())
print("Enter the second numerical value: ")
numTwo = float(input())
print("==================================================================================")

# SEQUENTIAL IF STATMENTS
# Check to see if the values are equal
if numOne == numTwo:
    print("Numerical Value #01 ==> " + toFixed(numOne,2) + " and Numerical Value #02 ==> " + toFixed(numTwo,2) + " are equal.")
# Check to see first value is larger than the second value
if numOne > numTwo:
    print("Numerical Value #01 ==> " + toFixed(numOne,2) + " is greater than Numerical Value #02 ==> " + toFixed(numTwo,2) + ".")
# Check to see second value is larger than the first value
if numTwo > numOne:
    print("Numerical Value #02 ==> " + toFixed(numTwo,2) + " is greater than Numerical Value #01 ==> " + toFixed(numOne,2) + ".")
print("==================================================================================")

# END PROGRAM
```

Indention is extremely important / significant in programming languages especially when it involves control structures such as the if statement. Indent 4 spaces.

SAMPLE RUNS / EXECUTIONS FOLLOW:

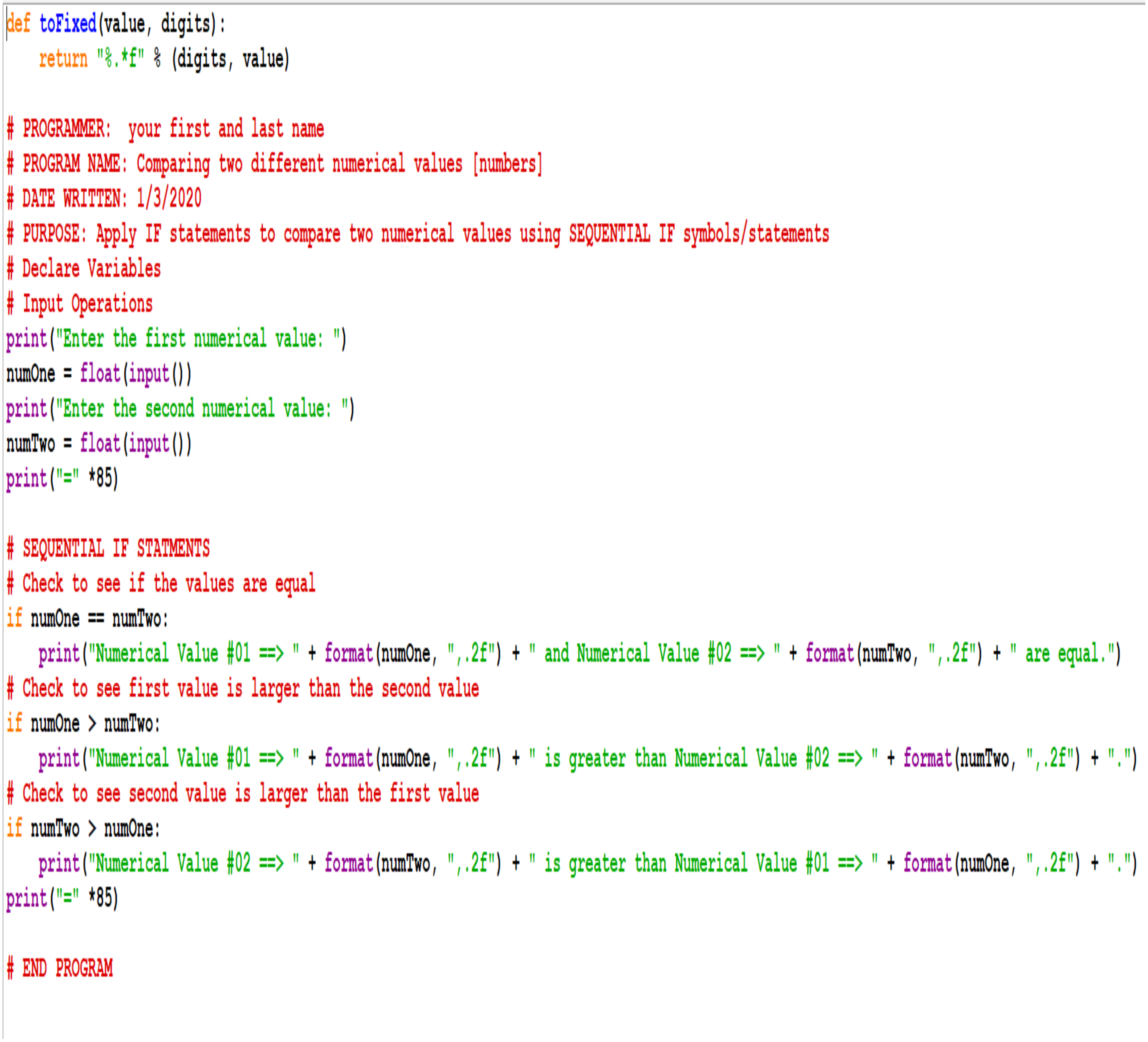
SAMPLE RUNS / EXECUTIONS:



