

Exemplar Evidence NEA

Phones 4 U

Analysis

What is the task?

- The task I have been given is to create a program that calculates the bonuses of staff members that work in a phone shop based on sales. If the staff member averages over 4 mobile phone sales per month they receive their bonus which is calculated by using the average sales and multiplying it by £5. The program needs to inform the manager of the total sales of each staff member and also the total sales of the shop overall.

Stages to Development

- I will break the problem down to solve the task in sections. My evidence will show how I solved each of the sections below individually to solve the overall task:
 1. Get the program to take input from the user, calculate the average sales and then decide whether they get a bonus.
 2. Make the program work so it works for multiple staff members
 3. Calculate, store and then display the total sales for each staff member
 4. Calculate and then display the total sales for the shop
 5. Calculate and display the bonus amount for each staff member

Requirement 1

Get the program to take input from the user, calculate the average sales and then decide whether they get a bonus.

Success Criteria

- Allow the user to input their name
- Allow the user to input their sales for each month
- Each month of sales are added together to get a total
- Calculate the average sales based on the total sales
- Check if the average sales are enough to get a bonus
 - If they are output a message saying they are awarded with a bonus
 - If they aren't output a message saying they are not awarded a bonus

Approach to be Used

- I will use **input** to get the two inputs from the user and store them in variables
- I will have a loop that will repeat 12 times to take the 12 inputs of the monthly sales.
- I will create a running total of sales by adding the latest user input to the previous total already stored.
- I will divide the total sales by 12 to find the average
- I will use an IF statement to see if the average is more than 4 to decide whether or not they get a bonus.

IPOD Table

Input(s)	Process(es)	Output(s)	Decision(s)
Staff Name Staff Monthly sales (x12)	Calculate the total sales (monthly sales added together) Calculate the average sales (total divided by 12)	If they get a bonus or not	Whether they have entered 12 monthly sales Whether the average is enough to get a bonus

Variables to be Used

Variable Name	Data Type	Purpose
Name	String	To store the name of the staff member that has been entered
Total	Integer	To store the running total of sales as each month is entered by the user.
Sales	Integer	To store the sales for each individual month as they are entered by the user.
Average	Decimal	To store the average sales after the calculation has been made.

Possible Validation

Validation Type	Where	Reason
Presence check	Sales	To make sure that each time the number of sales for each month is entered rather than having blank entries.
Presence check	Name	To make sure that a staff member's name is entered
Format check	Sales	To make sure that the sales are a numerical value

