# **Program 1: Food Delivery Program  Report**

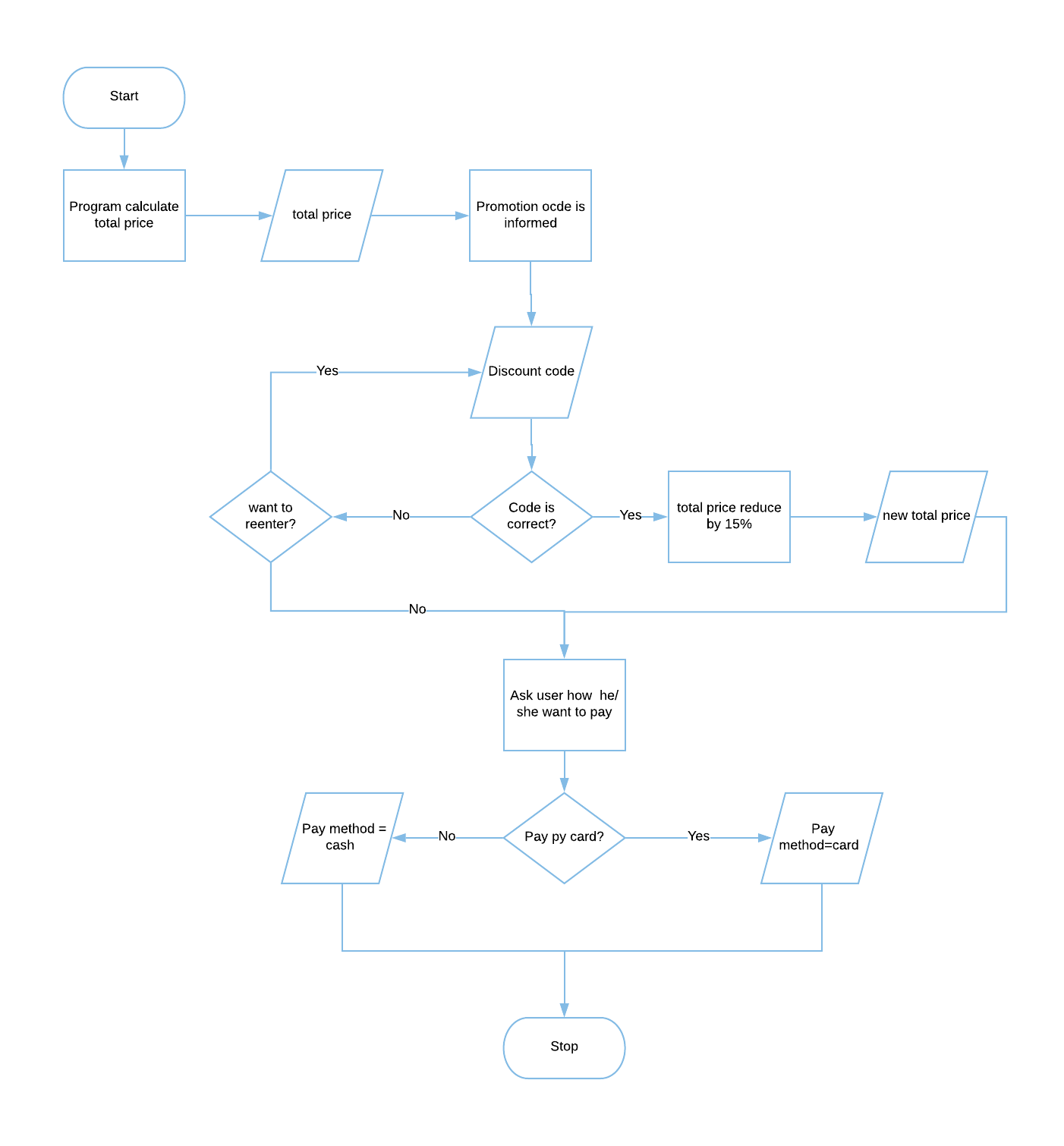
# Food delivery program flow chart







# Subprocess: Payment



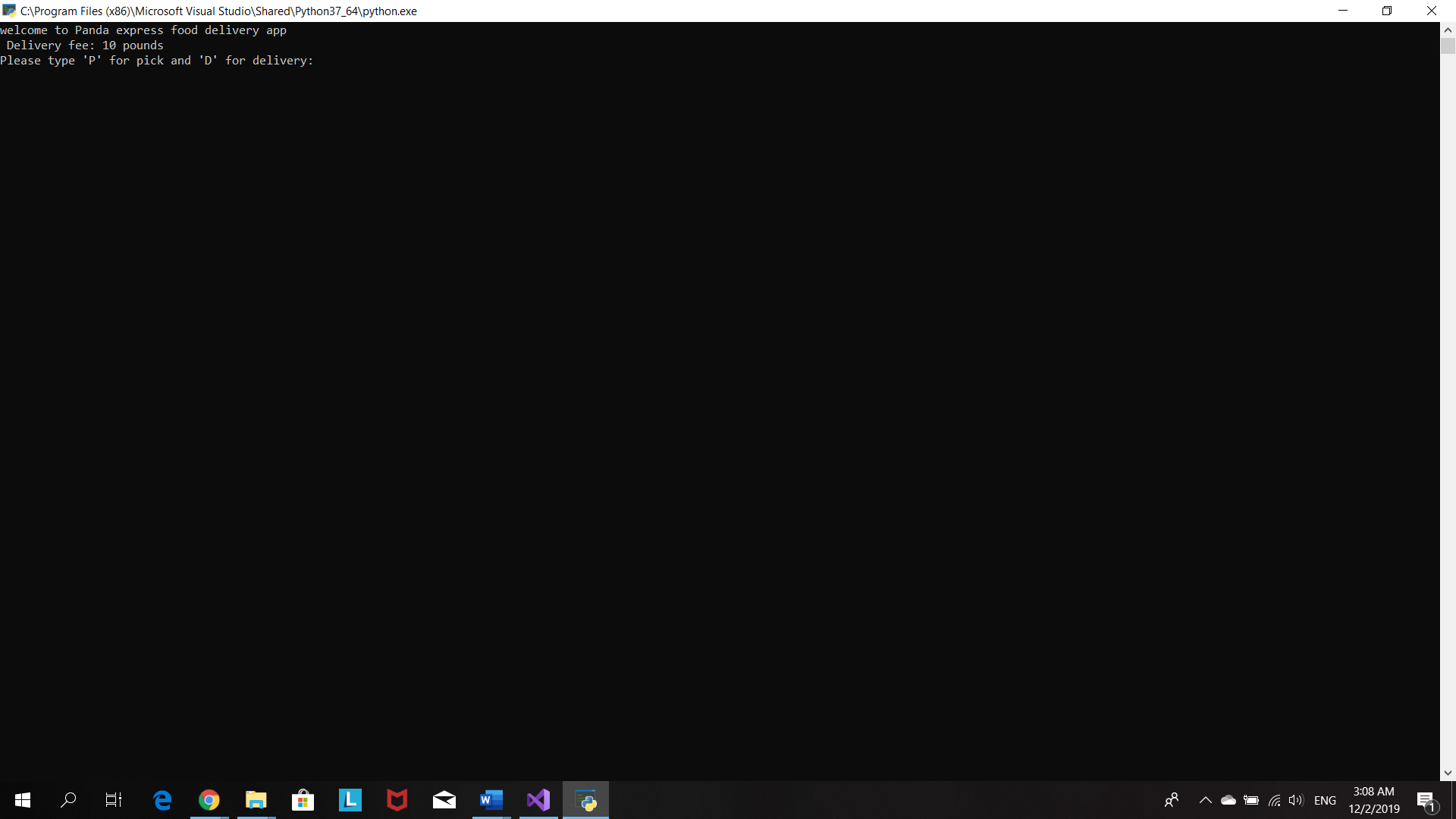
# Testing Table- Program 1: Food Delivery Program

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No. | Item to test | Test description | Test input | Expected result | Actual result | Comment |
| 1 | input("Please type 'P' for  pick and 'D' for delivery: ") | Typical valid data | P | Value accepted | Value accepted-program continue as normal | - |
| 2 | input("Please type 'P' for  pick and 'D' for delivery: ") | Extreme valid data | d | Value accepted | Value accepted-program continue as normal | - |
| 3 | input("Please type 'P' for  pick and 'D' for delivery: ") | Invalid data | T | Error massage: re-enter value | Value accepted -program continue but skip the rest of function pickupDelivery | Should let user input a correct data again |
| 4 | input("Please type 'P' for  pick and 'D' for delivery: ") | Invalid extreme data | i | Error massage: re-enter value | Value accepted -program continue but skip the rest of function pickupDelivery | Should let user input a correct value again |
| 5 | input("Please type 'P' for  pick and 'D' for delivery: ") | Erroneous data | 100 | Error massage: re-enter value | Value accepted -program continue but skip the rest of function pickupDelivery | Should let user input a correct value again |
| 6 | input("Please type 'P' for  pick and 'D' for delivery: ") | Erroneous data | $%^&(() | Error massage: re-enter value | Value accepted -program continue but skip the rest of function pickupDelivery | Should let user input a correct value again |
| 7 | input("Costumer Name: ") | Typical valid data | Jay | Value accepted | Value accepted-program continue as normal | - |
| 8 | input("Costumer Name: ") | Extreme valid data | ppppppppppppppppppp | Value accepted | Value accepted-program continue as normal |  |
| 9 | input("Costumer Name: ") | Invalid data | 10 | Error massage: re-enter value | Value accepted-program continue as normal | Should let user input a correct value again |
| 10 | input("Costumer Name: ") | Invalid extreme data | -810 | Error massage: re-enter value | Value accepted-program continue as normal | Should let user input a correct value again |
| 11 | input("Costumer Name: ") | Erroneous data | $%^&(() | Error massage: re-enter value | Value accepted-program continue as normal | Should let user input a correct value again |
| 12 | input("Costumer Name: ") | Erroneous data | “HELP” | Error massage:  Re-enter value | Value accepted-program continue as normal | Should let user input a correct value again |
| 13 | input("Please enter you house number: ") | Typical valid data | 211 | Value accepted | Value accepted-program continue as normal | - |
| 14 | input("Please enter you house number: ") | Extreme valid data | 2000000 | Error massage: re-enter value | Value accepted-program continue as normal | Limit number of house number |