```
 RESTART: G:\SPRING 2020\COP1000\FLOWGORITHM PROGRAMS\lastname_firstname_Num_Values_Se
quential.py
Enter the first numerical value:
1000.50
Enter the second numerical value:
1000.50
==================================================================================
Numerical Value #01 ==> 1000.50 and Numerical Value #02 ==> 1000.50 are equal.
==================================================================================
>>>
 RESTART: G:\SPRING 2020\COP1000\FLOWGORITHM PROGRAMS\lastname_firstname_Num_Values_Se
quential.py
Enter the first numerical value:
5000.85
Enter the second numerical value:
2700.82
==================================================================================
Numerical Value #01 ==> 5000.85 is greater than Numerical Value #02 ==> 2700.82.
==================================================================================
>>>
 RESTART: G:\SPRING 2020\COP1000\FLOWGORITHM PROGRAMS\lastname_firstname_Num_Values_Se
quential.py
Enter the first numerical value:
3700.25
Enter the second numerical value:
4520.75
==================================================================================
Numerical Value #02 ==> 4520.75 is greater than Numerical Value #01 ==> 3700.25.
==================================================================================
>>> |
```

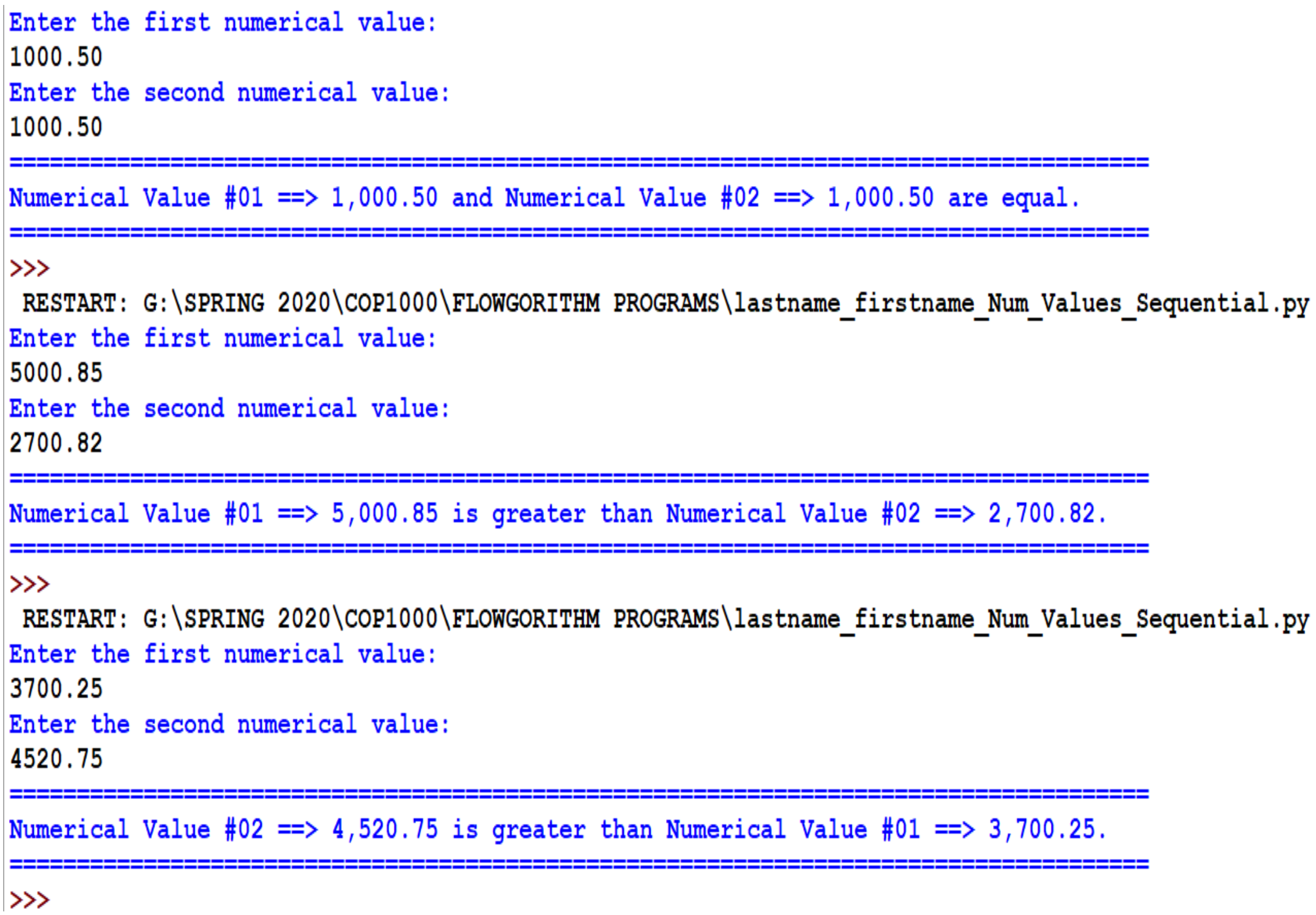
Change the ToFixed function in the print statements to format, and shorten the print statements used to display the lines as in the following illustrations:

You may also combine the print/input statement or leave as converted.