Pseudocode

INPUT name

Total = 0

FOR loop to repeat 12 times

 INPUT monthly sales

 Total = total + monthly sales

END FOR

Average = total / 12

If average > 4 then

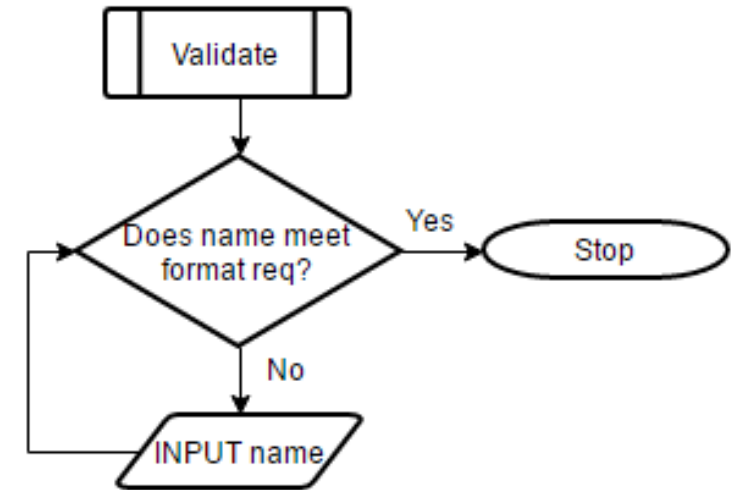
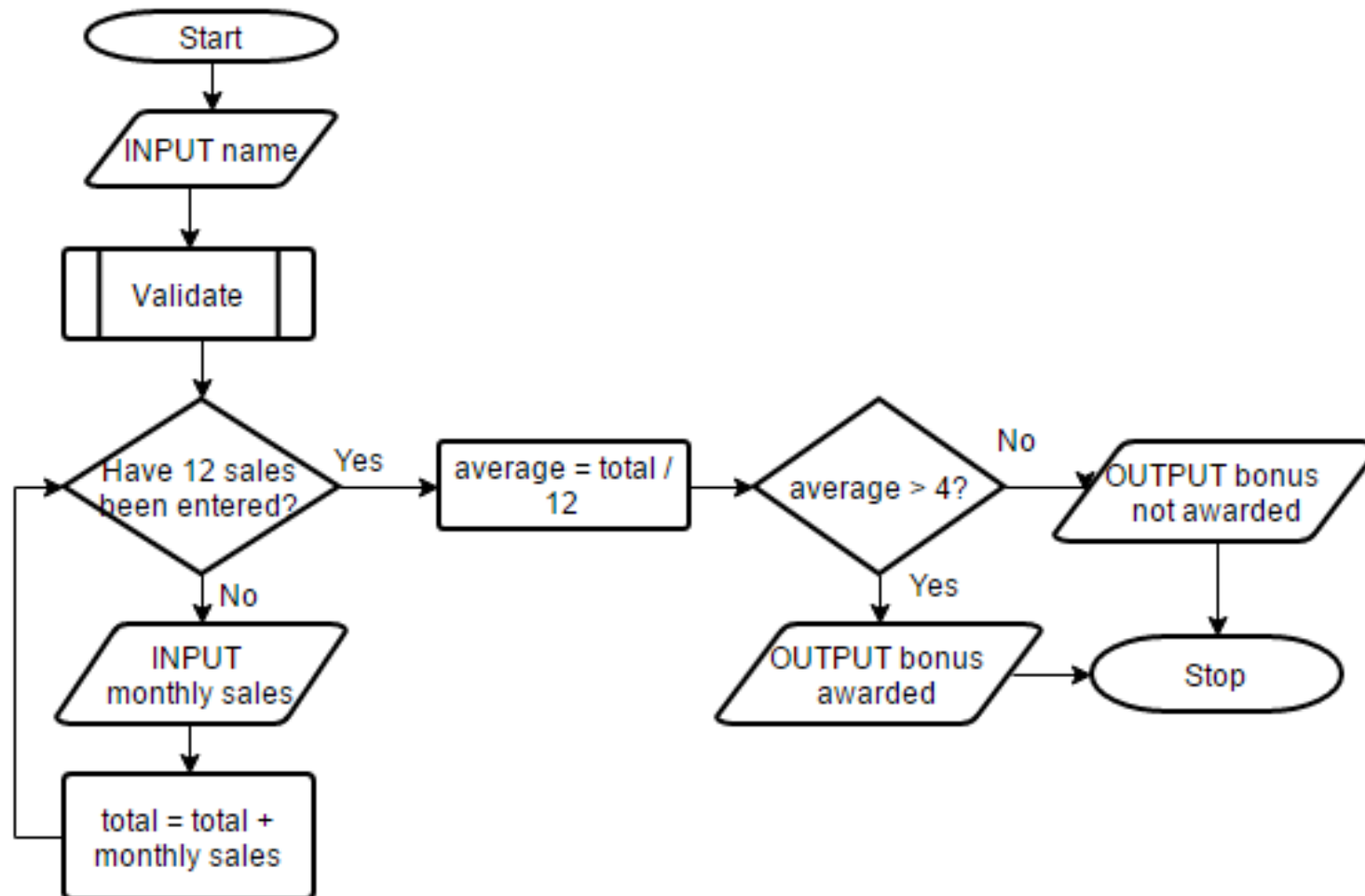
 OUTPUT "Bonus Awarded for " + name

ELSE

 OUTPUT "Bonus not awarded for " + name + ". Average Sales: " + average

ENDIF

Flowchart



Test Plan

Test Number	Test Type	Test Data	Reason	Expected Outcome
1	Valid	Name: Bob Sales: 3,4,5,3,3,3,4, 4,3,3,3,3	Enter a valid name with valid sales that average not enough for a bonus	It will say that a bonus is not awarded as the average is only 3.
2	Valid	Name: Sally Sales: 5,6,7,5,6,7,5, 6,7,5,6,7	Enter a valid name with valid sales that average enough for a bonus	It will say that a bonus is awarded for Sally.
3	Null Value	Name: nothing entered	To see if it repeatedly asks for a staff name when one isn't entered	It will keep saying enter the name of the staff member until one is entered.
4	Invalid	Name: Bob Sales: six	To see if the program will ask for the month 1 sales to be entered again until a numerical value is entered.	It will keep saying enter month 1 sales until a number is entered.

Development

- I first got the program to take the staff member name as an **input** and stored the string in a variable called name.

```
name = input("Enter the name of the staff member: ")
```

- To check that it stored it, I tested the program by adding print(name) underneath before moving on.

```
name = input("Enter the name of the staff member: ")  
print(name)
```



```
Enter the name of the staff member: Bob  
Bob
```

- I noticed that if I didn't enter a name the program continued (see below), therefore I added a while loop that will repeat while the length of the input is 0 i.e. blank.

```
Enter the name of the staff member:  
>>> |
```

Before

```
Enter the name of the staff member:  
Enter the name of the staff member:  
Enter the name of the staff member: Bob  
>>>
```

After

```
name = input("Enter the name of the staff member: ")  
while len(name)==0:  
    name = input("Enter the name of the staff member: ")
```

