LAB 4-2 ASSIGNMENT

Ch. 3 Modules

Start: 02/02/2024 4:37 PM
Name: Imani Hollie

EXCERCISE 3 – HOW MUCH INSURANCE?

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

Scenario: Calculate Insurance

Many financial experts advise that property owners should insure their homes or buildings for at least 80% of the amount it would cost to replace the structure. Design a modular program that asks the user to enter the replacement cost of a building and then displays the minimum amount of insurance they should buy for the property.

Step 1: The Algorithm

1. MODULE 1 – main()
    a. Get the total for the replacement cost:
        i. Prompt for the total amount needed for replacement cost
    b. Call module
2. MODULE 2 – minInsCov()
    a. Calculate the minimum insurance:
        i. Multiply minimum insurance by total for replacement cost for property
            i. Multiply 0.8 by the total for replacement cost for Minimum Insurance
    b. Display total for replacement cost and total for minimum insurance coverage:
        i. Display Total Replacement Cost
        ii. Display Minimum Insurance Coverage

The Input, Processing, and Output

Table 1-1 Calculating Minimum Insurance Coverage (x)			
INPUTS	**Input Type**	**Value**	**Data Type**
Total Replacement Cost (totalCost)	Variable	(a)	Float
PROCEDURE	$x = a * 0.8$ $minInsCov = totalCost * 0.8$		
OUTPUTS	**Output Type**	**Value**	**Data Type**
Minimum Insurance Coverage (minInsCov)	Variable	(x)	Float

The IPO for Table 1-1 is as follows:

1. The inputs for Table 1-1 are as follows:
    a. Total Replacement Cost (a)
2. The procedure for Table 1-1 are as follows:
    a. $x = a * 0.8$
    $minInsCov = totalCost * 0.8$
3. The output for Table 1-1 are as follows:
    a. Minimum Insurance Coverage (x)

