# LAB 8 ASSIGNMENT

**Ch. 5 Repetition Structures**

Start: 02/21/2024 5:54 PM
Name: Imani Hollie

## EXCERCISE 2 – AREAS OF RECTANGLES

Write the Algorithm, Pseudocode, Flowchart, and Python Code for the following programming problem:

> **Scenario: Area of Rectangles**
>
> The area of a rectangle is the rectangle's length times its width. Design a program that asks for the length and width of two rectangles. The program should tell the user which rectangle has the greater area, or whether the areas are the same.

**Step 1: The Algorithm**

1. MODULE 1 – main()
    a. Get the measurements for two rectangles:
        i. Prompt for the length
        ii. Prompt for the width
    b. Call module 2
    c. Call module 3
2. MODULE 2 – recArea1()
    a. Calculate the areas for two rectangle:
        i. Multiply the length by the width for the total area.
    b. Display the measurements calculated:
        i. Display Length
        ii. Display Width
        iii. Display Rectangle Area
3. MODULE 3 – recArea2()
    a. Calculate the areas for two rectangle:
        i. Multiply the length by the width for the total area.
    b. Display the measurements calculated:
        i. Display Length
        ii. Display Width
        iii. Display Rectangle Area

## The Input, Processing, and Output

| Table 1-1 Calculating Areas of Rectangles (x/y) | | | |
|---|---|---|---|
| **INPUTS** | **Input Type** | **Value** | **Data Type** |
| Rectangle Length (recL) | Variable | (a) | Float |
| Rectangle Width (recW) | Variable | (b) | Float |
| **PROCEDURE** | $x = a * b$<br>$areaRec = recL + recW$ | | |
| **OUTPUTS** | **Output Type** | **Value** | **Data Type** |
| Rectangle Area (recArea) | Variable | (x) | Float |

The IPO for Table 1-1 is as follows:

1. The inputs for Table 1-1 are as follows:
   a. Rectangle Length (a)
   b. Rectangle Width (b)
2. The procedure for Table 1-1 are as follows:
   a. $x = a * b$
      $recArea = recL + recW$
3. The output for Table 1-1 are as follows:
   a. Rectangle Area (x)

| Table 1-2 Nested Decision Structure: Mixing Colors | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Mixing Red as priColor1** | | | **Mixing Yellow as priColor1** | | | **Mixing Blue as priColor1** | | |
| **priColor1**<br>**(a)** | **priColor2**<br>**(b)** | | **priColor1**<br>**(a)** | **priColor2**<br>**(b)** | | **priColor1**<br>**(a)** | **priColor2**<br>**(b)** | |
| RED | YELLOW | BLUE | YELLOW | RED | BLUE | BLUE | RED | YELLOW |
| Red | Yellow | Blue | Yellow | Red | Blue | Blue | Red | Yellow |
| red | yellow | blue | yellow | red | blue | blue | red | yellow |
| **mixSecCol**<br>**(x)** | Orange | Purple | **mixSecCol**<br>**(x)** | Orange | Green | **mixSecCol**<br>**(x)** | Purple | Green |