Development

- Next I added a for loop that repeats 12 times and inside the loop asked the user to enter the monthly sales (one for each month of the year). In the input message x+1 means that it will use the counter from the loop to display which month number.

```
for x in range(0,12):  
    sales = input("Enter the sales for month " + str(x+1) + ": ")
```

- I then tested the program to make sure it asked for 12 months of sales, which it did.
- To create the total, I added a variable at the start which set the total to 0 and then after each time a sales had been input added

```
total = 0  
for x in range(0,12):  
    sales = input("Enter the sales for month " + str(x+1) + ": ")  
    total = total + int(sales)
```

- At the end of the loop I printed total to make sure it was calculating correctly. I entered 6 for 6 months and 5 for the other 6 months, which should give a total of 66.

```
Enter the name of the staff member: Bob  
Enter the sales for month 1: 3  
Enter the sales for month 2: 4  
Enter the sales for month 3: 5  
Enter the sales for month 4: 6  
Enter the sales for month 5: 3  
Enter the sales for month 6: 4  
Enter the sales for month 7: 5  
Enter the sales for month 8: 6  
Enter the sales for month 9: 3  
Enter the sales for month 10: 4  
Enter the sales for month 11: 5  
Enter the sales for month 12: 6  
>>> |
```

```
Enter the name of the staff member: Bob  
Enter the sales for month 1: 6  
Enter the sales for month 2: 6  
Enter the sales for month 3: 6  
Enter the sales for month 4: 6  
Enter the sales for month 5: 6  
Enter the sales for month 6: 6  
Enter the sales for month 7: 5  
Enter the sales for month 8: 5  
Enter the sales for month 9: 5  
Enter the sales for month 10: 5  
Enter the sales for month 11: 5  
Enter the sales for month 12: 5  
66  
>>> |
```

Development

- At the end of the loop I printed total to make sure it was calculating correctly. I entered 6 for 6 months and 5 for the other 6 months, which should give a total of 66. I got this error.

```
Enter the name of the staff member: Bob
Enter the sales for month 1: 23
Traceback (most recent call last):
  File "I:\OGA\16-17\Admin\Training\Programming - Process of Success\Requirement
  1.py", line 7, in <module>
    total = total + sales
TypeError: unsupported operand type(s) for +: 'int' and 'str'
>>> |
```

```
total = total + int(sales)
```

- I realised that I forgot to convert sales to an integer as it was a string from the input, it then worked. The next stage was to work out the average and then see if this was more than 4, this was the easy bit. I divided the total by 12 to get the average and then used an IF to see if the average was more than 4.

```
average = total / 12
if average > 4:
    print("Bonus awarded for " + name)
else:
    print("Bonus not awarded for " + name + ".")
```

When I ran it, I saw that the average sales if the bonus was not awarded was to loads of decimal places. I rounded it to 2 decimal places.

```
Bonus not awarded for Bob. Average Sales 1.6666666666666667
Bonus not awarded for Bob. Average Sales 1.67
```

```
Enter the name of the staff member: Bob
Enter the sales for month 1: 6
Enter the sales for month 2: 6
Enter the sales for month 3: 6
Enter the sales for month 4: 6
Enter the sales for month 5: 6
Enter the sales for month 6: 6
Enter the sales for month 7: 5
Enter the sales for month 8: 5
Enter the sales for month 9: 5
Enter the sales for month 10: 5
Enter the sales for month 11: 5
Enter the sales for month 12: 5
66
>>> |
```


Annotated Code

```
#this takes input from the user and stores in a variable called name
name = input("Enter the name of the staff member: ")
#this is a while loop that will repeat while the length of name is 0 i.e. not entered
while len(name)==0:
    #this is the same as the first line of code but is inside the loop
    name = input("Enter the name of the staff member: ")
#this sets the total sales to 0
total = 0
#this is a for loop that repeats 12 times, for each month of the year
for x in range(0,12):
    #this takes input from the user for the sales and stores in a variable called sales
    #x+1 in the input means it will say what month we are looking for using the loop counter
    sales = input("Enter the sales for month " + str(x+1) + ": ")
    #this calculates the running total by adding the sales just entered to the previous total
    total = total + int(sales)
#after the loop, this calculates the average which is the total divided by 12 (the number of months)
average = total / 12
#this checks if the average is more than 4
if average > 4:
    #if it is then it will say bonus awarded and give the name of the staff member
    print("Bonus awarded for " + name)
else:
    #if it isn't then it will say bonus not awarded and what their average sales was
    print("Bonus not awarded for " + name + ". Average Sales " + str(round(average,2)))
```