| 15 | input("Please enter you house number: ") | Invalid data | T | Error massage: re-enter value | Value accepted-program continue as normal | Allow user to input only number |
| 16 | input("Please enter you house number: ") | Extreme valid data | pppppppppppppppppppppppppppppppp | Error massage: re-enter value | Value accepted-program continue as normal | Allow user to input only number |
| 17 | input("Please enter you house number: ") | Erroneous data | $%^& | Error massage: re-enter value | Value accepted-program continue as normal | Allow user to input only number |
| 18 | input("Please enter you house number: ") | Erroneous data | “HELP!!!” | Error massage: re-enter value | Value accepted-program continue as normal | Allow user to input only number |
| 19 | street=input("Please enter street: ") | Typical valid data | Baker St | Value accepted | Value accepted-program continue as normal | - |
| 20 | street=input("Please enter street: ") | Extreme valid data | qwertyuiopdsfvhvjsdjsd St | Error massage: re-enter value | Value accepted-program continue as normal | Limit number or input character |
| 21 | street=input("Please enter street: ") | Invalid data | 10 | Error massage: re-enter value | Value accepted-program continue as normal | Allow user to input only letter and dot |
| 22 | street=input("Please enter street: ") | Invalid extreme data | -785 | Error massage: re-enter value | Value accepted-program continue as normal | Allow user to input only letter and dot |
| 23 | street=input("Please enter street: ") | Erroneous data | $£%^&\* | Error massage: re-enter value | Value accepted-program continue as normal | Allow user to input only letter and dot |
| 24 | street=input("Please enter street: ") | Erroneous data | “Sherlock Holmes wow!!!!” | Error massage: re-enter value | Value accepted-program continue as normal | Allow user to input only letter and dot |
| 25 | postcode=input("Please  enter postcode: ") | Typical valid data | NW1 6XE | Value accepted | Value accepted-program continue as normal | - |
| 26 | postcode=input("Please  enter postcode: ") | Extreme valid data | NW120 6XE | Error massage: re-enter value | Value accepted-program continue as normal | Limit amount of character can be inputted |
| 27 | postcode=input("Please  enter postcode: ") | Invalid data | 64545 | Error massage: re-enter value | Value accepted-program continue as normal | Should let user input a correct value again |
| 28 | postcode=input("Please  enter postcode: ") | Invalid extreme data | --888888 | Error massage: re-enter value | Value accepted-program continue as normal | Limit amount of character can be inputted |
| 29 | postcode=input("Please  enter postcode: ") | Erroneous data | $%^% | Error massage: re-enter value | Value accepted-program continue as normal | Should let user input a correct value again |
| 30 | postcode=input("Please  enter postcode: ") | Erroneous data | £$%^&\*(093-406834-0683 | Error massage: re-enter value | Value accepted-program continue as normal | Limit amount of character can be inputted |