**Step 2: The Pseudocode**

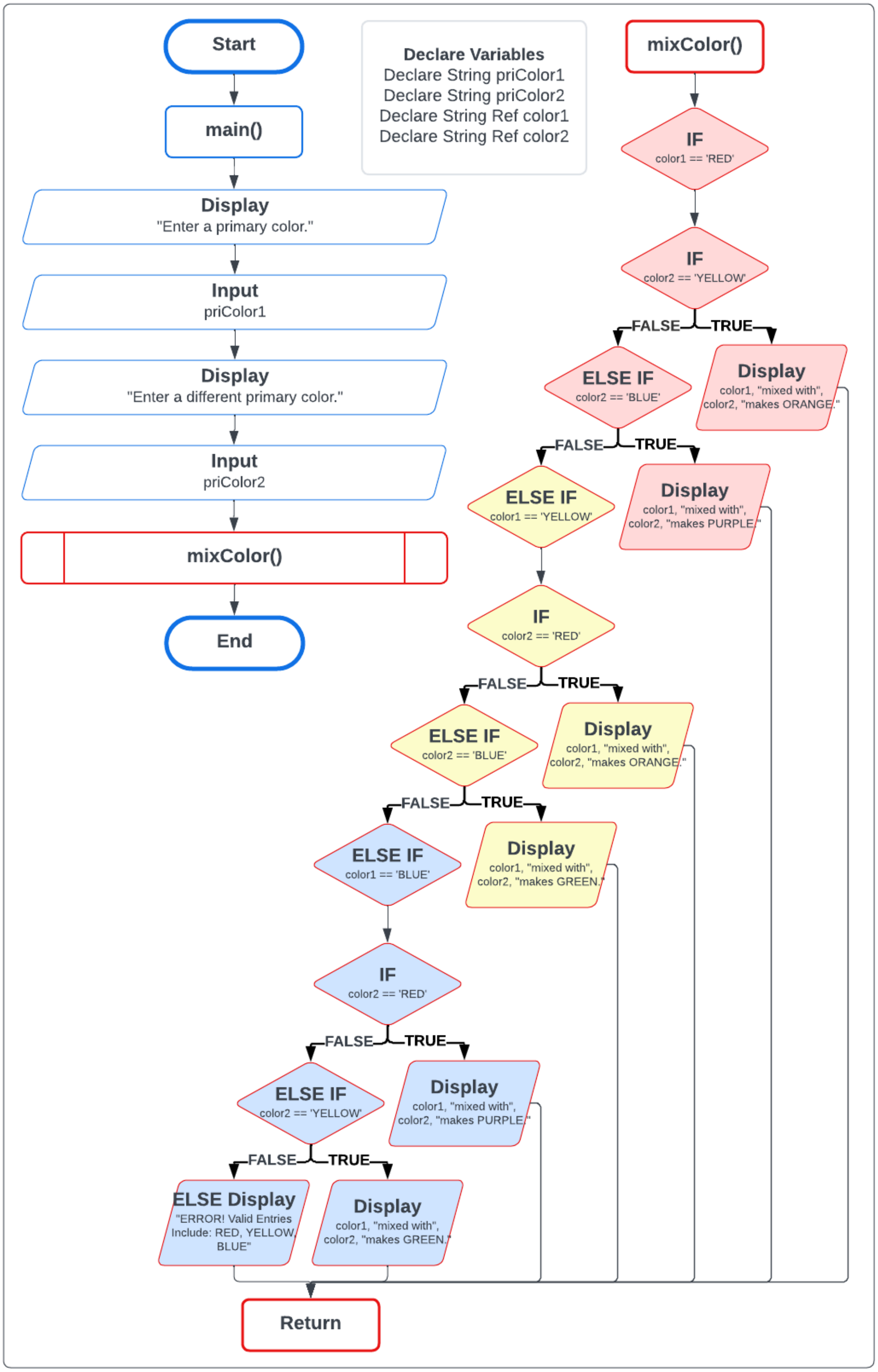
Refer to Tables 1-1 and 1-2 in Step 1 for the needed variables.

```
//This program takes in two primary colors.
//Output is then printed to the screen.

//Declare the main module
//main() input and calls mixColor()
Module main()

    //Declare variables
    Declare String priColor1
    Declare String priColor2

    //Input priColor1 and priColor2
    Display “Enter a primary color.”
    Input priColor1
    Display “Enter a primary color.”
    Input priColor2

    //Call module
    Call mixColor(priColor1, priColor2)

End Module

//Declare the mixColor module
//mixColor() calculates and outputs
Module mixColor(String Ref color1, String Ref color2)

    //Declare variables
    Declare String color1
    Declare String color2

    //Calculate mixSecCol with nested If-Then-Else

    //1st If calculates with color1 as Red
    If color1 == ‘RED’ OR ‘Red’ OR ‘red’ Then
        //Red + Yellow = Orange
        If color2 == ‘YELLOW’ OR ‘Yellow’ OR ‘yellow’ Then
            Display color1, “ mixed with”, color2, “makes ORANGE.”
        //Red + Blue = Purple
        Else If color2 == ‘BLUE OR ‘Blue’ OR ‘blue’ Then
            Display color1, “ mixed with”, color2, “makes PURPLE.”

        //2nd If calculates with color1 as Yellow
        Else If color1 == ‘YELLOW’ OR ‘Yellow’ OR ‘yellow’ Then
            //Yellow + Red = Orange
            If color2 == ‘RED’ OR ‘Red’ OR ‘red’ Then
                Display color1, “ mixed with”, Color2, “makes ORANGE.”
            //Yellow + Blue = Green
```

```
          ▪ Else If color2 == 'BLUE' OR 'Blue' OR 'blue' Then
            ▪ Display Color1, " mixed with", Color2, "makes GREEN."

            //3rd If calculates with color2 as Blue
          ▪ Else If color1 == 'BLUE' OR 'Blue' OR 'blue' Then
              //Blue + Red = Purple
            ▪ If color2 == 'RED' OR 'Red' OR 'red' Then
              ▪ Display color1, " mixed with", color2, "makes PURPLE."
              //Blue + Yellow = Green
            ▪ Else If color2 == 'YELLOW' OR 'Yellow' OR 'yellow' Then
              ▪ Display Color1, " mixed with", Color2, "makes GREEN."

              //Outputs an error message if argument passes
            ▪ Else
              ▪ Display "ERROR! Valid entries include."
              ▪ Display "RED, Red, red."
              ▪ Display "YELLOW, Yellow, yellow."
              ▪ Display "BLUE, Blue, blue."

              //End the If-Then-Else Loop
            ▪ End If
          ▪ End If
        ▪ End If
    g. End If

  10. End Module
```

