|  |  |
| --- | --- |
| A picture of a winding road and trees  Shopping List  Names | Abstract  ***A shopping list for items purchased online is represented by a software. The customer's name, address, goods purchased, and overall price should all be recorded on the shopping list. Create a different list for each customer, then use the keyboard to input the list information for five customers.***  Name  Programming |

Contents

[Introduction 2](#_Toc70122178)

[Algorithm Design 2](#_Toc70122179)

[Customers\_entry() 2](#_Toc70122180)

[**Main()** 2](#_Toc70122181)

[**Customers\_cost(shopping\_list)** 2](#_Toc70122182)

[**Write\_to\_file ()** 2](#_Toc70122183)

[FlowChart 3](#_Toc70122184)

[Implementation 4](#_Toc70122185)

[Summary 5](#_Toc70122186)

# Introduction

A shopping list for items purchased online is represented by a software. The customer's name, address, goods purchased, and overall price should all be recorded on the shopping list. Create a different list for each customer, then use the keyboard to input the list information for five customers.

You can create functions to measure the net cost for all consumers within the same package or by importing a module. In a file named "shopping.txt," write the specifics for each shopping list on separate lines.

# Algorithm Design

The design was achieved by making a separate module for inputting all customers data the **module contains:**

**Customers\_entry()**

To add each customer's information, such as name, address, objects, and complete, to a list provided within this feature, and then to the main program's main list.

**Main program contains:**

## **Main()**

To do what has been asked of us, we use all of the previous tasks in the correct order.

## **Customers\_cost(shopping\_list)**

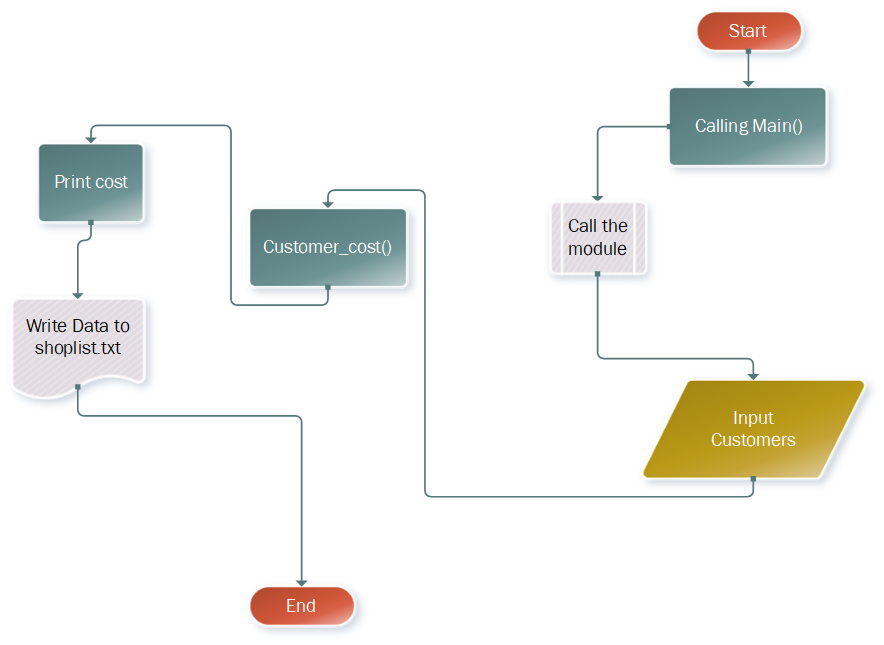
This function calculates total cost by iterating over all of the data in the shopping list and focusing on the fourth variable, which contains the price value.

## **Write\_to\_file ()**

We use python file methods to write the records, making sure to write and customer list line by line in the file shoplist.txt.

# FlowChart

This is the flowchart that demonstrate the workflow:



# Implementation

The input Module:

filename = "shopping.txt"  
  
  
def customers\_entry():  
 customers\_count = int(input('Enter the Number of Customers you want to add ? '))  
 customers\_count += 1  
 customers = []  
 c = []  
 for i in range(1, customers\_count, 1):  
 name = input('input customer name: ')  
 address = input('input customer address: ')  
 items = int(input('input items bought: '))  
 total = float(input('input total\_price: '))  
 c.append(name)  
 c.append(address)  
 c.append(items)  
 c.append(total)  
 customers.append(c)  
 # reset list object  
 c = []  
 print('\n')  
  
 return customers

**The main python file**

from helper\_module import \*  
  
  
# calculate the total cost function  
def customers\_cost(shopping\_list):  
 s = 0  
 for i in range(len(shopping\_list)):  
 s += shopping\_list[i][3]  
 return s  
  
  
# write the data to file  
def write\_to\_file(data):  
 try:  
 write\_file = open(filename, "a")  
 for i in range(len(data)):  
 write\_file.write('%-10s %30s %18d %18.2f\n'  
 % (data[i][0], data[i][1], data[i][2], data[i][3]))  
 except IOError:  
 print("not found")  
 finally:  
 write\_file.close()  
  
  
# call our functions in 1 method  
def main\_method():  
 shop\_list = customers\_entry()  
 cost = customers\_cost(shop\_list)  
 print('The Cost of customers entered:', cost)  
 write\_to\_file(shop\_list)  
  
  
# run the program in one call  
main\_method()

# Summary

The software demonstrates a well-organized method of implementing what the creator was required to do by using the programming style by dividing the problem into small modules as functions and then using them in a sequential order to produce the corresponding and valid output that meets our requirements.