```
def toFixed(value, digits):
    return "%.*f" % (digits, value)

# PROGRAMMER:  your first and last name
# PROGRAM NAME: Comparing two different numerical values [numbers]
# DATE WRITTEN: 1/3/2020
# PURPOSE: Apply IF statements to compare two numerical values using SEQUENTIAL IF symbols/statements
# Declare Variables
# Input Operations
print("Enter the first numerical value: ")
numOne = float(input())
print("Enter the second numerical value: ")
numTwo = float(input())
print("=" *85)

# SEQUENTIAL IF STATMENTS
# Check to see if the values are equal
if numOne == numTwo:
    print("Numerical Value #01 ==> " + format(numOne, ",.2f") + " and Numerical Value #02 ==> " + format(numTwo, ",.2f") + " are equal.")
# Check to see first value is larger than the second value
if numOne > numTwo:
    print("Numerical Value #01 ==> " + format(numOne, ",.2f") + " is greater than Numerical Value #02 ==> " + format(numTwo, ",.2f") + ".")
# Check to see second value is larger than the first value
if numTwo > numOne:
    print("Numerical Value #02 ==> " + format(numTwo, ",.2f") + " is greater than Numerical Value #01 ==> " + format(numOne, ",.2f") + ".")
print("=" *85)

# END PROGRAM
```

SAMPLE RUNS OF THE REVISED PYTHON PROGRAM FOLLOW:

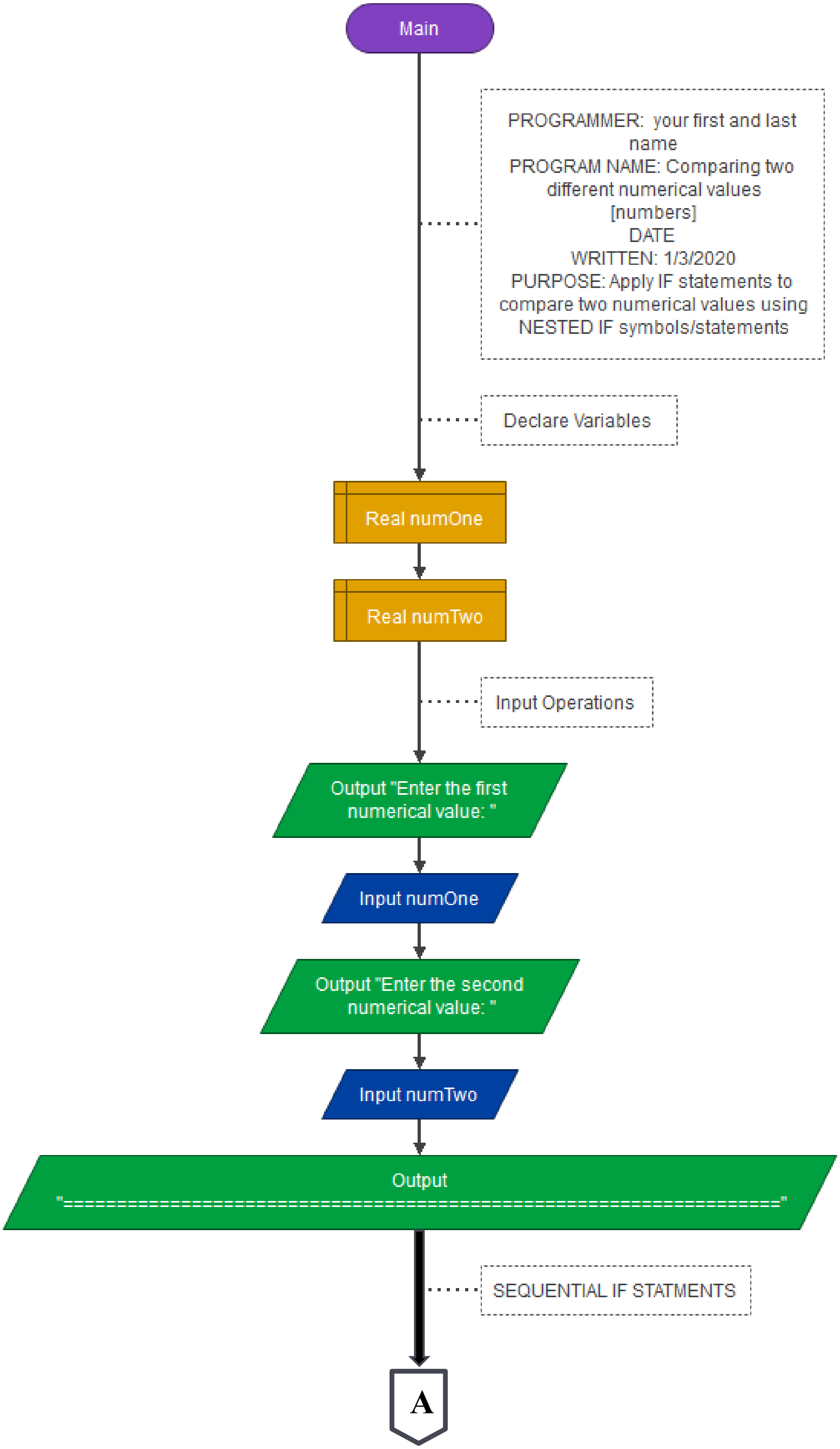


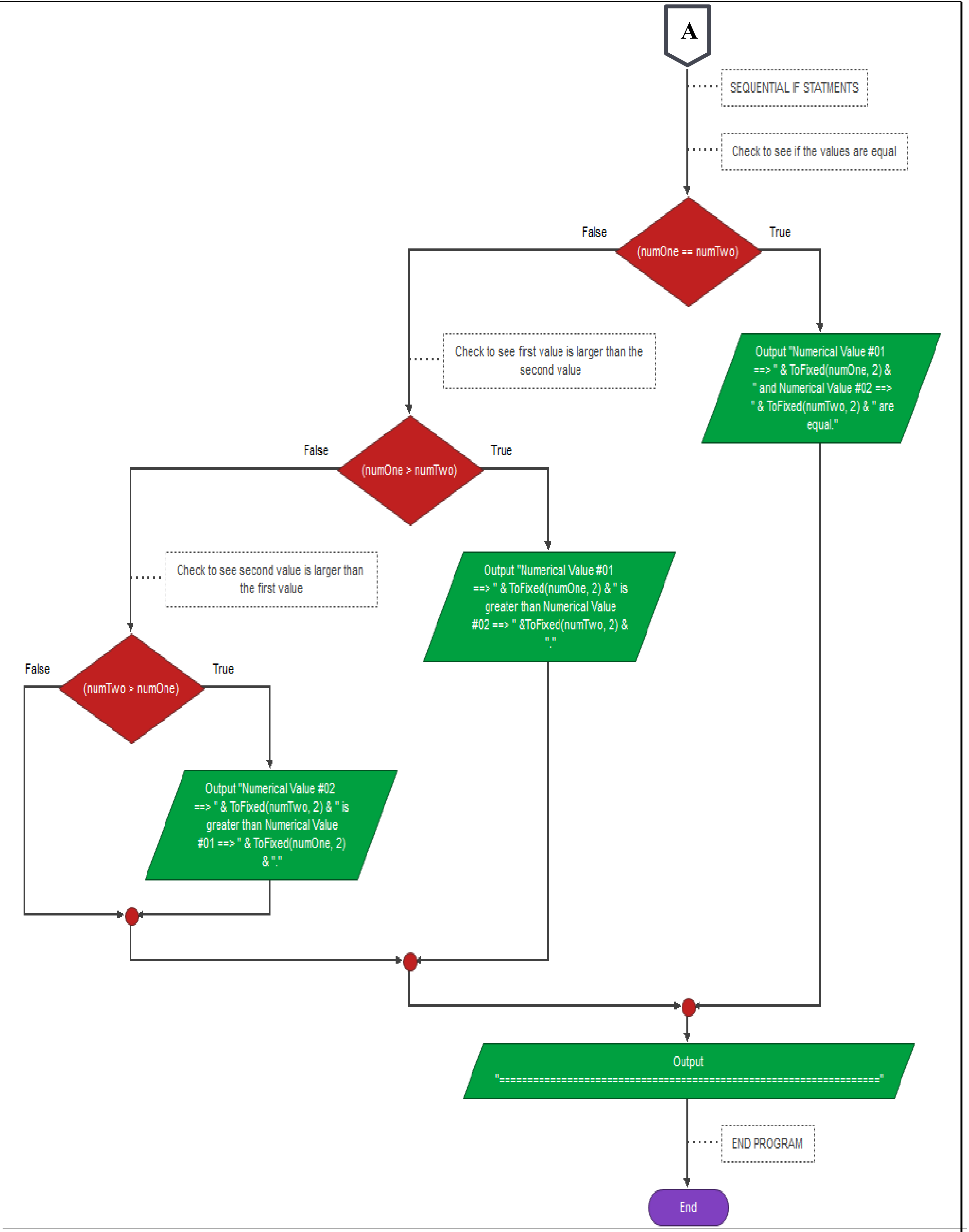
```
Enter the first numerical value:
1000.50
Enter the second numerical value:
1000.50
=====================================================================================
Numerical Value #01 ==> 1,000.50 and Numerical Value #02 ==> 1,000.50 are equal.
=====================================================================================