| 31 | input("Do you want your order to deliver in specific time? (Y/N): ") | Typical valid data | Y | Value accepted | Value accepted - program continue as normal | - |
| 32 | input("Do you want your order to deliver in specific time? (Y/N): ") | Extreme valid data | nnnn | Error massage: re-enter value | Error massage – program let user re-enter the value | - |
| 33 | input("Do you want your order to deliver in specific time? (Y/N): ") | Invalid data | q | Error massage: re-enter value | Error massage – program let user re-enter the value | Allow user to re-enter the correct choice |
| 34 | input("Do you want your order to deliver in specific time? (Y/N): ") | Extreme valid data | tttttt | Error massage: re-enter value | Error massage – program let user re-enter the value | Allow user to re-enter the correct choice |
| 35 | input("Do you want your order to deliver in specific time? (Y/N): ") | Erroneous data | 50 | Error massage: re-enter value | Error massage – program let user re-enter the value | Limit the input, allow user to input only y and n |
| 36 | input("Do you want your order to deliver in specific time? (Y/N): ") | Erroneous data | -\*32534534 | Error massage: re-enter value | Error massage – program let user re-enter the value | Limit the input, allow user to input only y and n |
| 37 | input("What time do you want your order to be deliver") | Typical valid data | 14:00 | Value accepted | Value accepted - program continue as normal | - |
| 38 | input("What time do you want your order to be deliver") | Extreme valid data | 180:00 | Error massage: re-enter value | Value accepted - program continue as normal | Limit range of number that can be inputted |
| 39 | input("What time do you want your order to be deliver") | Invalid data | e | Error massage: re-enter value | Value accepted - program continue as normal | Allow only number to be inputted |
| 40 | input("What time do you want your order to be deliver") | Extreme valid data | ttttttttttt | Error massage: re-enter value | Value accepted - program continue as normal | Allow only number to be inputted |
| 41 | input("What time do you want your order to be deliver") | Erroneous data | (1\*50) | Error massage: re-enter value | Value accepted - program continue as normal | Limit range of number that can be inputted |
| 42 | input("What time do you want your order to be deliver") | Erroneous data | print(“hi”) | Error massage: re-enter value | Value accepted - program continue as normal | Allow only number to be inputted |
| 43 | input("Confirm Information (Y/N):") | Typical valid data | Y | Value accepted | Value accepted - program continue as normal | - |
| 44 | input("Confirm Information (Y/N):") | Extreme valid data | nnnn | Error massage: re-enter value | Error massage – program let user re-enter the value | - |
| 45 | input("Confirm Information (Y/N):") | Invalid data | q | Error massage: re-enter value | Error massage – program let user re-enter the value | - |