## Step 3: The Flowchart

Refer to the png file submitted along with the PDF file as it contains the Flowchart.



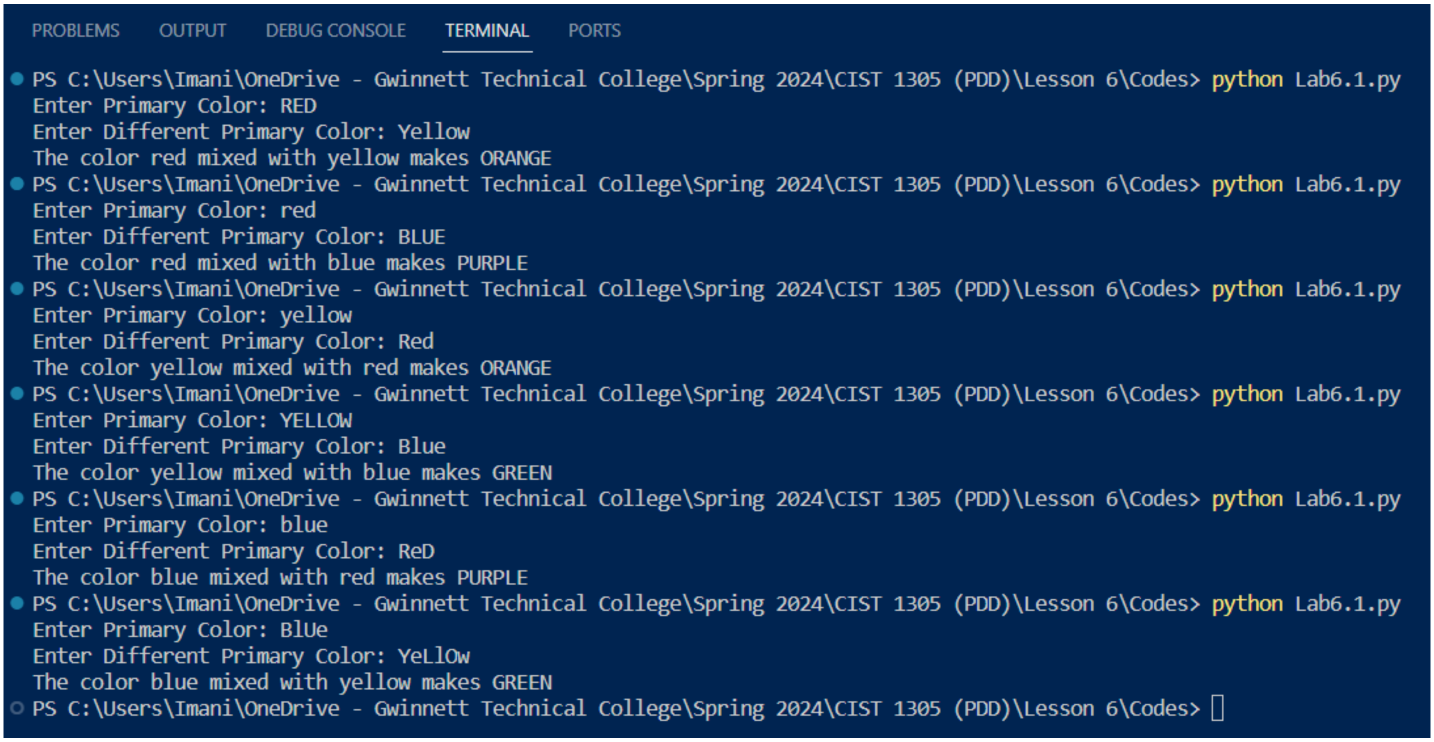
### Step 4: The Python Code

Refer to the txt file submitted along with the PDF file as it contains the Python Code.

```
#Imani Hollie 02.10.2024
#This program will collect two primary colors
#(primary colors, secondary color, or error)

#Module 1 - main() [Input and Calls mixColor()]
#Inputs--------------------------------------------------------------------------------------
#Strings can’t be floats, so I just used input()
priColor1 = input('Enter Primary Color: ')
priColor2 = input('Enter Different Primary Color: ')

#Module 2 - mixColor [Calculations and Output]
def mixColor(color1, color2):
    #Variables-------------------------------------------------------------------------------
    #The 'lower()' method converts string to lowercase for case-insensitivity
    color1 = color1.lower()
    color2 = color2.lower()
    #Calculations----------------------------------------------------------------------------
    #IF-THEN-ELSE decision structure will display secondary color or error message
    #IF color1 = red THEN IF color 2 = yellow/blue THEN display orange/purple
    if color1 == 'red':
        if color2 == 'yellow':
            print(f'The color {color1} mixed with {color2} makes ORANGE')
        elif color2 == 'blue':
            print(f'The color {color1} mixed with {color2} makes PURPLE')
        #End If
    #IF color1 = yellow THEN IF color 2 = red/blue THEN display orange/green
    elif color1 == 'yellow':
        if color2 == 'red':
            print(f'The color {color1} mixed with {color2} makes ORANGE')
        elif color2 == 'blue':
            print(f'The color {color1} mixed with {color2} makes GREEN')
        #End If
    #IF color1 = blue THEN IF color 2 = red/yellow THEN display purple/green
    elif color1 == 'blue':
        if color2 == 'red':
            print(f'The color {color1} mixed with {color2} makes PURPLE')