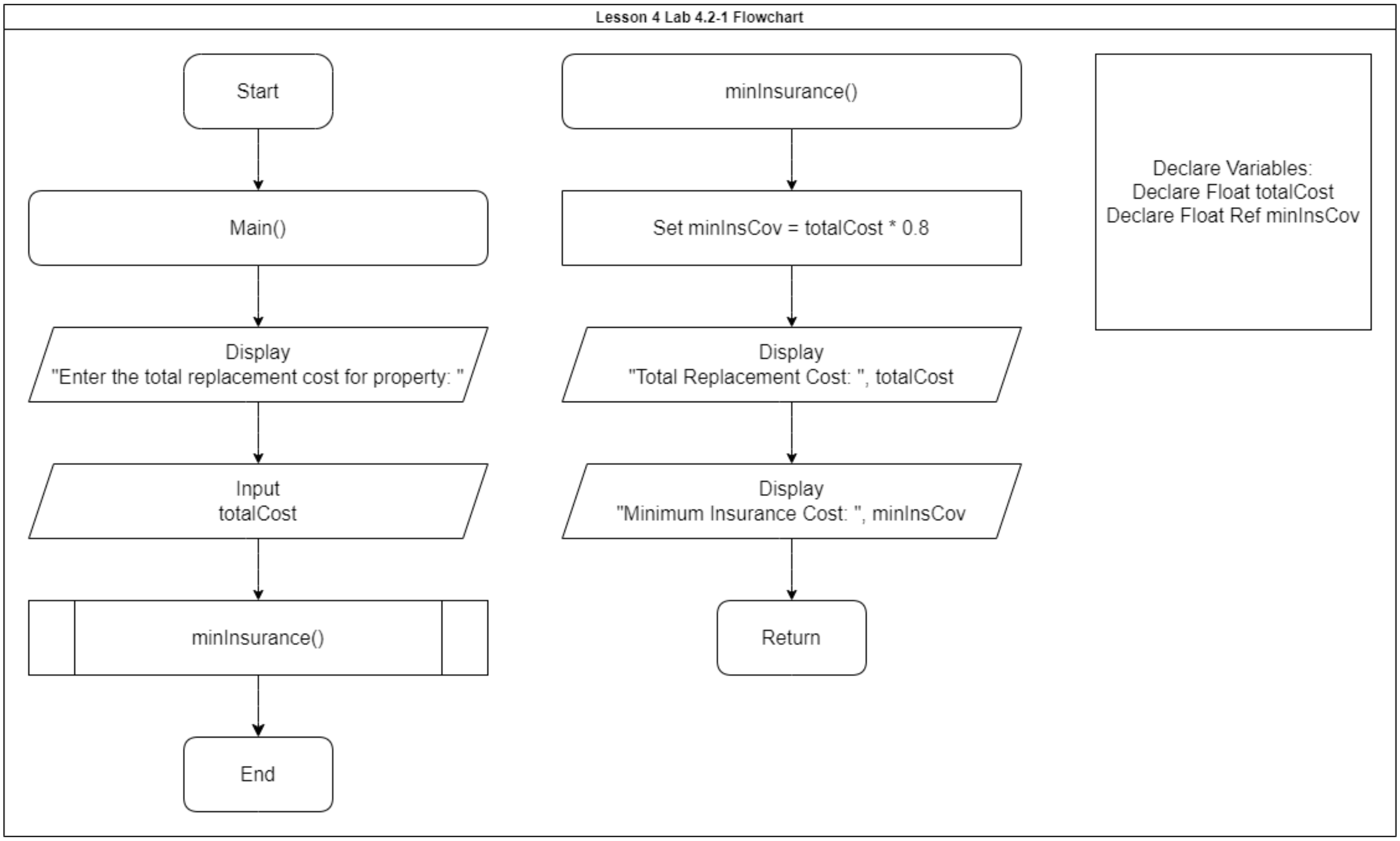
Step 2: The Pseudocode

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

```
1. //This program takes in the total replacement cost for property and
2. //the minimum amount of insurance recommended to cover it.
3. //Output is then printed to the screen.

4. //Declare the main module
5. //main() input and calls minInsurance()
6. Module main()

   a. //Declare variables
   b. Declare Float totalCost

   c. //Input totalCost
   d. Display “Enter the total replacement cost for your property.”
   e. Input totalCost

   f. //Call module
   g. Call minInsurance(totalCost)

7. End Module

8. //Declare the minInsurance module
9. //minInsurance() calculates and outputs
10. Module minInsurance(Float Ref minInsCov)

   a. //Declare variables
   b. Declare Float minInsCov

   c. //Calculate minInsCov
   d. Set minInsCov = totalCost * 0.8

   e. //Display total replacement cost and minimum insurance coverage
   f. Display “Total Replacement Cost: “, totalCost
   g. Display “Minimum Insurance Cost: “, minInsCov

11. End Module
```

Step 3: The Flowchart

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

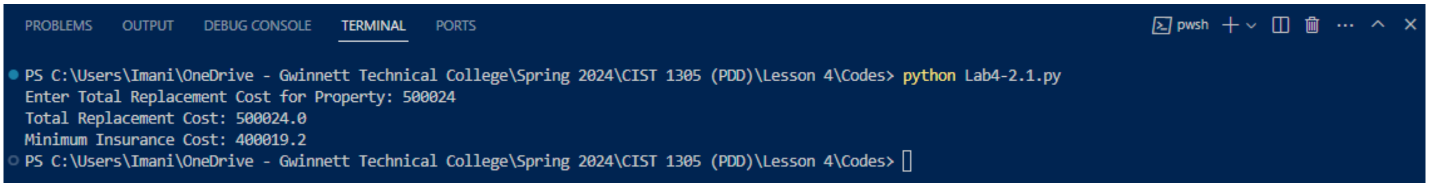

Lesson 4 Lab 4.2-1 Flowchart
Start
Main()
Display
"Enter the total replacement cost for property: "
Input
totalCost
minInsurance()
End
minInsurance()
Set minInsCov = totalCost * 0.8
Display
"Total Replacement Cost: ", totalCost
Display
"Minimum Insurance Cost: ", minInsCov
Return
Declare Variables:
Declare Float totalCost
Declare Float Ref minInsCov