| 46 | input("Confirm Information (Y/N):") | Extreme valid data | tttttt | Error massage: re-enter value | Error massage – program let user re-enter the value | - |
| 47 | input("Confirm Information (Y/N):") | Erroneous data | 50 | Error massage: re-enter value | Error massage – program let user re-enter the value | Allow user to input only y and n |
| 48 | input("Confirm Information (Y/N):") | Erroneous data | -\*32534534 | Error massage: re-enter value | Error massage – program let user re-enter the value | Allow user to input only y and n |
| 49 | int(input("How many plate you want to order : ")) | Typical valid data | 1 | Value accepted | Value accepted - program continue as normal | - |
| 50 | int(input("How many plate you want to order : ")) | Extreme valid data | 100 | Value accepted | Value accepted - program continue as normal | - |
| 51 | int(input("How many plate you want to order : ")) | Invalid data | -1 | Error massage: re-enter value | program do not crash but the program doesn’t run as normal and no error massage | Limit range of number that can be inputted |
| 52 | int(input("How many plate you want to order : ")) | Extreme valid data | -789 | Error massage: re-enter value | program do not crash but the program doesn’t continue as normal and no error massage | Limit range of number that can be inputted |
| 53 | int(input("How many plate you want to order : ")) | Erroneous data | P | Error massage: re-enter value | Program crashes | Limit range of number that can be inputted |
| 54 | int(input("How many plate you want to order : ")) | Erroneous data | @$% | Error massage: re-enter value | Program crashes | Limit range of number that can be inputted |
| 55 | int(input("How many entress you want(minimum 2 maximum 5 entress): ")) | Typical valid data | 3 | Value accepted | Value accepted - program continue as normal | - |
| 56 | int(input("How many entress you want(minimum 2 maximum 5 entress): ")) | Extreme valid data | 70 | Error massage: re-enter value | Error massage – program let user re-enter the value | - |
| 57 | int(input("How many entress you want(minimum 2 maximum 5 entress): ")) | Invalid data | -3 | Error massage: re-enter value | Error massage – program let user re-enter the value | - |
| 58 | int(input("How many entress you want(minimum 2 maximum 5 entress): ")) | Extreme valid data | -70 | Error massage: re-enter value | Error massage – program let user re-enter the value | - |
| 59 | int(input("How many entress you want(minimum 2 maximum 5 entress): ")) | Erroneous data | y | Error massage: re-enter value | Program crashes | Error massage – program let user re-enter the value |
| 60 | int(input("How many entress you want(minimum 2 maximum 5 entress): ")) | Erroneous data | $#t | Error massage: re-enter value | Program crashes | Error massage – program let user re-enter the value |
| 61 | int(input("Please select  the entree: ")) | Typical valid data | 2 | Value accepted | Value accepted - program continue as normal | - |