Testing

Test Number	Test Type	Test Data	Reason	Expected Outcome	Actual Outcome	Pass/Fail?
1	Valid	Name: Bob Sales: 3,4,5,3,3,3,4,4,3,3,3,3	Enter a valid name with valid sales that average not enough for a bonus	It will say that a bonus is not awarded as the average is only 3.	It did say that a bonus is not awarded as the average is only 3.	Pass

```
Enter the name of the staff member: Bob
Enter the sales for month 1: 3
Enter the sales for month 2: 4
Enter the sales for month 3: 5
Enter the sales for month 4: 3
Enter the sales for month 5: 3
Enter the sales for month 6: 3
Enter the sales for month 7: 4
Enter the sales for month 8: 4
Enter the sales for month 9: 3
Enter the sales for month 10: 3
Enter the sales for month 11: 3
Enter the sales for month 12: 3
Bonus not awarded for Bob. Average Sales 3.0
```

Testing

Test Number	Test Type	Test Data	Reason	Expected Outcome	Actual Outcome	Pass/Fail?
2	Valid	Name: Sally Sales: 5,6,7,5,6,7,5,6,7,5,6,7	Enter a valid name with valid sales that average enough for a bonus	It will say that a bonus is awarded for Sally.	It did say that a bonus is awarded for Sally.	Pass

```
Enter the name of the staff member: Sally
Enter the sales for month 1: 5
Enter the sales for month 2: 6
Enter the sales for month 3: 7
Enter the sales for month 4: 5
Enter the sales for month 5: 6
Enter the sales for month 6: 7
Enter the sales for month 7: 5
Enter the sales for month 8: 6
Enter the sales for month 9: 7
Enter the sales for month 10: 5
Enter the sales for month 11: 6
Enter the sales for month 12: 7
Bonus awarded for Sally
```

Testing

Test Number	Test Type	Test Data	Reason	Expected Outcome	Actual Outcome	Pass/Fail?
3	Null Value	Name: nothing entered	To see if it repeatedly asks for a staff name when one isn't entered	It will keep saying enter the name of the staff member until one is entered.	It kept saying enter the name of the staff member until one is entered.	Pass

```
Enter the name of the staff member:
Enter the name of the staff member:
Enter the name of the staff member:
Enter the name of the staff member:
Enter the name of the staff member:
Enter the name of the staff member: |
```

Testing

Test Number	Test Type	Test Data	Reason	Expected Outcome	Actual Outcome	Pass/Fail?
4	Invalid	Name: Bob Sales: six	To see if the program will ask for the month 1 sales to be entered again until a numerical value is entered.	It will keep saying enter month 1 sales until a number is entered.	It errored saying it expected it a number	Fail

```
Enter the name of the staff member: Bob
Enter the sales for month 1: six
Traceback (most recent call last):
  File "I:\OGA\16-17\Admin\Training\Programming - Process of Success\Requirement
  1.py", line 15, in <module>
    total = total + int(sales)
ValueError: invalid literal for int() with base 10: 'six'
>>> |
```

```
Enter the name of the staff member: Bob
Enter the sales for month 1: six
Enter the sales for month 1: six
Enter the sales for month 1: 6
Enter the sales for month 2: |
```

```
while sales.isdigit()==False:
    sales = input("Enter the sales for month " + str(x+1) + ": ")
    x = x + 1
```

As the test failed, I added a while loop that checked if sales was a digit and will repeat while it isn't to stop the error happening. This fixed the problem, as you can see above.

Final Updated Code

```
#this takes input from the user and stores in a variable called name
name = input("Enter the name of the staff member: ")
#this is a while loop that will repeat while the length of name is 0 i.e. not entered
while len(name)==0:
    #this is the same as the first line of code but is inside the loop
    name = input("Enter the name of the staff member: ")
#this sets the total sales to 0
total = 0
#this is a for loop that repeats 12 times, for each month of the year
for x in range(0,12):
    #this takes input from the user for the sales and stores in a variable called sales
    #x+1 in the input means it will say what month we are looking for using the loop counter
    sales = input("Enter the sales for month " + str(x+1) + ": ")
    #a while loop that will check to see if sales is a digit and will repeat while its not
    while sales.isdigit()==False:
        #this takes input from the user for the sales and stores in a variable called sales
        sales = input("Enter the sales for month " + str(x+1) + ": ")
    #this calculates the running total by adding the sales just entered to the previous total
    total = total + int(sales)
#after the loop, this calculates the average which is the total divided by 12 (the number of months)
average = total / 12
#this checks if the average is more than 4
if average > 4:
    #if it is then it will say bonus awarded and give the name of the staff member
    print("Bonus awarded for " + name)
else:
    #if it isn't then it will say bonus not awarded and what their average sales was
    print("Bonus not awarded for " + name + ". Average Sales " + str(round(average,2)))
```