Step 4: The Python Code

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

```
#Imani Hollie 02.02.2024
#This program will collect total replacement cost for property
#(replacement total and min. insurance coverage)

#Module 1 - main() [Input and Calls minIns()]
totalCost = float(input('Enter Total Replacement Cost for Property: '))

#Module 2 - minIns() [Calculations and Output]
def minIns(Cost):
    #Calculations-------------------------------------------------------------------------
    minInsCov = Cost * 0.8
    #Output-------------------------------------------------------------------------------
    print('Total Replacement Cost:', Cost)
    print('Minimum Insurance Cost:', minInsCov)
    #Output is then printed to the screen
#End Module 2

minIns(totalCost)
#End Module 1
```

Screenshot of Terminal



```
PS C:\Users\Imani\OneDrive - Gwinnett Technical College\Spring 2024\CIST 1305 (PDD)\Lesson 4\Codes> python Lab4-2.1.py
Enter Total Replacement Cost for Property: 500024
Total Replacement Cost: 500024.0
Minimum Insurance Cost: 400019.2
PS C:\Users\Imani\OneDrive - Gwinnett Technical College\Spring 2024\CIST 1305 (PDD)\Lesson 4\Codes>
```

EXERCISE 8 – STADIUM SEATING

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

> **Scenario: Stadium Seating**
>
> There are three seating categories at a stadium. For a softball game, Class A seats cost $15, Class B seats cost 12$, and Class C seats cost $9. Design a modular program that asks how many tickets for each class of seats were sold, and then displays the amount of income generated from ticket sales.

Step 1: The Algorithm

1. MODULE 1 – main()
   a. Get the total for the seats sold:
      i. Prompts for the total amount of Class A tickets sold
      ii. Prompts for the total amount of Class B tickets sold
      iii. Prompts for the total amount of Class C tickets sold
   b. Calculate the Total Seats Sold:
      i. Add the total of each ticket class for total seats sold
   c. Display Total Seats Sold:
      i. Display Total Class A Seats
      ii. Display Total Class B Seats
      iii. Display Total Class C Seats
      iv. Display Total Seats
2. MODULE 2 – totalTicketSales()
   a. Calculate the amount for each seat sold:
      i. Multiply total Class A tickets sold by 15
      ii. Multiply total Class B tickets sold by12
      iii. Multiply total Class C tickets sold by 9
      iv. Add the total of each sale for ticket class and add the total for total sales sold
   b. Display total for each ticket class sold and total ticket sales:
      i. Display Total Class A Ticket Sales
      ii. Display Total Class B Ticket Sales
      iii. Display Total Class C Ticket Sales
      iv. Display Total Ticket Sales

The Input, Processing, and Output

Table 2-1 Calculating Total Seats Sold (x)			
INPUTS	**Input Type**	**Value**	**Data Type**
Class A Seats (aSeat)	Variable	(a)	Float
Class B Seats (bSeat)	Variable	(b)	Float
Class C Seats (cSeat)	Variable	(c)	Float
PROCEDURE	$x = (a + b + c)$ $totalSeats = aSeat + bSeat + cSeat$		
OUTPUTS	**Output Type**	**Value**	**Data Type**
Total Seats (totalSeats)	Variable	(x)	Float

The IPO for Table 2-1 is as follows:

1. The inputs for Table 2-1 are as follows:
   a. Class A Seats (a)
   b. Class B Seats (b)
   c. Class C Seats (c)
2. The procedures for Table 2-1 are as follows:
   a. $x = (a + b + c)$
      $totalSeats = classASeat + classBSeat + classCSeat$
3. The output for Table 2-1 are as follows:
   a. Total Class A Seats (a)
   b. Total Class B Seats (b)
   c. Total Class C Seats (c)
   d. Total Seats (x)