| 62 | int(input("Please select  the entree: ")) | Extreme valid data | 20 | Error massage: re-enter value | Error massage – program let user re-enter the value | - |
| 63 | int(input("Please select  the entree: ")) | Invalid data | -2 | Error massage: re-enter value | Error massage – program let user re-enter the value | - |
| 64 | int(input("Please select  the entree: ")) | Extreme valid data | -20 | Error massage: re-enter value | Error massage – program let user re-enter the value | - |
| 65 | int(input("Please select  the entree: ")) | Erroneous data | r | Error massage: re-enter value | Error massage – program let user re-enter the value | - |
| 66 | int(input("Please select  the entree: ")) | Erroneous data | %$EF | Error massage: re-enter value | Error massage – program let user re-enter the value | - |
| 67 | int(input("Please select your side: ")) | Typical valid data | 2 | Value accepted | Value accepted - program continue as normal | - |
| 68 | int(input("Please select your side: ")) | Extreme valid data | 20 | Error massage: re-enter value | Error massage – program let user re-enter the value | - |
| 69 | int(input("Please select your side: ")) | Invalid data | -2 | Error massage: re-enter value | Error massage – program let user re-enter the value | - |
| 70 | int(input("Please select your side: ")) | Extreme valid data | -20 | Error massage: re-enter value | Error massage – program let user re-enter the value | - |
| 71 | int(input("Please select your side: ")) | Erroneous data | G | Error massage: re-enter value | Error massage – program let user re-enter the value | - |
| 72 | int(input("Please select your side: ")) | Erroneous data | $%^ | Error massage: re-enter value | Error massage – program let user re-enter the value | - |
| 73 | input("Confirm order (Y/N): ") | Typical valid data | Y | Value accepted | Value accepted - program continue as normal | - |
| 74 | input("Confirm order (Y/N): ") | Extreme valid data | y | Value accepted | Value accepted - program continue as normal | - |
| 75 | input("Confirm order (Y/N): ") | Invalid data | t | Error massage: re-enter value | Program respond like Y is inputted | Should let user input a correct data again |
| 76 | input("Confirm order (Y/N): ") | Extreme valid data | tttttttttttt | Error massage: re-enter value | Program respond like Y is inputted | Should let user input a correct data again |
| 77 | input("Confirm order (Y/N): ") | Erroneous data | 32 | Error massage: re-enter value | Program respond like Y is inputted | Should let user input a correct data again |
| 78 | input("Confirm order (Y/N): ") | Erroneous data | @#$%^ | Error massage: re-enter value | Program respond like Y is inputted | Should let user input a correct data again |
| 79 | input("Please insert discount code: ") | Typical valid data | happypanda | Value accepted | Value accepted - program continue as normal | - |
| 80 | input("Please insert discount code: ") | Extreme valid data | HaPpyPanDa | Value accepted | Value accepted - program continue as normal | - |
