LAB 6 ASSIGNMENT
Ch. 4 Decision Structures and Boolean Logic
Start: 02/09/2024 6:04 PM
Name: Imani Hollie

EXCERCISE 5 – COLOR MIXER

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

Scenario: Color Mixer

The colors red, blue, and yellow are known as primary colors because they cannot be made by mixing other colors. When you mix two primary colors, you get a secondary color, as shown here:

- When you mix red and blue, you get purple
- When you mix red and yellow, you get yellow
- When you mix yellow and blue, you get green

Design a program that prompts the user to enter the names of two primary colors to mix. If the user enters anything other than, 'red', 'blue', or 'yellow', the program should display an error message. Otherwise, the program should display the name of the secondary color that results.

Step 1: The Algorithm

1. MODULE 1 – main()
   a. Get the two colors to mix:
      i. Prompt for the first color
      ii. Prompt for the second color
   b. Call module
2. MODULE 2 – colorMix()
   a. Calculate the color mixed using nested If-Then-Else structure:
      i. If user inputs red and blue, then display purple
      ii. Else, If user inputs red and yellow, then display orange
      iii. Else, If user inputs yellow and blue, then display green
      iv. Else, If user inputs anything else, then display an error message
   b. Display the color mixed:
      i. Display First Color
      ii. Display Second Color
      iii. Display Mixed Color

The Input, Processing, and Output

Table 1-1 Calculating Color Mixed (x)			
INPUTS	**Input Type**	**Value**	**Data Type**
First Primary Color (priColor1)	Variable	(a)	String
First Secondary Color (priColor2)	Variable	(b)	String
PROCEDURE	$x = a + b$ $SecColor = priColor1 + priColor2$		
OUTPUTS	**Output Type**	**Value**	**Data Type**
Mixed Secondary Color (SecColor)	Variable	(x)	String

The IPO for Table 1-1 is as follows:

1. The inputs for Table 1-1 are as follows:
    a. First Primary Color (a)
    b. Second Primary Color (b)
2. The procedure for Table 1-1 are as follows:
    a. $x = a + b$
       $mixSecColor = priColor1 + priColor2$
3. The output for Table 1-1 are as follows:
    a. Mixed Secondary Color (x)

Table 1-2 Nested Decision Structure: Mixing Colors								
Mixing Red as priColor1			**Mixing Yellow as priColor1**			**Mixing Blue as priColor1**		
priColor1 (a)	**priColor2 (b)**		**priColor1 (a)**	**priColor2 (b)**		**priColor1 (a)**	**priColor2 (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 (x)	Orange	Purple	**mixSecCol (x)**	Orange	Green	**mixSecCol (x)**	Purple	Green

Step 2: The Pseudocode

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

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

3. **//Declare the main module**
4. **//main() input and calls mixColor()**
5. Module main()

   a. **//Declare variables**
   b. Declare String priColor1
   c. Declare String priColor2

   d. **//Input priColor1 and priColor2**
   e. Display "Enter a primary color."
   f. Input priColor1
   g. Display "Enter a primary color."
   h. Input priColor2

   i. **//Call module**
   j. Call mixColor(priColor1, priColor2)

6. End Module

7. **//Declare the mixColor module**
8. **//mixColor() calculates and outputs**
9. Module mixColor(String Ref color1, String Ref color2)

   a. **//Declare variables**
   b. Declare String color1
   c. Declare String color2

   d. **//Calculate mixSecCol with nested If-Then-Else**

   e. **//1st If calculates with color1 as Red**
   f. 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

STARTING OUT WITH PROGRAMMING LOGIC AND DESIGN CH. 4

- 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.

