**Create a simple computer program to meet a set brief**

US18740 Version 6 Level 2 Credits 3

**Plan**

**For: Mr Kneed**

**Purpose: To determine store success and create a program capable of automating calculations of sales information on stores to be reported back to Mr Kneed as a sales summary.**

**Computer Language:** Python 3.7

**Flow Chart:**

**Diagram, schematic

Description automatically generated**

**Input**: Must be an integer or float

**Output:** cash register slip

**Layout (Pseudocode):**

Def printCenteredText(textToCentre, lineSpacesAfterText = 0, lineSpacesBeforeText = 0):

If lineSpacesBeforeText is greater than 0: print empty lines

Print the text centred in the console

If lineSpacesAfterText is greater than 0: print empty lines

Def printOneCharacterAcrossTerminal(character):

Print the character all across the terminal

Def addStores():

While forever:

While the input from user is not in Yes and not in No:

Ask for user input

Make user input lowercase and get rid of whitespace

If userInput is not in Yes and not in No:

Tell the user they must enter Yes or No

If userInput is in Yes:

Ask user for store inputs

If user input is wrong:

Tell the user they entered wrong input

For items in a dummy store items variable:

Ask the user if they want to add the store

If they do:

Add the store

Otherwise:

Don’t add the store

If there are no stores:

Tell the user they must have stores to continue

Def addStoreInformation(listOfStores):

For stores in list:

For days in daysInAWeek:

Try:

Ask for revenue for that store

Except:

Tell the user that they entered wrong input

Ask the user how they want the stores to be sorted

While there are incorrect stores:

For store in storeNames:

Try:

Get store value

Except:

Add store to incorrect stores

If there is more than one incorrect store:

Add store

For the amount of stores:

Calculate total, gst, gst inclusive, max, best day, min, worst day, mean, and average

Print information

If there is more than one store:

If store revenues are the same:

Tell the user all stores performed equally

Otherwise:

Calculate best store and worst store

Print best and worst stores

Print farewell

**Functions:**

printCenteredText(textToCentre, lineSpacesAfterText = 0, lineSpacesBeforeText = 0)

printOneCharacterAcrossTerminal(character)

addStoreInformation(listToUse)

Confirmation()

**Procedures:**

main()

Start of the program.

Define variables and functions

Print Welcome

Get user to add the stores they want to be added

Ask the user if they want the stores to be alphabetically ordered

Get the user to add store information

Calculate store total, gst, gstInclusive, max, bestDay, min, worstDay, mean, and average

Sort the stores depending on how the user wanted it

Print store information

Print best and worst store if there is more than 1 store

**Lists:**

storeNames

finalStoreNamesList

daysInAWeek

incorrectStores

salesNumbersList

stores

storeTotalSales

storeTotalSalesNames

lineToPrint

Dummy\_Stores

**Variables:**

printAlphabeticalOrder

userInput

storeNameInput

confirmAddStoreInput

correctStore

salesNumbers

confirmSalesNumbersAreCorrect

total

gst

gstInclusive

maxValue

bestDay

minValue

worstDay

mean

average

bestPerformingSales

bestPerformingStore

worstPerformingSales

worstPerformingStore

**Stakeholder consultations:** *(at least 3)*

**Kovid Dev – If the user enters xxx, should I make the program only move onto the next step when entering store names and ignore xxx when doing anything else or should the program move onto the next step whenever xxx is entered**

* Only when the user is entering store names

**Andrew Gossen – If the user gets rid of all of their stores, should I display a “bankrupt sequence” or should I do something else**

* You shouldn’t allow them to have no stores

**Bradley – Should I display a message saying “If you want to skip store input enter xxx” when the user is entering store names**

* **Yes**

**Milestones:** *(at least 3)*

* Saturday 14th May 7:18PM: Store inputs have been added
* Saturday 14th May 7:26PM: Fixed bugs that wouldn’t make the while loop continue on when a user entered a wrong value by setting variable back to default value after the user entered a wrong value so the while loop conditions would be met and restart the loop
* Saturday 14th May 7:34PM: Added for loop to ask the user if all the stores in the storesList are correct
* Saturday 14th May 8PM: You can now record store revenue
* Saturday 14th May 8:06PM: All of the metrics for the store get calculated (average, mean, worst day etc.) and put into a stores dictionary
* Saturday 14th May 8:17PM: Stores now get sorted into alphabetical order or descending alphabetical order depending on what the user wanted
* Saturday 14th May 8:20PM: All stores’ values now get printed
* Saturday 14th May 8:35PM: Now a user gets asked if the sales numbers for a store are correct so if they entered it wrong they can enter the values again and make sure they are correct
* Saturday 14th 8:40PM: Calculate the best store revenue and worst store revenue and print it out
* Saturday 14th 8:43PM: If a user says all stores are incorrect so there are no more stores in the storeNames list, “bankrupt sequence” gets initiated and the program ends.
* Friday 20th May 2:34PM: Allow entering floats and ints for store revenues instead of just ints
* Friday 20th May 2:37PM: Use main() function
* Monday 23rd May 8:35PM: I made the program more user friendly by accepting both y, yes, n, no, and non-case sensitive variations of those; I further tested this in my program as shown in the image in Test 3
* Monday 23rd May 9:51PM: Got rid of “bankrupt sequence” and now the user is forced to have at least 1 store before continuing
* Monday 23rd May 9:53PM: If the user enters ‘xxx’ while entering store names, the program moves onto letting the user enter store revenue

**Testing procedures:**

Testing procedures that I used are:

* Example data from teacher and seeing if results from my program match up with the teacher’s results
* Putting in random data and using a calculator to see if the results match up with the calculators results

**Resources used:** *(at least 2)*

Documents are help provided by the teacher

Python – 3.7

www.stackoverflow.com

**Testing:**

**Test 1 – Testing if store inputs work and if the user enters a wrong value they have to re-enter a correct value**

**What happened: If the user entered a wrong input the loop would stop because its conditions were not properly met so the code would move on even though the user hasn’t entered the values that they need to**

**How I fixed this: I set the variable back to its original value when someone enters a wrong value so the loop conditions are met**

**Test 2 – Testing if printing out store statistics work**

**What happened: print(f’Statistic name here-${value[‘statistic name here]}:' gave error ‘Unterminated expression in f-string; missing close brace’**

**How I fixed this: I set a variable with the name of the statistic to value[‘statistic name’] and then printed that variable**

**Test 3**

Text

Description automatically generated

Inputs:

yeS

y

YeS

Y

yEs