| 81 | input("Please insert discount code: ") | Invalid data | sadpanda | Error massage: re-enter value | Error massage – program let user re-enter the value | - |
| 82 | input("Please insert discount code: ") | Extreme valid data | SaDPandAA | Error massage: re-enter value | Error massage – program let user re-enter the value | - |
| 83 | input("Please insert discount code: ") | Erroneous data | -8541 | Error massage: re-enter value | Error massage – program let user re-enter the value | - |
| 84 | input("Please insert discount code: ") | Erroneous data | @#$%^& | Error massage: re-enter value | Error massage – program let user re-enter the value | - |
| 85 | input("Do you want to re-enter discount code (Y/N): ") | Typical valid data | Y | Value accepted | Value accepted - program continue as normal | - |
| 86 | input("Do you want to re-enter discount code (Y/N): ") | Extreme valid data | y | Value accepted | Value accepted - error massage -program continue as normal | Error massage should not display |
| 87 | input("Do you want to re-enter discount code (Y/N): ") | Invalid data | t | Error massage: re-enter value | Error massage – program let user re-enter the value | - |
| 88 | input("Do you want to re-enter discount code (Y/N): ") | Extreme valid data | tttttttttttt | Error massage: re-enter value | Error massage – program let user re-enter the value | - |
| 89 | input("Do you want to re-enter discount code (Y/N): ") | Erroneous data | 32 | Error massage: re-enter value | Error massage – program let user re-enter the value | - |
| 90 | input("Do you want to re-enter discount code (Y/N): ") | Erroneous data | @#$%^ | Error massage: re-enter value | Error massage – program let user re-enter the value | - |
| 91 | input("How you world like to pay(CASH/CARD): ") | Typical valid data | CASH | Value accepted | Value accepted - program continue as normal | - |
| 92 | input("How you world like to pay(CASH/CARD): ") | Extreme valid data | cash | Value accepted | Value accepted - program continue as normal | - |
| 93 | input("How you world like to pay(CASH/CARD): ") | Invalid data | No money | Error massage: re-enter value | Error massage – program let user re-enter the value | - |
| 94 | input("How you world like to pay(CASH/CARD): ") | Extreme valid data | No MoNeyyyyy | Error massage: re-enter value | Error massage – program let user re-enter the value | - |
| 95 | input("How you world like to pay(CASH/CARD): ") | Erroneous data | 852 | Error massage: re-enter value | Error massage – program let user re-enter the value | Allow only letter to be inputted |
| 96 | input("How you world like to pay(CASH/CARD): ") | Erroneous data | #$%^&\* | Error massage: re-enter value | Error massage – program let user re-enter the value | Allow only letter to be inputted |

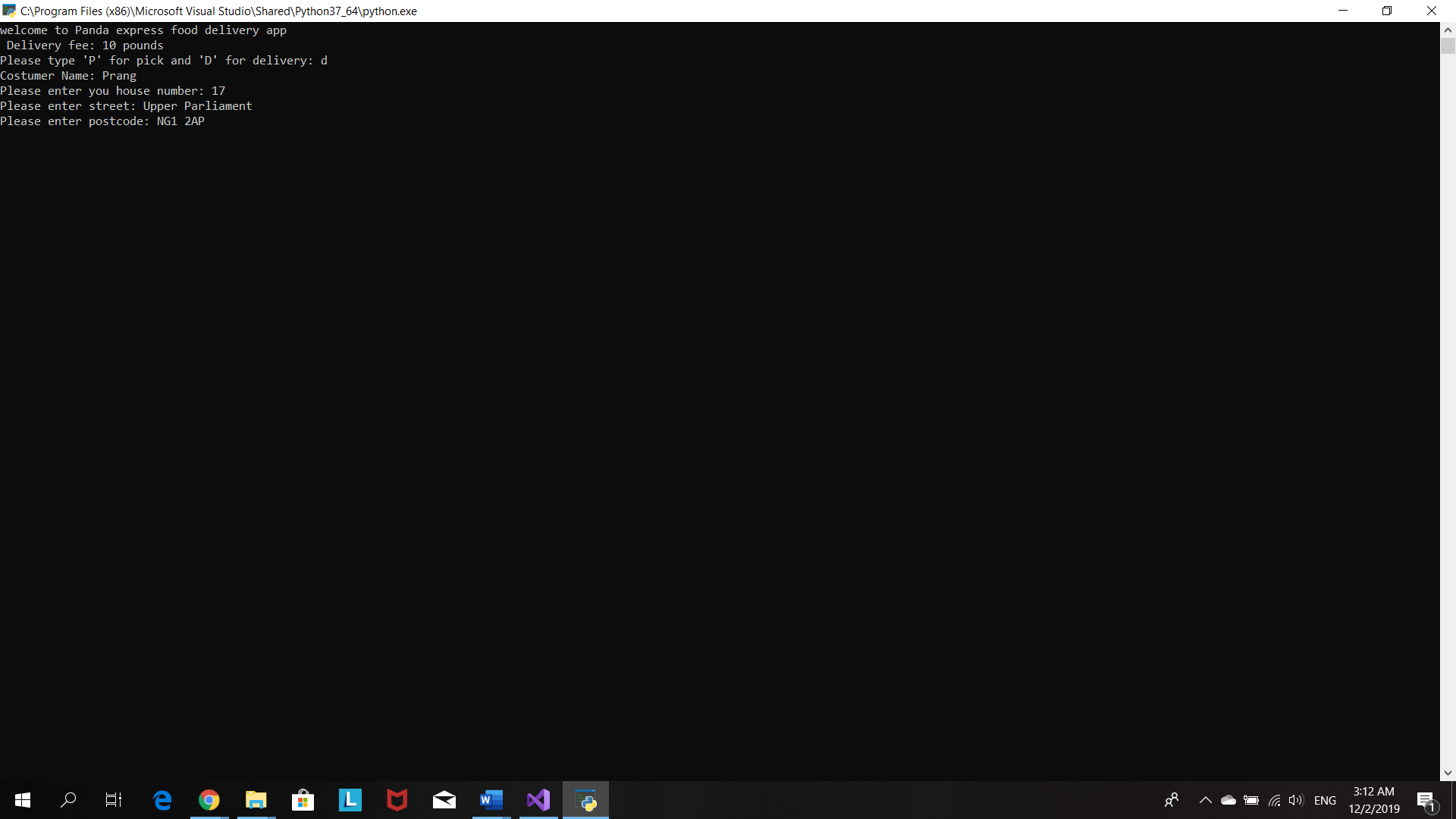
Screen shot

Start the program

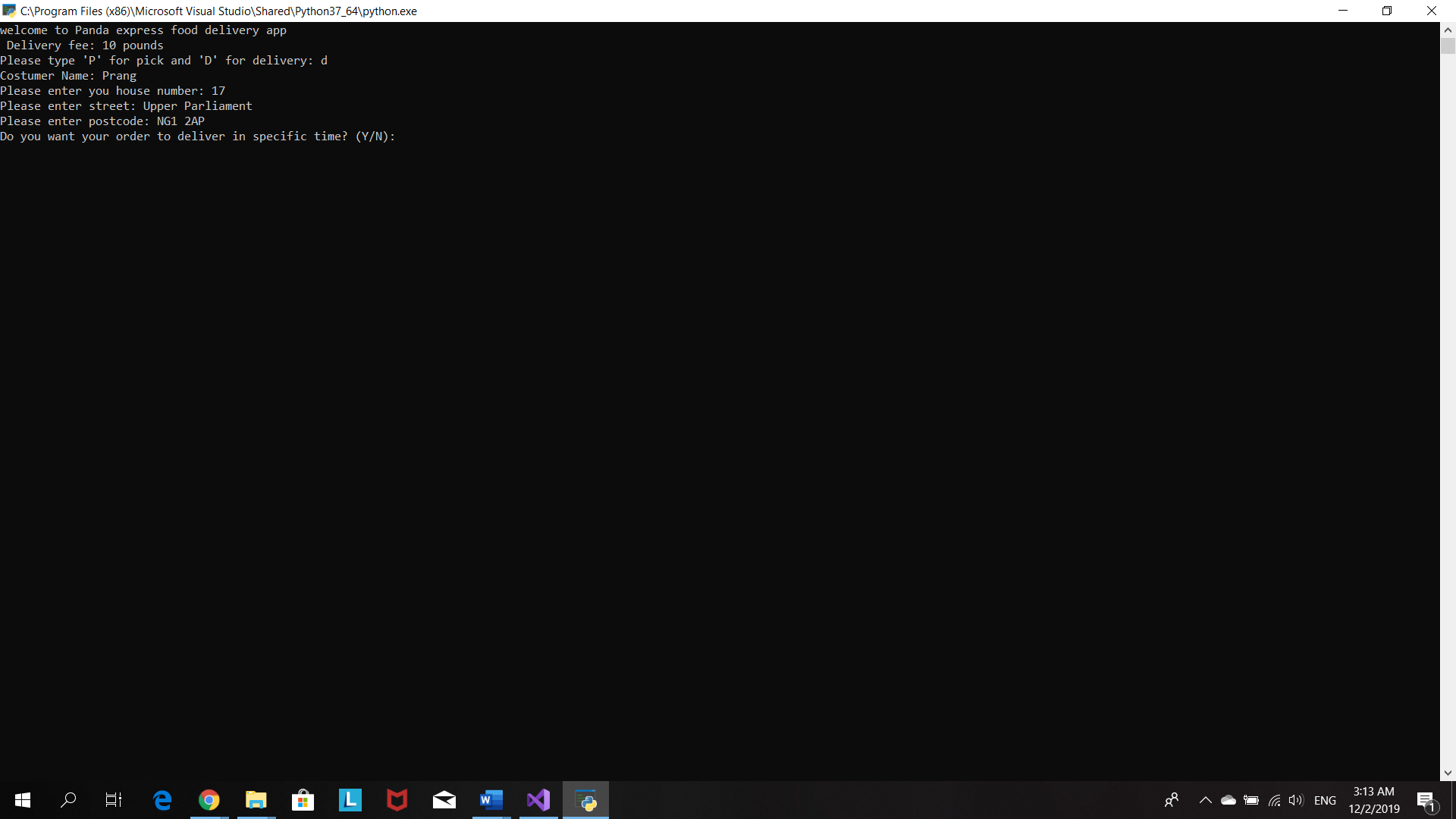
Ask costumer whether he/she wants order to be delivered or want to pick up the order by himself.