STARTING OUT WITH PROGRAMMING LOGIC AND DESIGN CH. 4
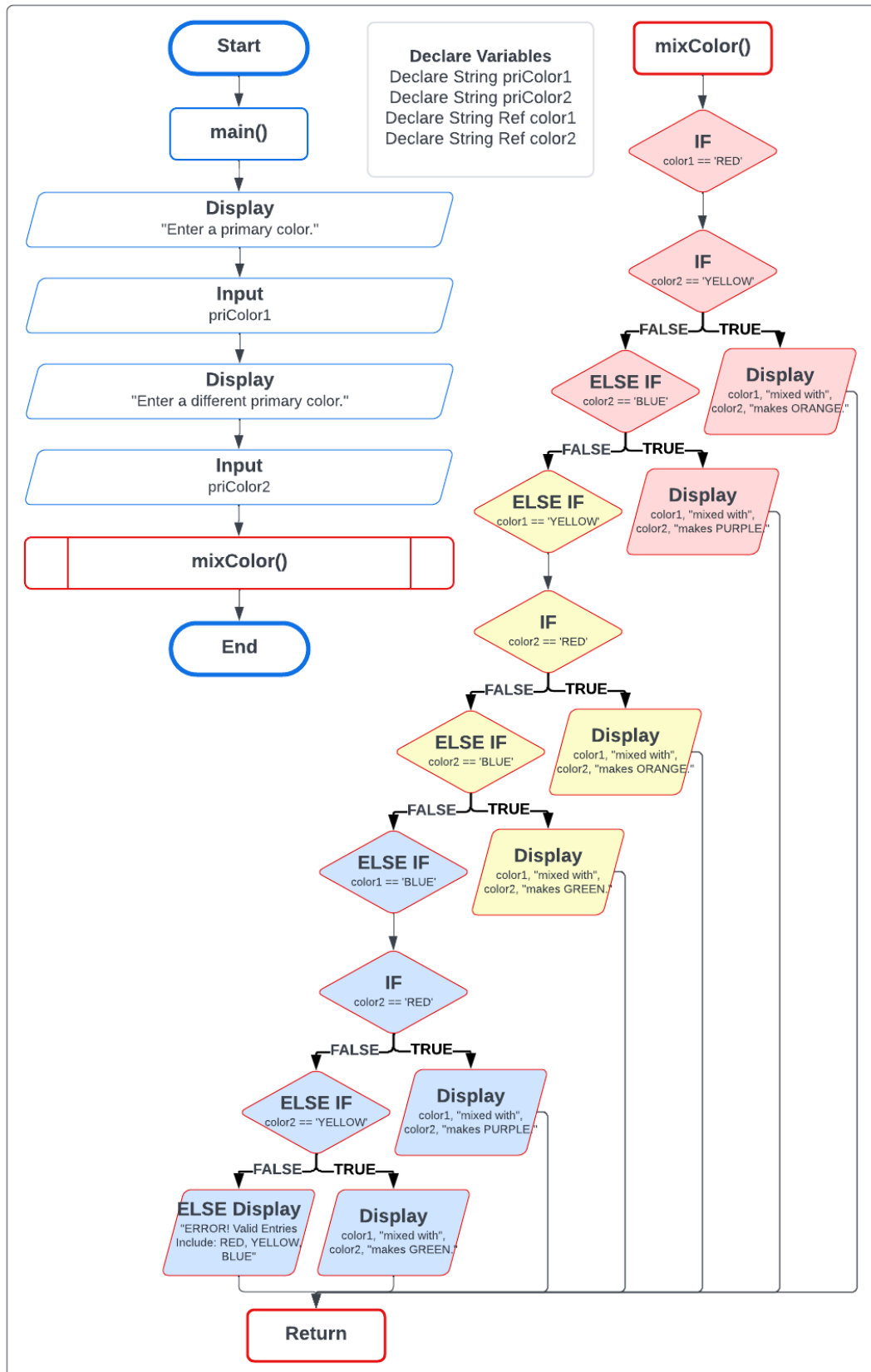
Step 4: The Python Code