Table 2-2 Calculating Total Sales (y)			
INPUTS	**Input Type**	**Value**	**Data Type**
Class A Seats (aSeat)	Variable	(a)	Float
Class B Seats (bSeat)	Variable	(b)	Float
Class C Seats (cSeat)	Variable	(c)	Float
Class A Cost (aCost)	Variable	(d)	Float
Class B Cost (bCost)	Variable	(e)	Float
Class C Cost (cCost)	Variable	(f)	Float
PROCEDURE	$d = a * 15$ $aCost = aSeat * 15$		
	$e = b * 12$ $bCost = bSeat * 12$		
	$f = c * 9$ $cCost = cSeat * 9$		
	$y = (d + e + f)$ $totalCost = aCost + bCost + cCost$		
OUTPUTS	**Output Type**	**Value**	**Data Type**
Total Sales (totalSales)	Variable	(y)	Float

The IPO for Table 2-2 is as follows:

1. The inputs for Table 2-2 are as follows:
   a. Class A Seats (a)
   b. Class B Seats (b)
   c. Class C Seats (c)

   d. Class A Cost (d)
   e. Class B Cost (e)
   f. Class C Cost (f)
   g. Total Cost (y)
2. The procedures for Table 2-2 are as follows:
   a. $d = a * 15$
      $aCost = aSeat * 15$
   b. $e = b * 12$
      $bCost = bSeat * 12$
   c. $f = c * 9$
      $cCost = cSeat * 15$
   d. $y = (d + e + f)$
      $totalCost = aCost + bCost + cCost$
3. The output for Table 2-2 are as follows:
   a. Total Class A Cost (d)
   b. Total Class B Cost (e)
   c. Total Class C Cost (f)
   d. Total Sales (y)