>>>
 RESTART: G:\SPRING 2020\COP1000\FLOWGORITHM PROGRAMS\lastname_firstname_Num_Values_Sequential.py
Enter the first numerical value:
5000.85
Enter the second numerical value:
2700.82
=====================================================================================
Numerical Value #01 ==> 5,000.85 is greater than Numerical Value #02 ==> 2,700.82.
=====================================================================================
>>>
 RESTART: G:\SPRING 2020\COP1000\FLOWGORITHM PROGRAMS\lastname_firstname_Num_Values_Sequential.py
Enter the first numerical value:
3700.25
Enter the second numerical value:
4520.75
=====================================================================================
Numerical Value #02 ==> 4,520.75 is greater than Numerical Value #01 ==> 3,700.25.
=====================================================================================
>>>
```

SEQUENTIAL VERSION [Takes longer, some redundancy]

The program must test all conditions even if the condition was met by the first condition in the sequence. To mitigate redundancy, the flowgorithm program can be revised as a nested if statement as illustrated.

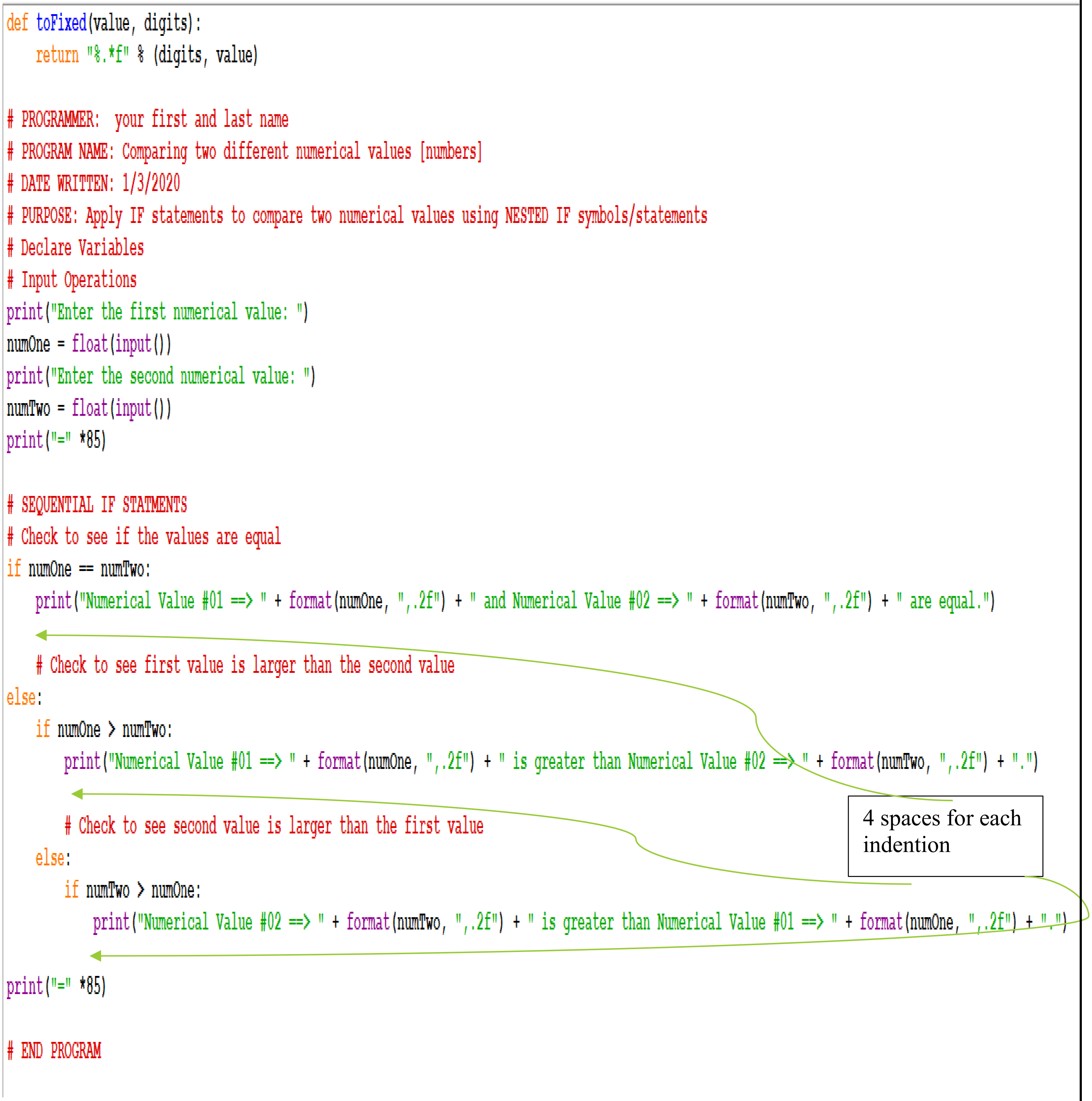
NESTED IF - FLOWGORITM PROGRAM


Main
PROGRAMMER: your first and last name
PROGRAM NAME: Comparing two different numerical values [numbers]
DATE
WRITTEN: 1/3/2020
PURPOSE: Apply IF statements to compare two numerical values using NESTED IF symbols/statements
Declare Variables
Real numOne
Real numTwo
Input Operations
Output "Enter the first numerical value: "
Input numOne
Output "Enter the second numerical value: "
Input numTwo
Output "=="
SEQUENTIAL IF STATMENTS