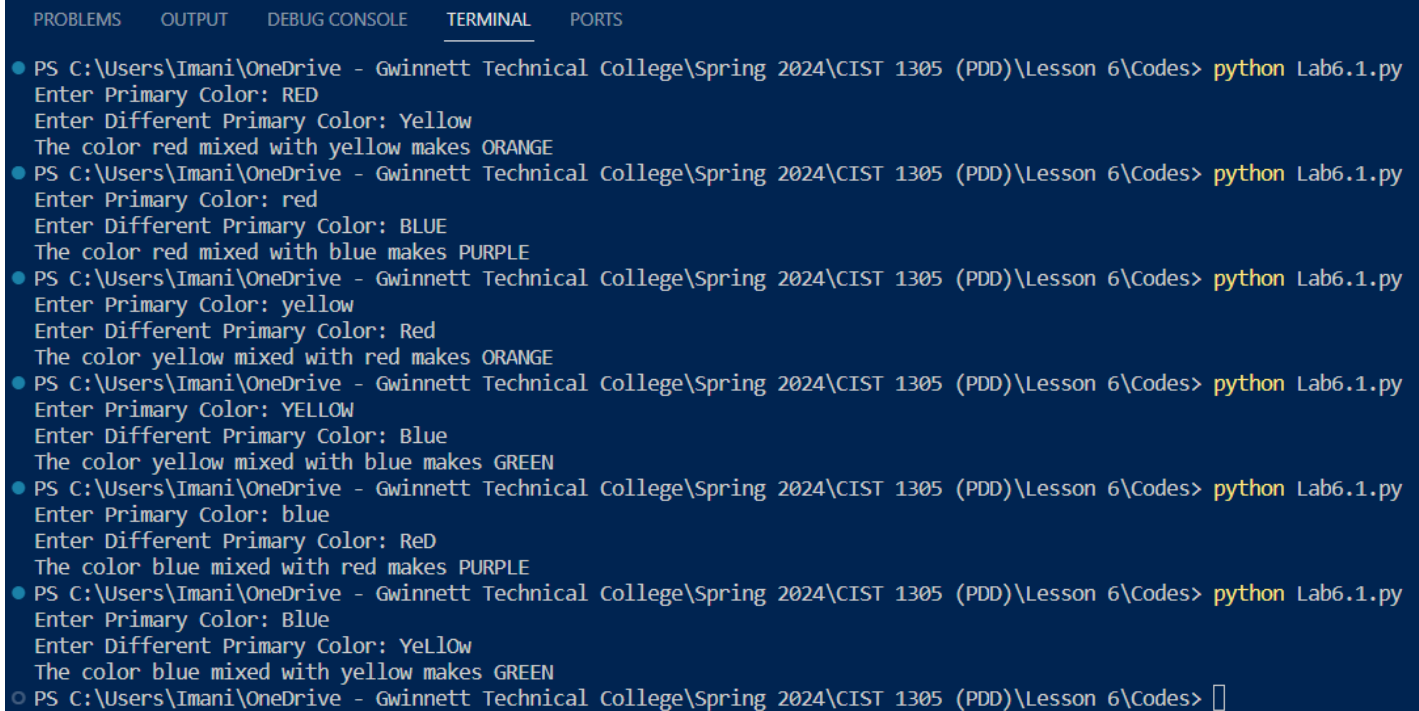Refer to the txt file submitted along with the PDF file as it contains the Python Code.

```python
#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

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

● 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 9 – SHIPPING CHARGES

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

Scenario: Shipping Charges

The Fast Freight Shipping Company charges the following rates:

Weight of Package	Rate per Pound
2 pounds or less	$1.10
Over 2 pounds but no more than 6 pounds	$2.20
Over 6 pounds but no more than 10 pounds	$3.70
Over 10 pounds	$3.80

Design a program that asks the user to enter the weight of a package and then displays the shipping charges.

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	$x = a * b$ $totalCost = totalLbs * lbsRate$		
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.

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

3. **//Declare the main module**