Evaluation

Success Criteria	How I met it	Evidence
Allow the user to input their name	I used the input command to display a message on screen so that the user entered their name	Slide 14 development
Allow the user to input their sales for each month	I used the input command again to display a message asking for the monthly sales. I put this inside a for loop so that it repeated and asked for the monthly sales 12 times.	Slide 14 development
Each month of sales are added together to get a total	I created a variable called total and after the sales were input it added the sales to the existing total	Slide 15 Development
Calculate the average sales based on the total sales	I created a variable called average that divided the total by 12 after the loop	Slide 16 Development
Check if the average sales are enough to get a bonus	I added an if statement that checked if the average was more than 4 which was what was needed to get a bonus.	Slide 16 Development
If they are output a message saying they are awarded with a bonus	I added a print message in the true part of the IF that said bonus awarded	Slide 19 Testing
If they aren't output a message saying they are not awarded a bonus	I added a print message in the false part of the IF that said the bonus wasn't awarded	Slide 18 Testing

Evaluation

- On top of the original success criteria that I came up with, I also got the program to deal with validation. I made sure that a name had to be entered at the start and also made sure that the numbers entered for sales were actual numbers rather than letters to prevent errors. This made my program more reliable.

Requirement 2

Make the program work so it works for multiple staff members

Success Criteria

- Build on the my program from the first requirement
- After it has calculated whether the first person should get a bonus ask if they want to check another staff member
- Re-run the program if they want to check another member of staff

Approach to be Used

- I will put my previous program inside a loop that will repeat while a value is true.
- After the sales for one person has been entered the program will ask the user if they want to enter another.

IPOD Table – Requirement 2

Input(s)	Process(es)	Output(s)	Decision(s)
Whether they want to run the program again			Does the user want to enter another set of sales

Variables to be Used – Requirement 2

Variable Name	Data Type	Purpose
Another	String	To store whether the user wants to add sales for another member of staff.

Possible Validation

Validation Type	Where	Reason
Format Check	When the user says whether they want to enter another	To make sure they enter a Y or an N, otherwise keep asking them to say whether they want to enter another again.

Pseudocode

Another="Y"

WHILE another == "Y"

 INPUT name

 Total = 0

 FOR loop to repeat 12 times

 INPUT monthly sales

 Total = total + monthly sales

 END FOR

 Average = total / 12

 If average > 4 then

 OUTPUT "Bonus Awarded for " + name

 ELSE

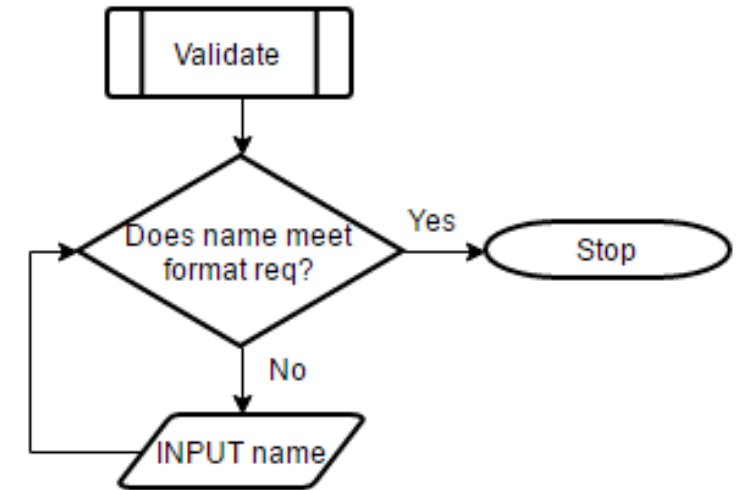
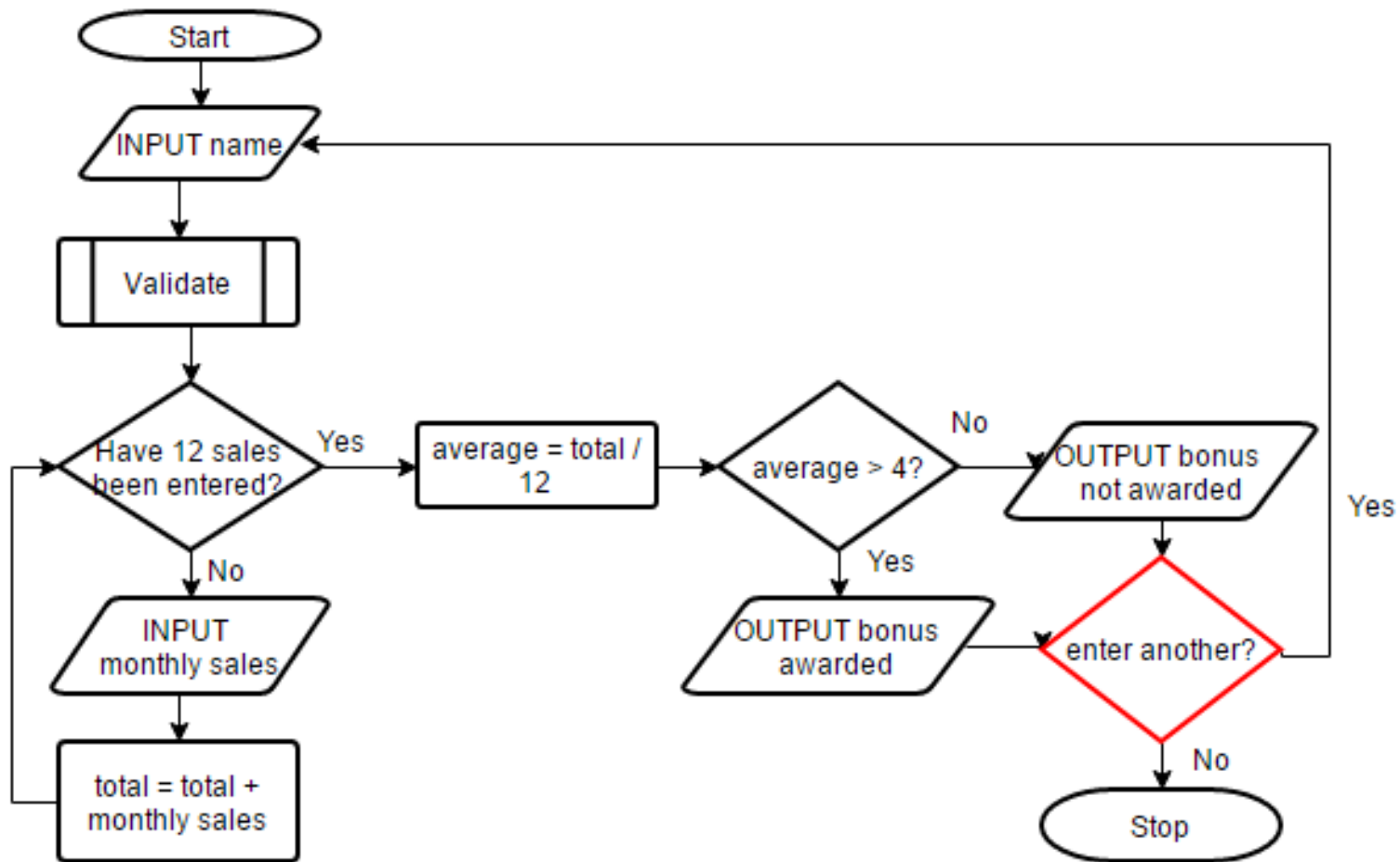
 OUTPUT "Bonus not awarded for " + name + ". Average Sales: " + average