        elif color2 == 'yellow':
            print(f'The color {color1} mixed with {color2} makes GREEN')
        #If argument passes through - display error message
        else:
            print('ERROR! Valid Entries Include: RED, YELLOW, BLUE')
        #End If
    #End If
#End Module 2
```

```
#Calling Module 2 mixColor()-----------------------------------------------------------------
mixColor(priColor1, priColor2)
#Output is then printed to the screen.

#End Module 1
```

**Screenshot of Terminal**



```
PS C:\Users\Imani\OneDrive - Gwinnett Technical College\Spring 2024\CIST 1305 (PDD)\Lesson 6\Codes> python Lab6.1.py
Enter Primary Color: RED
Enter Different Primary Color: Yellow
The color red mixed with yellow makes ORANGE
PS C:\Users\Imani\OneDrive - Gwinnett Technical College\Spring 2024\CIST 1305 (PDD)\Lesson 6\Codes> python Lab6.1.py
Enter Primary Color: red
Enter Different Primary Color: BLUE
The color red mixed with blue makes PURPLE
PS C:\Users\Imani\OneDrive - Gwinnett Technical College\Spring 2024\CIST 1305 (PDD)\Lesson 6\Codes> python Lab6.1.py
Enter Primary Color: yellow
Enter Different Primary Color: Red
The color yellow mixed with red makes ORANGE
PS C:\Users\Imani\OneDrive - Gwinnett Technical College\Spring 2024\CIST 1305 (PDD)\Lesson 6\Codes> python Lab6.1.py
Enter Primary Color: YELLOW
Enter Different Primary Color: Blue
The color yellow mixed with blue makes GREEN
PS C:\Users\Imani\OneDrive - Gwinnett Technical College\Spring 2024\CIST 1305 (PDD)\Lesson 6\Codes> python Lab6.1.py
Enter Primary Color: blue
Enter Different Primary Color: ReD
The color blue mixed with red makes PURPLE
PS C:\Users\Imani\OneDrive - Gwinnett Technical College\Spring 2024\CIST 1305 (PDD)\Lesson 6\Codes> python Lab6.1.py
Enter Primary Color: BlUe
Enter Different Primary Color: YeLlOw
The color blue mixed with yellow makes GREEN
PS C:\Users\Imani\OneDrive - Gwinnett Technical College\Spring 2024\CIST 1305 (PDD)\Lesson 6\Codes>
```