4. **//main() input and calls totalCost()**
5. Module main()

    a. **//Declare variables**
    b. Declare Float totalLbs

    c. **//Input totalLbs**
    d. Display "Enter the total weight (in pounds) of the package."
    e. Input totalLbs

    f. **//Call module**
    g. Call totalCost(totalLbs)

6. End Module

7. **//Declare the totalCost module**
8. **//totalCost() calculates and outputs**
9. Module totalCost(Float Ref lbs)

    a. **//Declare variables**
    b. Declare Float lbs

    c. **//Calculate mixSecCol with nested If-Then-Else**

    d. **//1st If calculates for lbs > 2**
    e. 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}"

STARTING OUT WITH PROGRAMMING LOGIC AND DESIGN CH. 4

- **//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
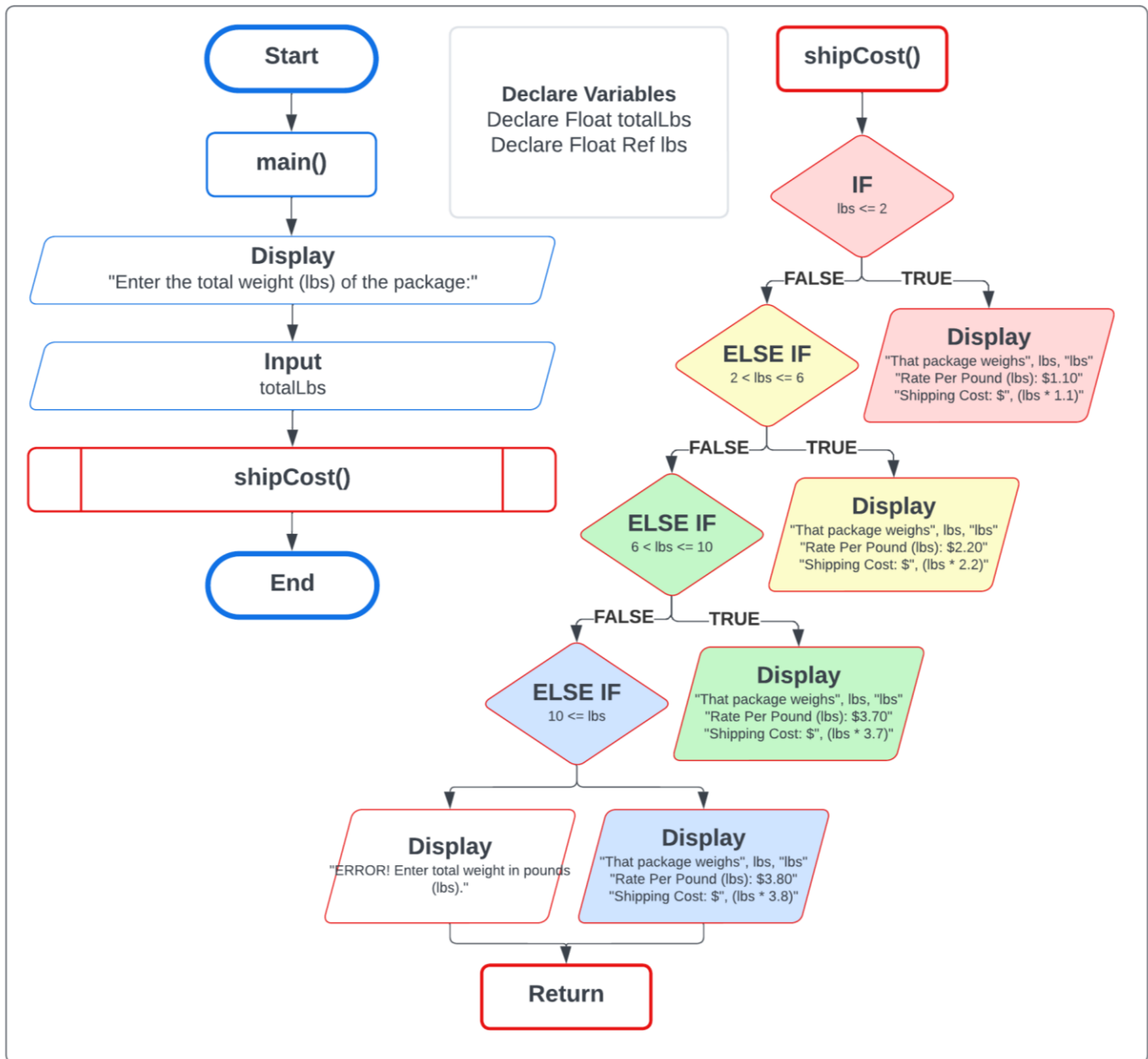 - Display "ERROR! Enter total weight in pounds (lbs)."

  - **//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.

Step 4: The Python Code

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