 ENDIF

 Another = input("Would you like to enter another set of sales? Y/N ")

END WHILE

Flowchart



Test Plan

Test Number	Test Type	Test Data	Reason	Expected Outcome
1	Valid	Name: Bob Sales: 3,4,5,3,3,3,4, 4,3,3,3,3 Another = Y Name: Sally Sales: 5,6,7,5,6,7,5, 6,7,5,6,7	Say that you want to enter another member of staff to see if the program repeats	It will say that a bonus is not awarded as the average is only 3, and then for the second member will say the bonus is awarded for Sally.
2	Valid	Name: Sally Sales: 5,6,7,5,6,7,5, 6,7,5,6,7 Another = N	Say that you don't want to enter another member of staff	It should end the program after the first person's details are entered
3	Invalid	Name: Sally Sales: 5,6,7,5,6,7,5, 6,7,5,6,7 Another = HI	To see if the program handles another values that aren't Y or N	It should ask them to enter a choice until they enter Y or N
4	Invalid	Name: Sally Sales: 5,6,7,5,6,7,5, 6,7,5,6,7 Another = y	To see if the program handles a lower case Y and still repeats	The program should still repeat and ask for another staff name

Development

- There isn't much code that needs to be added for this requirement. I first added a variable called `another` that will keep track of whether the user wants to enter another and at the beginning set it to Y as they will want to enter at least 1 member of staff. I then added a while loop that said while `another` is Y and put the code from the last requirement inside it.
- This worked, but will just repeat forever, therefore at the end of the program still inside the loop, I added an input to ask the user if they wanted to enter another.

```
another = input("Would you like to enter another set of sales? Y/N ")
```

- This shows it working for N and Y.

```
Bonus not awarded for Bob. Average Sales 4.0
Would you like to enter another set of sales? Y/N N
>>> |
```

```
another="Y"
while another == "Y":
    #this takes input from
    name = input("Enter the
-
-
Enter the name of the staff member: Bob
Enter the sales for month 1: 2
Enter the sales for month 2: 3
Enter the sales for month 3: 4
Enter the sales for month 4: 2
Enter the sales for month 5: 3
Enter the sales for month 6: 4
Enter the sales for month 7: 2
Enter the sales for month 8: 3
Enter the sales for month 9: 4
Enter the sales for month 10: 2
Enter the sales for month 11: 3
Enter the sales for month 12: 4
Bonus not awarded for Bob. Average Sales 3.0
Enter the name of the staff member: Sally
Enter the sales for month 1: 3
```

```
Bonus not awarded for Bob. Average Sales 4.0
Would you like to enter another set of sales? Y/N Y
Enter the name of the staff member: |
```