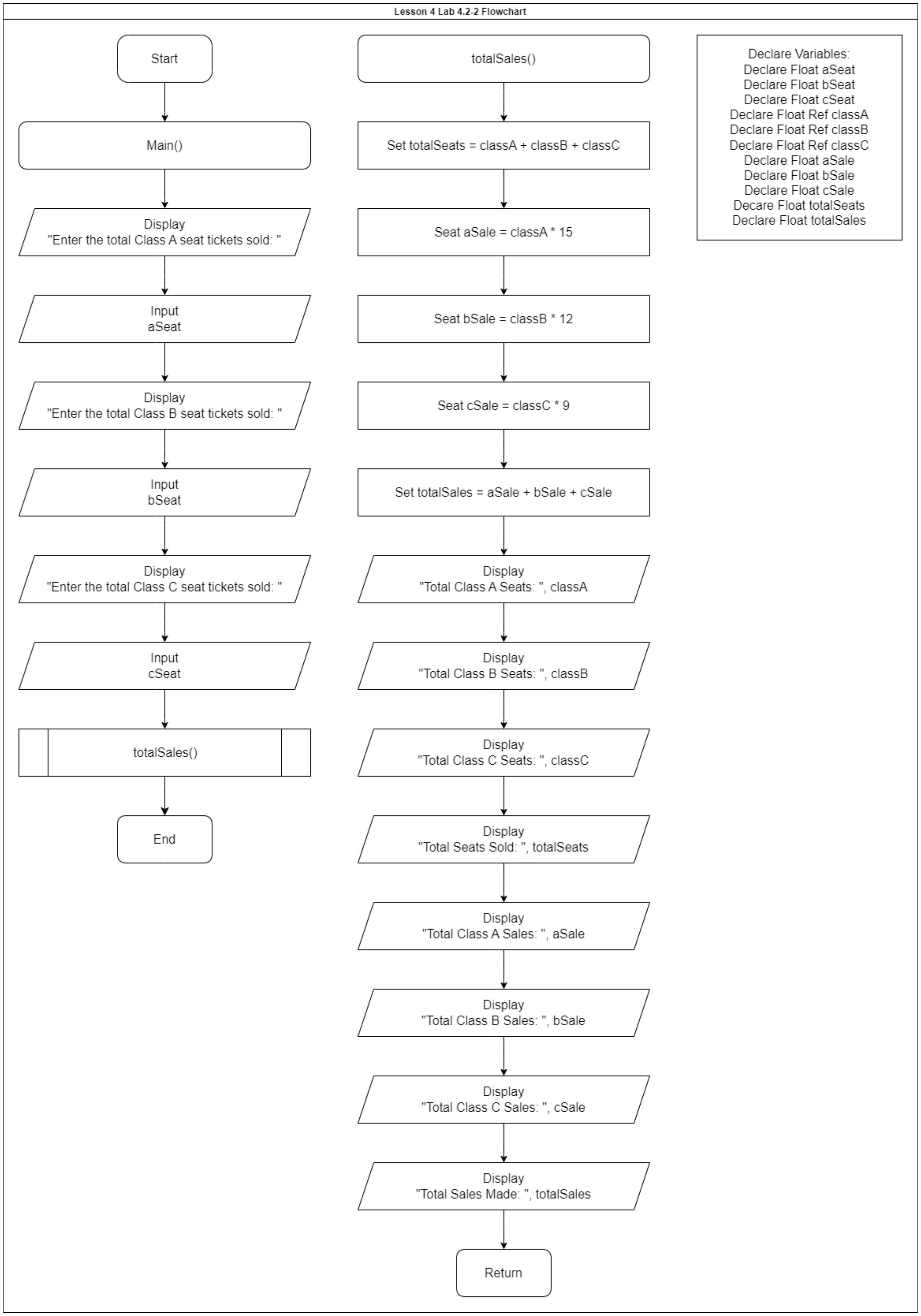
Step 2: The Pseudocode

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

```
//This program takes in the total number of seats sold per class.
//Output is then printed to the screen.

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

    //Declare variables
    Declare Float aSeat
    Declare Float bSeat
    Declare Float cSeat

    //Input totalCost
    Display “Enter the total amount of Class A tickets sold.”
    Input aSeat
    Display “Enter the total amount of Class B tickets sold.”
    Input bSeat
    Display “Enter the total amount of Class C tickets sold.”
    Input cSeat

    //Call module
    Call totalSales(aSeat, bSeat, cSeat)

End Module

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

    //Declare variables
    Declare Float totalSeats
    Declare Float totalCosts
    Declare Float aCost
    Declare Float bCost
    Declare Float cCost

    //Calculate minInsCov
    Set totalSeats = aSeat + bSeat + cSeat
    Set aCost = aSeat * 15
    Set bCost = bSeat * 12
    Set cCost = cCost * 9
    Set totalCosts = aCost + bCost + cCost

    //Display total replacement cost and minimum insurance coverage
    Display “Total Class A Seats Sold: “, aSeat
    Display “Total Class B Seats Sold: “, bSeat
    Display “Total Class C Seats Sold: “, cSeat
    Display “Total Seats Sold: “, totalSeats
    r. Display “Total Cost of Class A Seats: “, aCost
    s. Display “Total Cost of Class B Seats: “, bCost
    t. Display “Total Cost of Class C Seats: “, cCost
    u. Display “Total Sales Cost: “, totalCosts

10. End Module
```