```python
#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
```
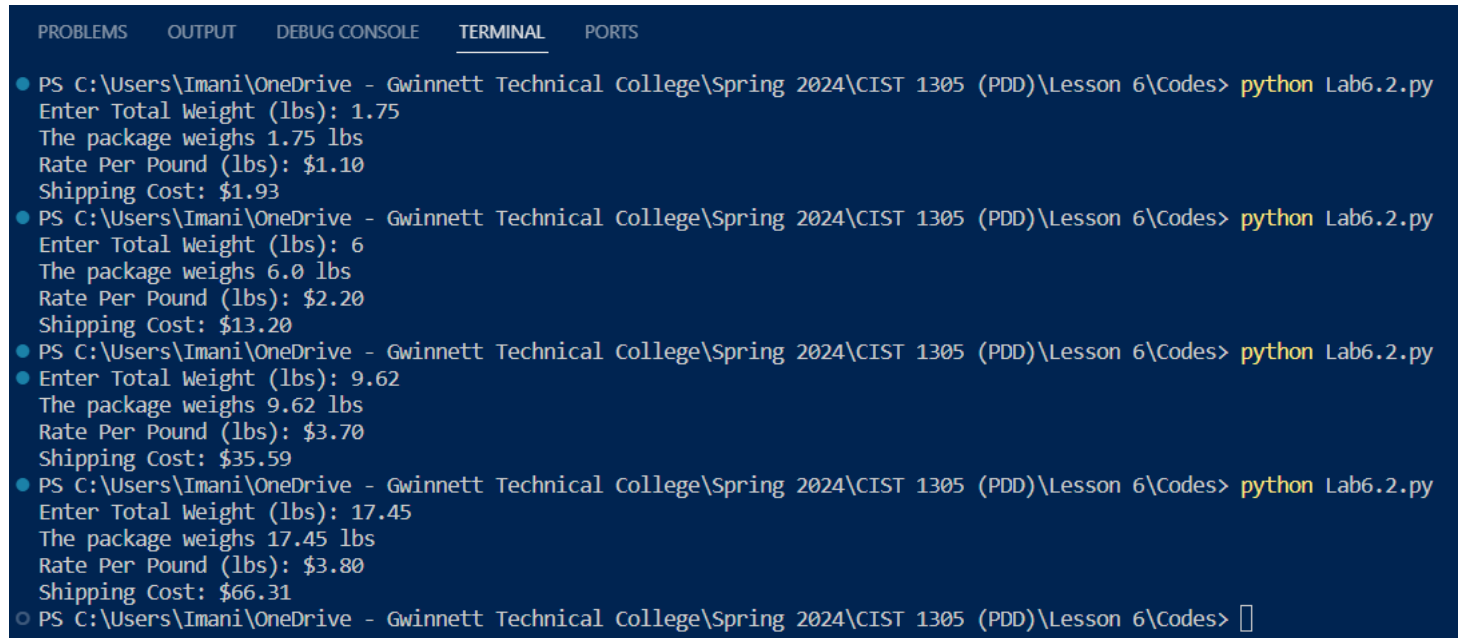
STARTING OUT WITH PROGRAMMING LOGIC AND DESIGN CH. 4

Screenshot of Terminal

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

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> 
```