A


A
SEQUENTIAL IF STATMENTS
Check to see if the values are equal
False
(numOne == numTwo)
True
Output "Numerical Value #01 ==> " & ToFixed(numOne, 2) & " and Numerical Value #02 ==> " & ToFixed(numTwo, 2) & " are equal."
Check to see first value is larger than the second value
False
(numOne > numTwo)
True
Output "Numerical Value #01 ==> " & ToFixed(numOne, 2) & " is greater than Numerical Value #02 ==> " &ToFixed(numTwo, 2) & "."
Check to see second value is larger than the first value
False
(numTwo > numOne)
True
Output "Numerical Value #02 ==> " & ToFixed(numTwo, 2) & " is greater than Numerical Value #01 ==> " & ToFixed(numOne, 2) & "."
Output "=="
END PROGRAM
End

TEST THE FLOWGORITHM PROGRAM TO MAKE SURE IT WORKS CORRECTLY.

PYTHON VERSION OF NESTED IF STATEMENTS:

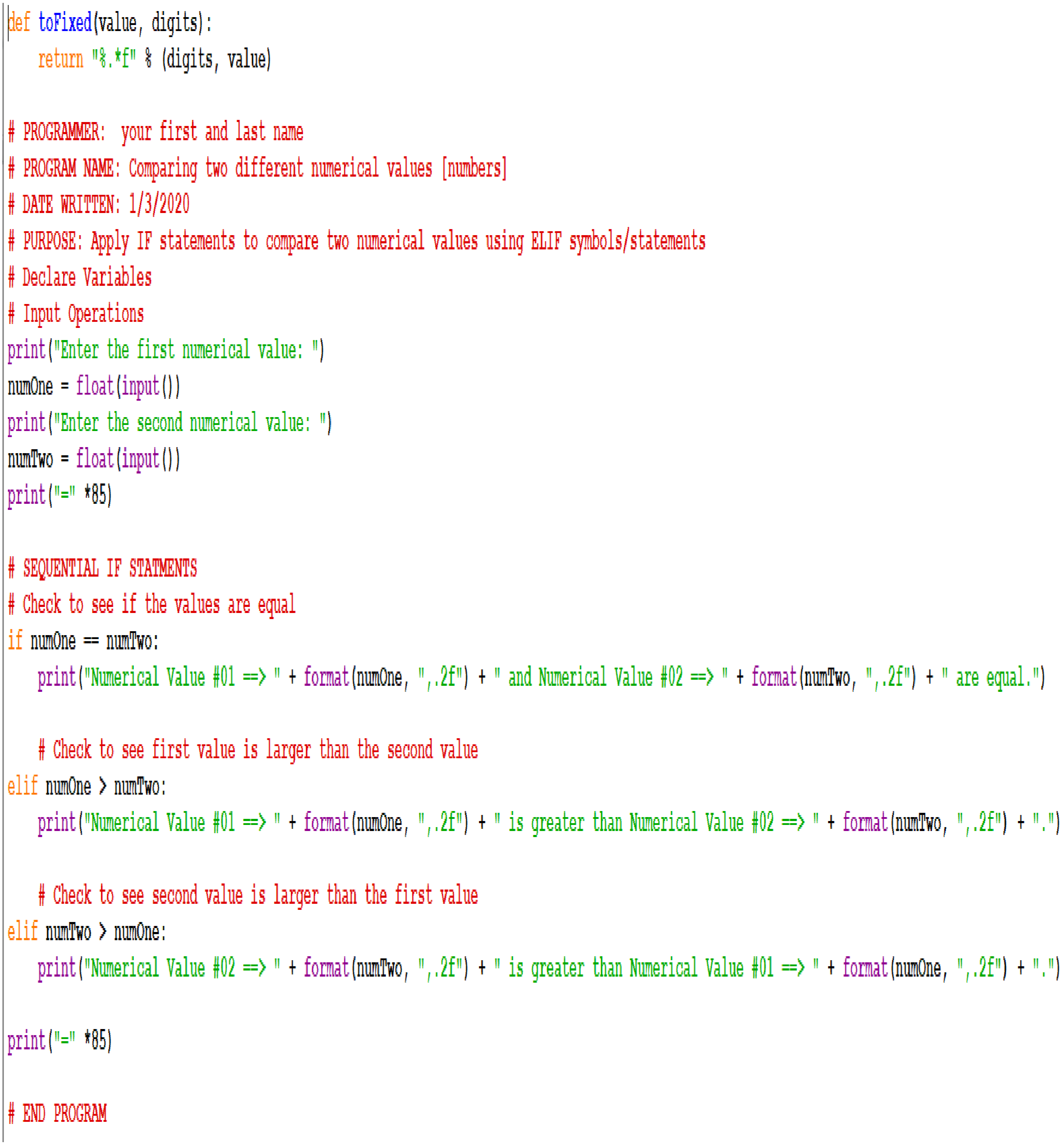