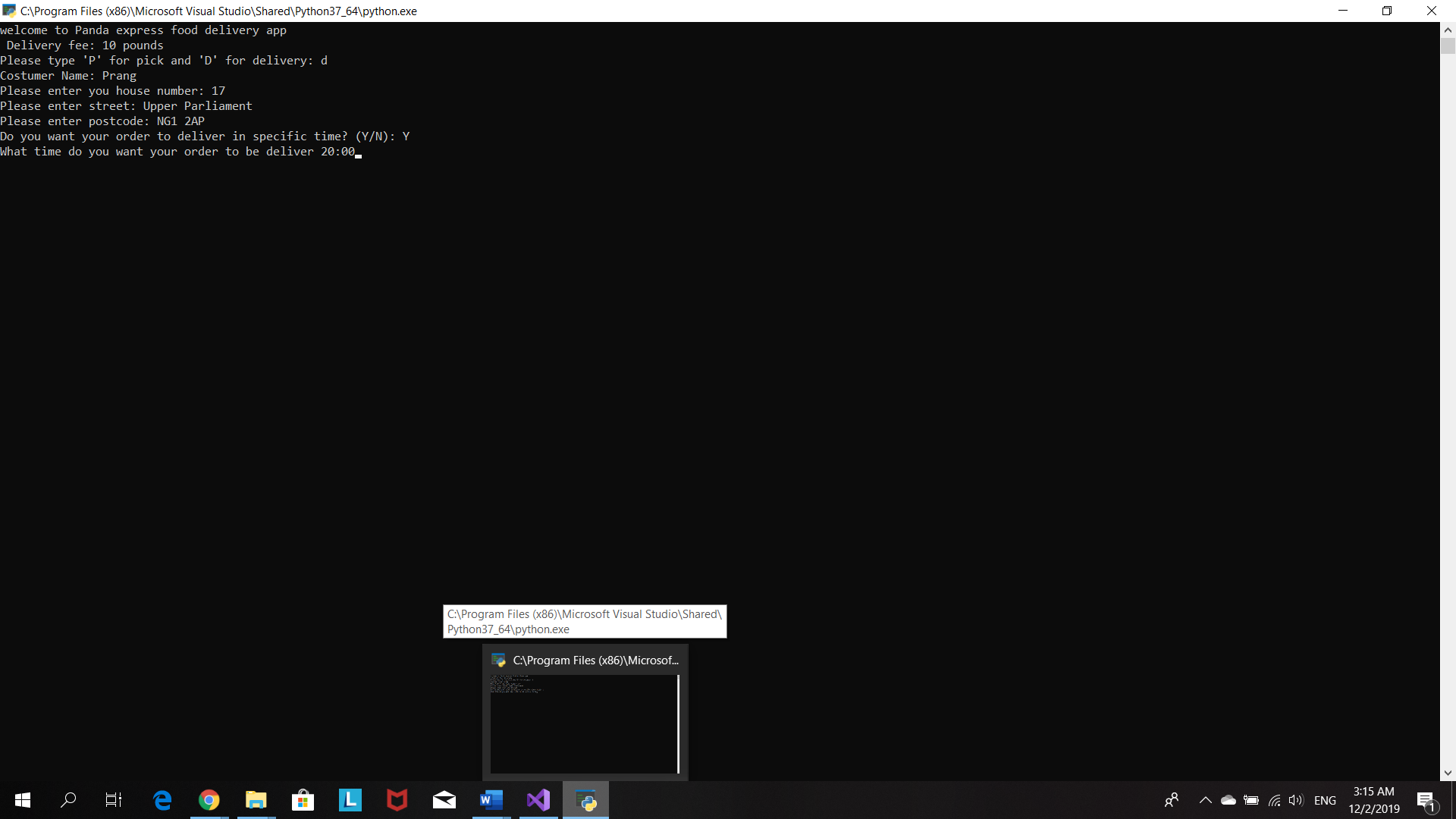
When costumer chooses ‘d’ for delivery , program will as for delivery details.



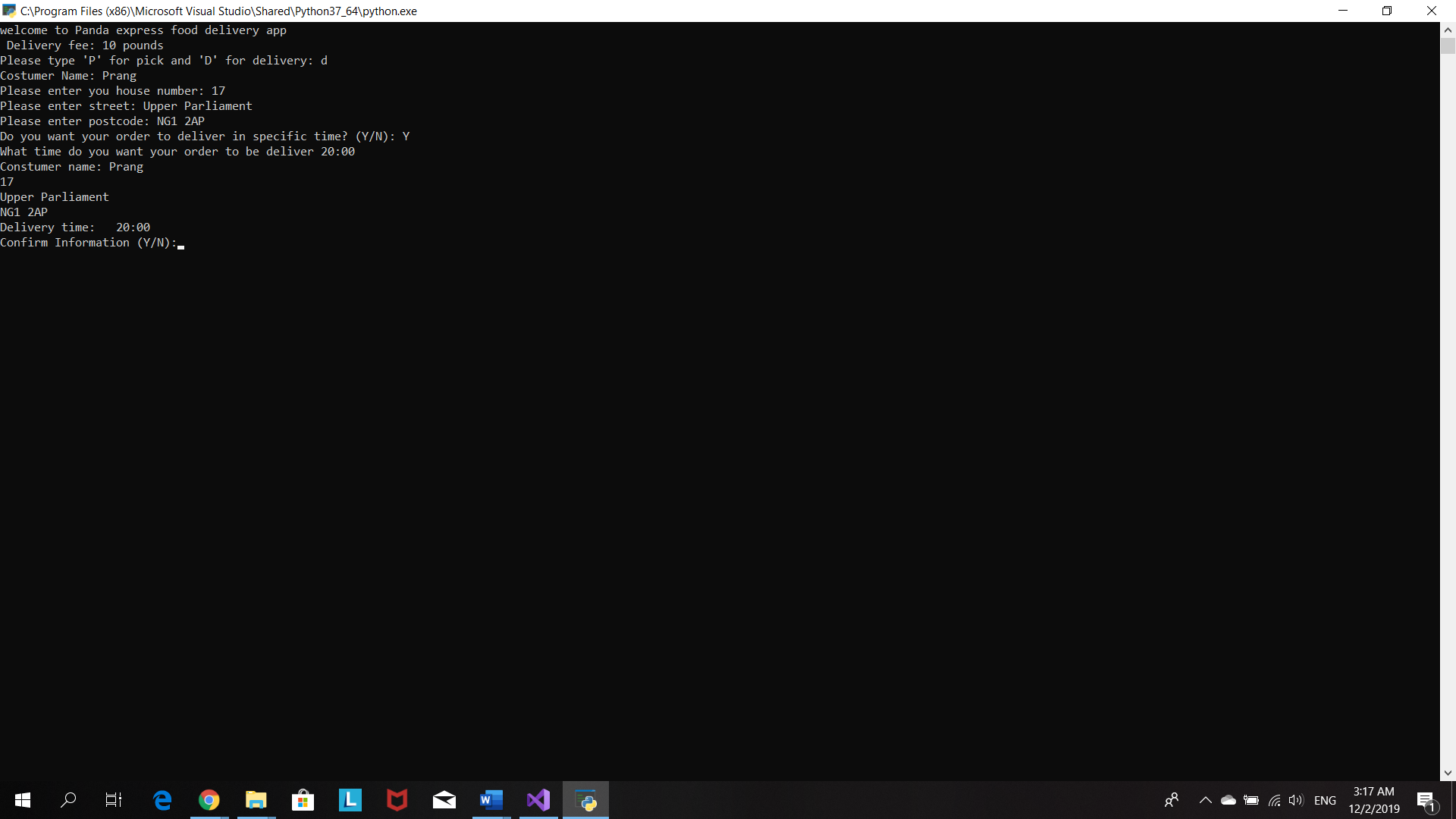
Program ask costumer do he/she wants to specific a delivery time.