Development

- I noticed that if I entered anything that was not Y it would stop the program, so I added a while loop that checked if it was not equal to Y or N. If it wasn't it would say invalid entry and get them to enter it again.

```
while another != "Y" and another != "N":  
    another = input("Invalid Entry. Would you like to enter another set of sales? Y/N ")
```

```
Bonus not awarded for Bob. Average Sales 4.0  
Would you like to enter another set of sales? Y/N hi  
>>>
```

```
Bonus not awarded for Bob. Average Sales 4.0  
Would you like to enter another set of sales? Y/N Hi  
Invalid Entry. Would you like to enter another set of sales? Y/N Hi  
Invalid Entry. Would you like to enter another set of sales? Y/N N  
>>> |
```

- My code was getting quite large at this point so I decided to put the first requirement into a function to separate it up. I defined a function and put the code inside. I then called the function sales() where the code previous was. My main program was now only a few lines big.

```
def sales():  
    #this takes input from th  
    name = input("Enter t  
    #this is a while loop  
    while len(name)==0:
```

```
#Main program  
#this set a variable called  
another="Y"  
#this is a while loop that  
while another == "Y":  
    sales()  
    another = input("Would  
    #this is a loop to chec
```

Annotated Code – Sales Function

```
#function to calculate average sales of a staff member
def sales():
    #this takes input from the user and stores in a variable called name
    name = input("Enter the name of the staff member: ")
    #this is a while loop that will repeat while the length of name is 0 i.e. not entered
    while len(name)==0:
        #this is the same as the first line of code but is inside the loop
        name = input("Enter the name of the staff member: ")
    #this sets the total sales to 0
    total = 0
    #this is a for loop that repeats 12 times, for each month of the year
    for x in range(0,12):
        #this takes input from the user for the sales and stores in a variable called sales
        #x+1 in the input means it will say what month we are looking for using the loop counter
        sales = input("Enter the sales for month " + str(x+1) + ": ")
        #a while loop that will check to see if sales is a digit and will repeat while its not
        while sales.isdigit()==False:
            #this takes input from the user for the sales and stores in a variable called sales
            sales = input("Enter the sales for month " + str(x+1) + ": ")
        #this calculates the running total by adding the sales just entered to the previous total
        total = total + int(sales)
    #after the loop, this calculates the average which is the total divided by 12 (the number of months)
    average = total / 12
    #this checks if the average is more than 4
    if average > 4:
        #if it is then it will say bonus awarded and give the name of the staff member
        print("Bonus awarded for " + name)
    else:
        #if it isn't then it will say bonus not awarded and what their average sales was
        print("Bonus not awarded for " + name + ". Average Sales " + str(round(average,2)))
    #this input asks whether the user wants to enter more sales and start the process again
```

Annotated Code – Main Program

```
#Main program
#this set a variable called another to Y which keeps track of whether to keep taking more staff sales
another="Y"
#this is a while loop that will repeat while the user still wants to enter more staff details
while another == "Y":
    #calls the function sales
    sales()
    another = input("Would you like to enter another set of sales? Y/N ")
    #this is a loop to check if the user has entered a Y or an N
    while another != "Y" and another != "N":
        #if they haven't it will ask them to reenter again
        another = input("Invalid Entry. Would you like to enter another set of sales? Y/N ")
```

Testing

Test Number	Test Type	Test Data	Reason	Expected Outcome	Actual Outcome	Pass/Fail?
1	Valid	Name: Bob Sales: 3,4,5,3,3,3,4, 4,3,3,3,3 Another = Y Name: Sally Sales: 5,6,7,5,6,7,5, 6,7,5,6,7	Say that you want to enter another member of staff to see if the program repeats	It will say that a bonus is not awarded as the average is only 3, and then for the second member will say the bonus is awarded for Sally.	It repeated and asked for the second set of information to show the loop works.	Pass

```

Enter the name of the staff member: Bob
Enter the sales for month 1: 3
Enter the sales for month 2: 4
Enter the sales for month 3: 5
Enter the sales for month 4: 3
Enter the sales for month 5: 3
Enter the sales for month 6: 3
Enter the sales for month 7: 4
Enter the sales for month 8: 4
Enter the sales for month 9: 3
Enter the sales for month 10: 3
Enter the sales for month 11: 3
Enter the sales for month 12: 3
Bonus not awarded for Bob. Average Sales 3.42
Would you like to enter another set of sales? Y/N Y
Enter the name of the staff member: Sally
  