# EXERCISE 10 – BODY MASS INDEX PROGRAM ENHANCEMENT

Write the Algorithm, Pseudocode, Flowchart, and Python Code for the following programming problem:

**Scenario: Body Mass Index Program Enhancement**

In Programming Exercise 6 in Ch. 3 you were asked to design a program that calculates a person's Body Mass Index (BMI). Recall from that exercise that the BMI is often used to determine whether a person with a sedentary lifestyle is overweight or underweight for their height. A person's BMI is calculated with the following formula:

$$BMI = Weight * 703/Height^2$$

In the formula, weight is measured in pounds (lbs.) and height is measured in inches (in.). Enhance the program so it displays a message indicating whether the person has optimal weight, is underweight, or is overweight. A sedentary person's weight is optimal if their BMI is between 18.5 – 25 lbs. If the BMI is less than 18.5 lbs., the person is underweight. If the BMI is greater than 25 lbs., the person is overweight.

**Step 1: The Algorithm**

1. MODULE 1 – main()
    a. Get the total weight of the package:
        i. Prompt for the total weight in pounds.
    b. Call module
2. MODULE 2 – shipCharge()
    a. Calculate the amount for shipping using nested If-Then-Else structure:
        i. If user inputs 2 lbs or less, then multiple total weight by 1.1
        ii. Else, If user inputs 2 – 6 lbs, then multiple total weight by 2.2
        iii. Else, If user inputs 6 – 10 lbs, then multiple total weight by 3.7
        iv. Else, If user inputs over 10 lbs, then multiple total weight by 3.8
    b. Display weight total, rate per pound, and shipping sale:
        i. Display Total Weight
        ii. Display Rate Per Pound (lbs.)
        iii. Display Total Shipping Charge

**The Input, Processing, and Output**

| Table 2-1 Calculating Total Seats Sold (x) | | | |
|---|---|---|---|
| **INPUTS** | **Input Type** | **Value** | **Data Type** |
| Total Weight (totalLbs) | Variable | (a) | Float |
| Rate Per Pound (lbsRate) | Constant | (b) | Real |
| **PROCEDURE** | $x = a * b$<br>$totalCost = totalLbs * lbsRate$ | | |
| **OUTPUTS** | **Output Type** | **Value** | **Data Type** |
| Total Cost (totalCost) | Variable | (x) | Float |

The IPO for Table 2-1 is as follows:

1. The inputs for Table 2-1 are as follows:
    a. Total Weight (a)
    b. Rate Per Pound (b)
2. The procedures for Table 2-1 are as follows:
    a. $x = a * b$
    $totalCost = totalLbs * lbsRate$
3. The output for Table 2-1 are as follows:
    a. Total Cost (x)

| Table 2-2 Nested Decision Structure: Shipping Charges | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Charge x > 2 lbs.** | | **Charge 2 < x > 6 lbs.** | | **Charge 6 < x > 10 lbs.** | | **Charge 10 < x lbs.** | |
| **totalLbs (a)** | **lbsRate (b)** | **totalLbs (a)** | **lbsRate (b)** | **totalLbs (a)** | **lbsRate (b)** | **totalLbs (a)** | **lbsRate (b)** |
| x | 1.1 | x | 2.2 | x | 3.7 | x | 3.8 |
| **totalCost (x)** | 1.1x | **totalCost (x)** | 2.2x | **totalCost (x)** | 3.7x | **totalCost (x)** | 3.8x |