Step 3: The Flowchart

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

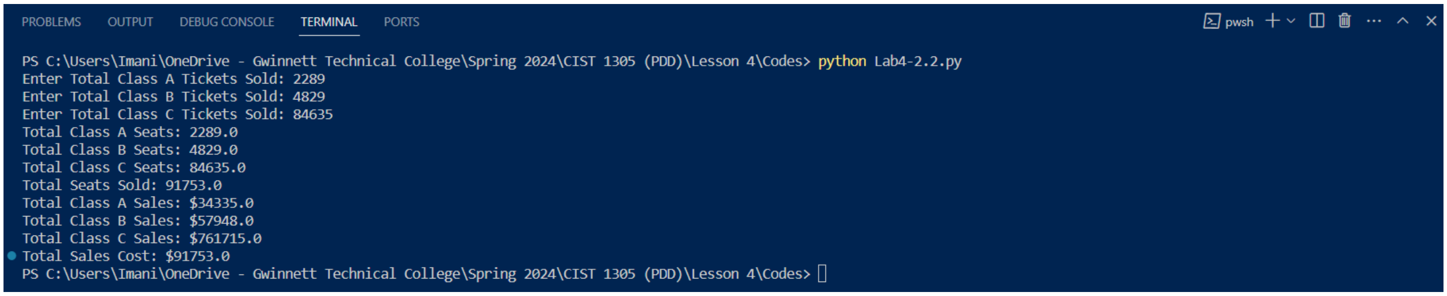

Lesson 4 Lab 4.2-2 Flowchart
Start
Main()
Display
"Enter the total Class A seat tickets sold: "
Input
aSeat
Display
"Enter the total Class B seat tickets sold: "
Input
bSeat
Display
"Enter the total Class C seat tickets sold: "
Input
cSeat
totalSales()
End
totalSales()
Set totalSeats = classA + classB + classC
Seat aSale = classA * 15
Seat bSale = classB * 12
Seat cSale = classC * 9
Set totalSales = aSale + bSale + cSale
Display
"Total Class A Seats: ", classA
Display
"Total Class B Seats: ", classB
Display
"Total Class C Seats: ", classC
Display
"Total Seats Sold: ", totalSeats
Display
"Total Class A Sales: ", aSale
Display
"Total Class B Sales: ", bSale
Display
"Total Class C Sales: ", cSale
Display
"Total Sales Made: ", totalSales
Return
Declare Variables:
Declare Float aSeat
Declare Float bSeat
Declare Float cSeat
Declare Float Ref classA
Declare Float Ref classB
Declare Float Ref classC
Declare Float aSale
Declare Float bSale
Declare Float cSale
Decare Float totalSeats
Declare Float totalSales

Step 4: The Python Code

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

```
#Imani Hollie 02.02.2024
#This program will collect total seats sold per class
#(total seats per class, total seats sold,
#total cost per class, total sales made)

#Module 1 - main() [Input and Calls totalSales()]
#Inputs---------------------------------------------------------------------------------
aSeat = float(input('Enter Total Class A Tickets Sold: '))
bSeat = float(input('Enter Total Class B Tickets Sold: '))
cSeat = float(input('Enter Total Class C Tickets Sold: '))

#Module 2 - totalSales() [Calculations and Output]
def totalSales(classA, classB, classC):
    #Calculations-----------------------------------------------------------------------
    totalSeats = classA + classB + classC
    aSale = classA * 15
    bSale = classB * 12
    cSale = classC * 9
    totalSales = classA + classB + classC
    #Output-----------------------------------------------------------------------------
    print(f'Total Class A Seats: {classA}')
    print(f'Total Class B Seats: {classB}')
    print(f'Total Class C Seats: {classC}')
    print(f'Total Seats Sold: {totalSeats}')
    print(f'Total Class A Sales: ${aSale}')
    print(f'Total Class B Sales: ${bSale}')
    print(f'Total Class C Sales: ${cSale}')
    print(f'Total Sales Cost: ${totalSales}')
    #Output is then printed to the screen
#End Module 2

#Calling Module 2 totalSales()----------------------------------------------------------
totalSales(aSeat, bSeat, cSeat)
#End Module 1
```

Screenshot of Terminal



```
PS C:\Users\Imani\OneDrive - Gwinnett Technical College\Spring 2024\CIST 1305 (PDD)\Lesson 4\Codes> python Lab4-2.2.py
Enter Total Class A Tickets Sold: 2289
Enter Total Class B Tickets Sold: 4829
Enter Total Class C Tickets Sold: 84635
Total Class A Seats: 2289.0
Total Class B Seats: 4829.0
Total Class C Seats: 84635.0
Total Seats Sold: 91753.0
Total Class A Sales: $34335.0
Total Class B Sales: $57948.0
Total Class C Sales: $761715.0
Total Sales Cost: $91753.0
PS C:\Users\Imani\OneDrive - Gwinnett Technical College\Spring 2024\CIST 1305 (PDD)\Lesson 4\Codes> 
```