```
def toFixed(value, digits):
    return "%.*f" % (digits, value)

# PROGRAMMER:  your first and last name
# PROGRAM NAME: Comparing two different numerical values [numbers]
# DATE WRITTEN: 1/3/2020
# PURPOSE: Apply IF statements to compare two numerical values using NESTED IF symbols/statements
# Declare Variables
# Input Operations
print("Enter the first numerical value: ")
numOne = float(input())
print("Enter the second numerical value: ")
numTwo = float(input())
print("=" *85)

# SEQUENTIAL IF STATMENTS
# Check to see if the values are equal
if numOne == numTwo:
    print("Numerical Value #01 ==> " + format(numOne, ",.2f") + " and Numerical Value #02 ==> " + format(numTwo, ",.2f") + " are equal.")

    # Check to see first value is larger than the second value
else:
    if numOne > numTwo:
        print("Numerical Value #01 ==> " + format(numOne, ",.2f") + " is greater than Numerical Value #02 ==> " + format(numTwo, ",.2f") + ".")

        # Check to see second value is larger than the first value
    else:
        if numTwo > numOne:
            print("Numerical Value #02 ==> " + format(numTwo, ",.2f") + " is greater than Numerical Value #01 ==> " + format(numOne, ",.2f") + ".")

print("=" *85)

# END PROGRAM
```

4 spaces for each indention

RUN / EXECUTION OF NESTED ELSE IF VERSION TO MAKE SURE THE PROGRAM WORKS CORRECTLY.

USING THE ELIF VERSION: This statement is exclusive to Python code only. This is not an option in Flowgorithm.



```
def toFixed(value, digits):
    return "%.*f" % (digits, value)

# PROGRAMMER:  your first and last name
# PROGRAM NAME: Comparing two different numerical values [numbers]
# DATE WRITTEN: 1/3/2020
# PURPOSE: Apply IF statements to compare two numerical values using ELIF symbols/statements
# Declare Variables
# Input Operations
print("Enter the first numerical value: ")
numOne = float(input())
print("Enter the second numerical value: ")
numTwo = float(input())
print("=" *85)

# SEQUENTIAL IF STATMENTS
# Check to see if the values are equal
if numOne == numTwo:
    print("Numerical Value #01 ==> " + format(numOne, ",.2f") + " and Numerical Value #02 ==> " + format(numTwo, ",.2f") + " are equal.")

    # Check to see first value is larger than the second value
elif numOne > numTwo:
    print("Numerical Value #01 ==> " + format(numOne, ",.2f") + " is greater than Numerical Value #02 ==> " + format(numTwo, ",.2f") + ".")

    # Check to see second value is larger than the first value
elif numTwo > numOne:
    print("Numerical Value #02 ==> " + format(numTwo, ",.2f") + " is greater than Numerical Value #01 ==> " + format(numOne, ",.2f") + ".")

print("=" *85)

# END PROGRAM
```