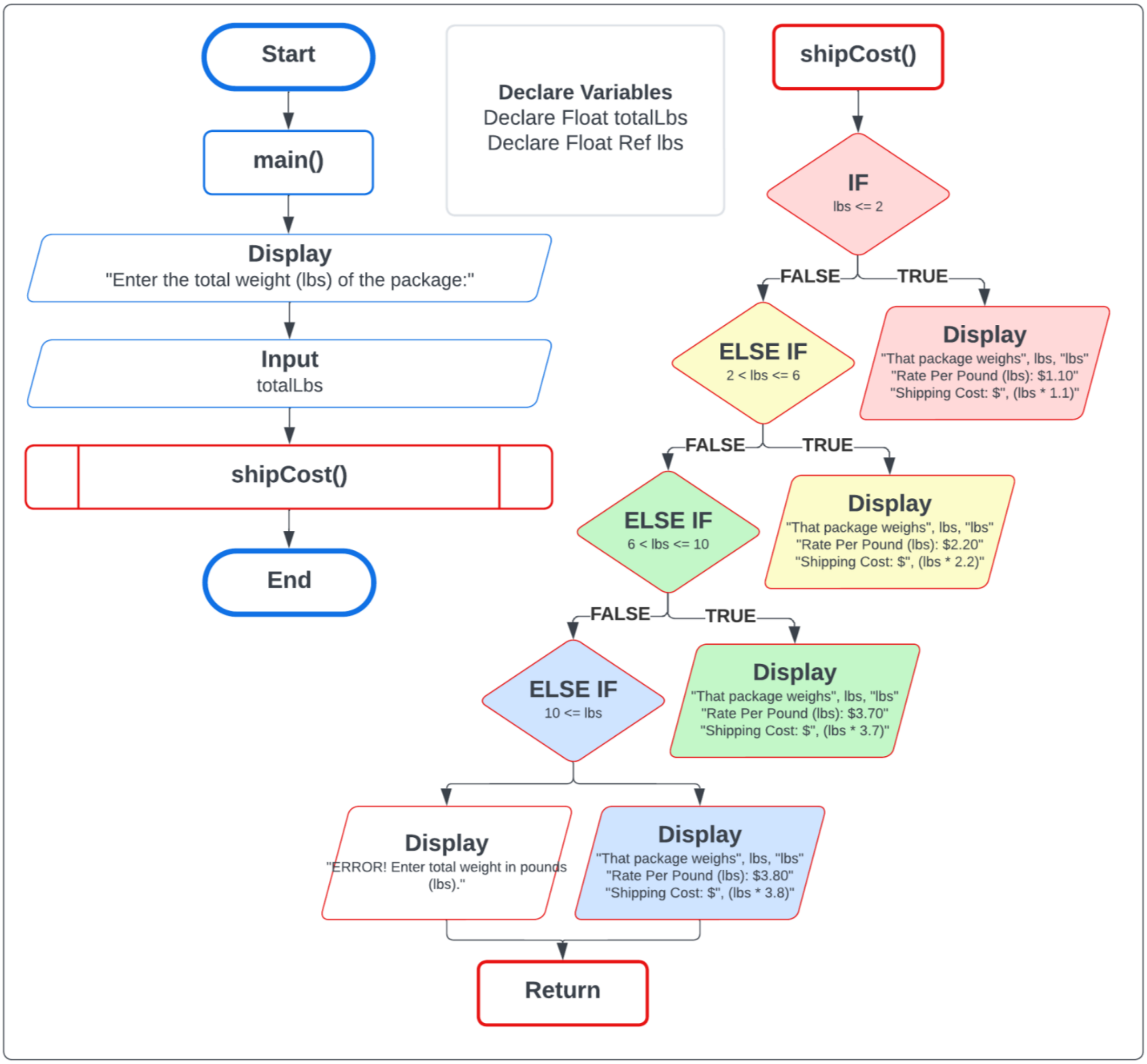
**Step 2: The Pseudocode**

Refer to Tables 2-1 and 2-2 in Step 1 for the needed variables.

```
//This program takes in the total weight of packages.
//Output is then printed to the screen.

//Declare the main module
//main() input and calls totalCost()
Module main()

    //Declare variables
    Declare Float totalLbs

    //Input totalLbs
    Display “Enter the total weight (in pounds) of the package.”
    Input totalLbs

    //Call module
    Call totalCost(totalLbs)

End Module

//Declare the totalCost module
//totalCost() calculates and outputs
Module totalCost(Float Ref lbs)

    //Declare variables
    Declare Float lbs

    //Calculate mixSecCol with nested If-Then-Else

    //1st If calculates for lbs > 2
    If lbs > 2 Then
        Display “The package weighs {lbs} lbs.”
        Display “Cost Per Pound: $1.10”
        Display “Shipping Cost: ${lbs * 1.1}”

        //2nd If calculates for 2 < lbs > 6
        Else If 2 < lbs and lbs > 6 Then
            Display “The package weighs {lbs} lbs.”
            Display “Cost Per Pound: $2.20”
            Display “Shipping Cost: ${lbs * 2.2}”

            //3rd If calculates for 6 < lbs > 10
            Else If 6 < lbs and lbs > 10 Then
                Display “The package weighs {lbs} lbs.”
                Display “Cost Per Pound: $3.70”
                Display “Shipping Cost: ${lbs * 3.7}”
            ▪ //4th If calculates for 10 < lbs
            ▪ Else If 10 < lbs Then
              ▪ Display "The package weighs {lbs} lbs."
              ▪ Display "Cost Per Pound: $3.80"
              ▪ Display "Shipping Cost: ${lbs * 3.8}"

              //Outputs an error message if argument passes
            ▪ Else
              ▪ Display "ERROR! Enter total weight in pounds (lbs)."

              //End the If-Then-Else Loop
            ▪ End If
          ▪ End If
        ▪ End If
    h. End If

  End Module
```