```

```

Enter the sales for month 1: 5
Enter the sales for month 2: 6
Enter the sales for month 3: 7
Enter the sales for month 4: 5
Enter the sales for month 5: 6
Enter the sales for month 6: 7
Enter the sales for month 7: 5
Enter the sales for month 8: 6
Enter the sales for month 9: 7
Enter the sales for month 10: 5
Enter the sales for month 11: 6
Enter the sales for month 12: 7
Bonus awarded for Sally
Would you like to enter another set of sales? Y/N
  
```

Testing

Test Number	Test Type	Test Data	Reason	Expected Outcome	Actual Outcome	Pass/Fail?
2	Valid	Name: Sally Sales: 5,6,7,5,6,7,5, 6,7,5,6,7 Another = N	Say that you don't want to enter another member of staff	It should end the program after the first person's details are entered	It did end after N was entered for repeating the process	Pass

```
Enter the name of the staff member: Sally
Enter the sales for month 1: 5
Enter the sales for month 2: 6
Enter the sales for month 3: 7
Enter the sales for month 4: 5
Enter the sales for month 5: 6
Enter the sales for month 6: 7
Enter the sales for month 7: 5
Enter the sales for month 8: 6
Enter the sales for month 9: 7
Enter the sales for month 10: 5
Enter the sales for month 11: 6
Enter the sales for month 12: 7
Bonus awarded for Sally
Would you like to enter another set of sales? Y/N N
>>> |
```

Testing

Test Number	Test Type	Test Data	Reason	Expected Outcome	Actual Outcome	Pass/Fail?
3	Invalid	Name: Sally Sales: 5,6,7,5,6,7,5, 6,7,5,6,7 Another = HI	To see if the program handles another values that aren't Y or N	It should ask them to enter a choice until they enter Y or N	It kept asking for a valid input until a Y or N was entered	Fail

```
Enter the name of the staff member: Sally
Enter the sales for month 1: 5
Enter the sales for month 2: 6
Enter the sales for month 3: 7
Enter the sales for month 4: 5
Enter the sales for month 5: 6
Enter the sales for month 6: 7
Enter the sales for month 7: 5
Enter the sales for month 8: 6
Enter the sales for month 9: 7
Enter the sales for month 10: 5
Enter the sales for month 11: 6
Enter the sales for month 12: 7
Bonus awarded for Sally
Would you like to enter another set of sales? Y/N HI
Invalid Entry. Would you like to enter another set of sales? Y/N HI
Invalid Entry. Would you like to enter another set of sales? Y/N N
>>> |
```

Testing

Test Number	Test Type	Test Data	Reason	Expected Outcome	Actual Outcome	Pass/Fail?
4	Invalid	Name: Sally Sales: 5,6,7,5,6,7,5, 6,7,5,6,7 Another = y	To see if the program handles a lower case Y and still repeats	The program should still repeat and ask for another staff name	The program said invalid entry, even though Y is valid	Fail

```
Enter the sales for month 12: 7
Bonus awarded for Sally
Would you like to enter another set of sales? Y/N y
Invalid Entry. Would you like to enter another set of sales? Y/N |
```

Before

```
Enter the sales for month 12: 7
Bonus awarded for Sally
Would you like to enter another set of sales? Y/N y
Enter the name of the staff member: |
```

After

```
another = input("Would you like to enter another set of sales? Y/N ")
another = another.upper()

another = input("Invalid Entry. Would you like to enter another set of sales? Y/N ")
another = another.upper()
```

To fix this issue I converted what was entered to upper case before it checked if it was a valid entry this sorted the problem and the test now passed.

Final Updated Code

```
#Main program
#this set a variable called another to Y which keeps track of whether to keep taking more staff sales
another="Y"
#this is a while loop that will repeat while the user still wants to enter more staff details
while another == "Y":
    #calls the function sales
    sales()
    another = input("Would you like to enter another set of sales? Y/N ")
    #this converts what has been entered by the user into upper case
    another = another.upper()
    #this is a loop to check if the user has entered a Y or an N
    while another != "Y" and another != "N":
        #if they haven't it will ask them to reenter again
        another = input("Invalid Entry. Would you like to enter another set of sales? Y/N ")
        #this converts what has been entered by the user into upper case
        another = another.upper()
```

The sales function was not changed during testing, the main program was.

Evaluation

Success Criteria	How I met it	Evidence
Build on the my program from the first requirement	I used my program from the previous task and added to it. In the end I decided to put the first requirement in a function to make it easier to follow	Slide 34 & 35 Code
After it has calculated whether the first person should get a bonus ask if they want to check another staff member	I asked the user using input whether they want to enter more sales and stored their response in a variable called another.	Slide 36 Testing
Re-run the program if they want to check another member of staff	I used a variable to store whether the user wanted to enter more sales and then inserted the previous code in a while loop that would repeat while they wanted to enter more	Slide 36 Testing

I also add validation to the new input from the user so that anything they entered was converted to uppercase as I was looking for a Y or an N. I also made sure that if they entered anything other than a Y or an N it asked them to enter their response again.