### Step 3: The Flowchart

Refer to the png file submitted along with the PDF file as it contains the Flowchart.

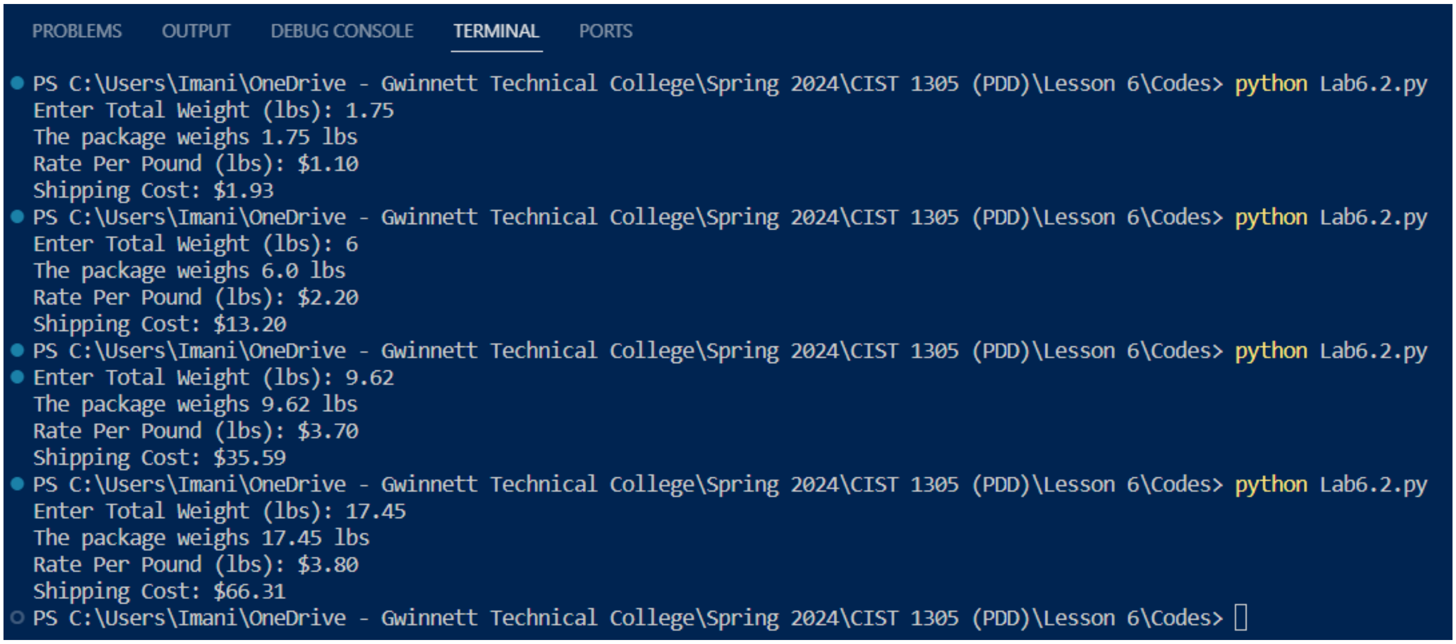

Start
main()
Display
"Enter the total weight (lbs) of the package:"
Input
totalLbs
shipCost()
End
Declare Variables
Declare Float totalLbs
Declare Float Ref lbs
shipCost()
IF
lbs <= 2
FALSE
TRUE
Display
"That package weighs", lbs, "lbs"
"Rate Per Pound (lbs): $1.10"
"Shipping Cost: $", (lbs * 1.1)"
ELSE IF
2 < lbs <= 6
FALSE
TRUE
Display
"That package weighs", lbs, "lbs"
"Rate Per Pound (lbs): $2.20"
"Shipping Cost: $", (lbs * 2.2)"
ELSE IF
6 < lbs <= 10
FALSE
TRUE
Display
"That package weighs", lbs, "lbs"
"Rate Per Pound (lbs): $3.70"
"Shipping Cost: $", (lbs * 3.7)"
ELSE IF
10 <= lbs
Display
"ERROR! Enter total weight in pounds (lbs)."
Display
"That package weighs", lbs, "lbs"
"Rate Per Pound (lbs): $3.80"
"Shipping Cost: $", (lbs * 3.8)"
Return

## Step 4: The Python Code

Refer to the txt file submitted along with the PDF file as it contains the Python Code.

```
#Imani Hollie 02.10.2024
#This program will collect package weight
#(package weight, rate per pound, final sale)

#Module 1 - main() [Input and Calls shipCost()]
#Inputs--------------------------------------------------------------------------------------
totalLbs = float(input('Enter Total Weight (lbs): '))

#Module 2 - shipCost [Calculations and Output]
def shipCost(lbs):
    #Calculations----------------------------------------------------------------------------
    #IF-THEN-ELSE decision structure will display package weight, shipping rate, and cost
    #if true or an error message if false
    #IF lbs <= (less than or equal to) 2 THEN Display package weight, shipping rate, and cost
    if lbs <= 2:
        print(f'The package weighs {lbs} lbs')
        print(f'Rate Per Pound (lbs): $1.10')
        #The '.2f' is used for decimals to limit the decimal places, two for cost
        print(f'Shipping Cost: ${lbs * 1.1:.2f}')
    #IF 2 < (less than) lbs <= 6 THEN Display package weight, shipping rate, and total cost
    elif 2 < lbs <= 6:
        print(f'The package weighs {lbs} lbs')
        print(f'Rate Per Pound (lbs): $2.20')
        print(f'Shipping Cost: ${lbs * 2.2:.2f}')
    #IF 6 < lbs <= 10 THEN Display package weight, shipping rate, and total cost
    elif 6 < lbs <= 10:
        print(f'The package weighs {lbs} lbs')
        print(f'Rate Per Pound (lbs): $3.70')
        print(f'Shipping Cost: ${lbs * 3.7:.2f}')
    #IF 10 < lbs THEN Display package weight, shipping rate, and total cost
    elif 10 < lbs:
        print(f'The package weighs {lbs} lbs')
        print(f'Rate Per Pound (lbs): $3.80')
        print(f'Shipping Cost: ${lbs * 3.8:.2f}')
    #If argument passes through - display error message
    else:
        print('ERROR! Enter total weight in pounds (lbs)')
    #End If
#End Module 2

#Calling Module 2 shipCost()-----------------------------------------------------------------
shipCost(totalLbs)
#Output is then printed to the screen.

#End Module 1
```

### Screenshot of Terminal



```
PS C:\Users\Imani\OneDrive - Gwinnett Technical College\Spring 2024\CIST 1305 (PDD)\Lesson 6\Codes> python Lab6.2.py
Enter Total Weight (lbs): 1.75
The package weighs 1.75 lbs
Rate Per Pound (lbs): $1.10
Shipping Cost: $1.93
PS C:\Users\Imani\OneDrive - Gwinnett Technical College\Spring 2024\CIST 1305 (PDD)\Lesson 6\Codes> python Lab6.2.py
Enter Total Weight (lbs): 6
The package weighs 6.0 lbs
Rate Per Pound (lbs): $2.20
Shipping Cost: $13.20
PS C:\Users\Imani\OneDrive - Gwinnett Technical College\Spring 2024\CIST 1305 (PDD)\Lesson 6\Codes> python Lab6.2.py
Enter Total Weight (lbs): 9.62
The package weighs 9.62 lbs
Rate Per Pound (lbs): $3.70
Shipping Cost: $35.59
PS C:\Users\Imani\OneDrive - Gwinnett Technical College\Spring 2024\CIST 1305 (PDD)\Lesson 6\Codes> python Lab6.2.py
Enter Total Weight (lbs): 17.45
The package weighs 17.45 lbs
Rate Per Pound (lbs): $3.80
Shipping Cost: $66.31
PS C:\Users\Imani\OneDrive - Gwinnett Technical College\Spring 2024\CIST 1305 (PDD)\Lesson 6\Codes>
```