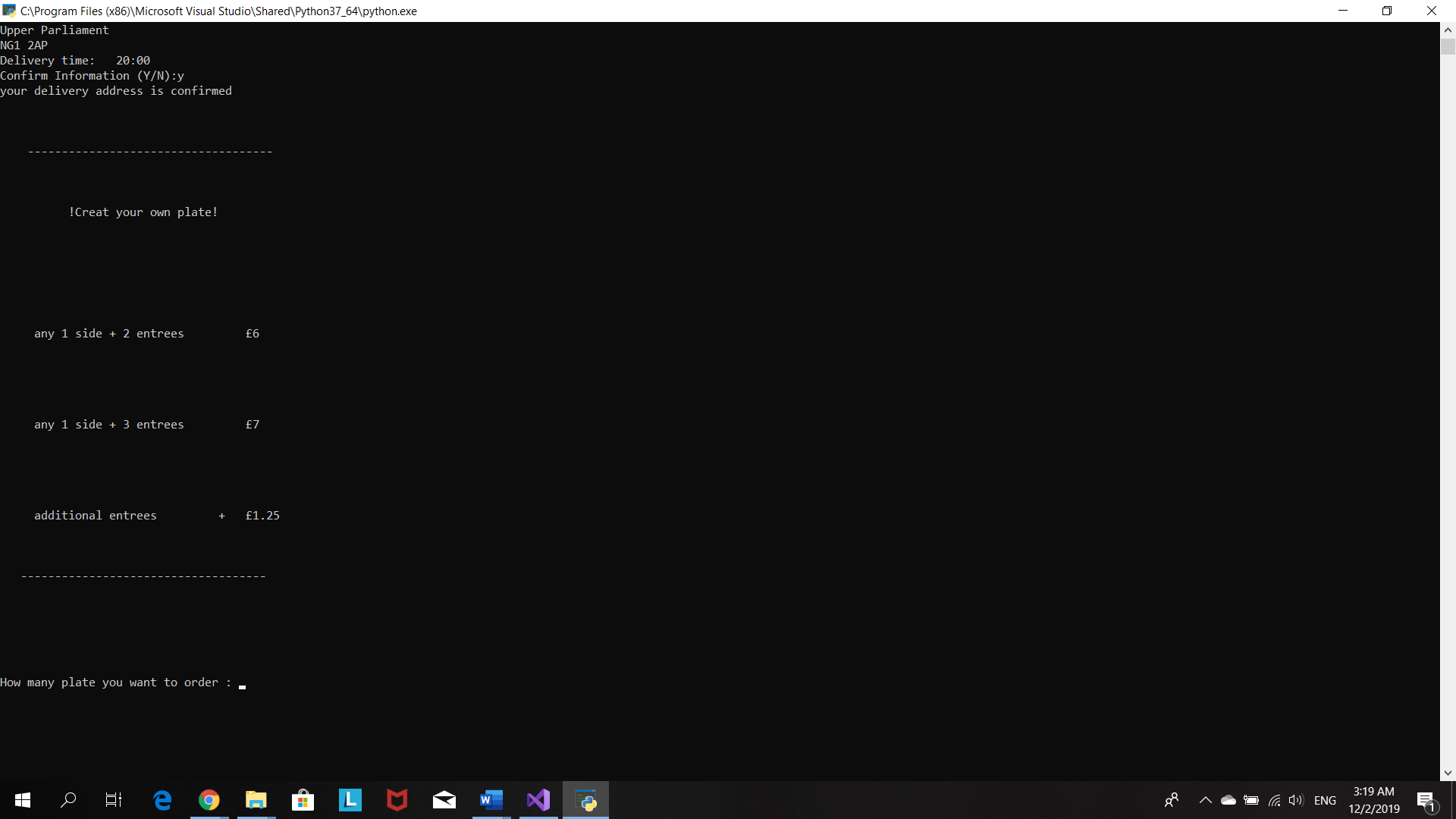
If costumer choose ‘y’ for yes, program will ask costumer to choose the time .



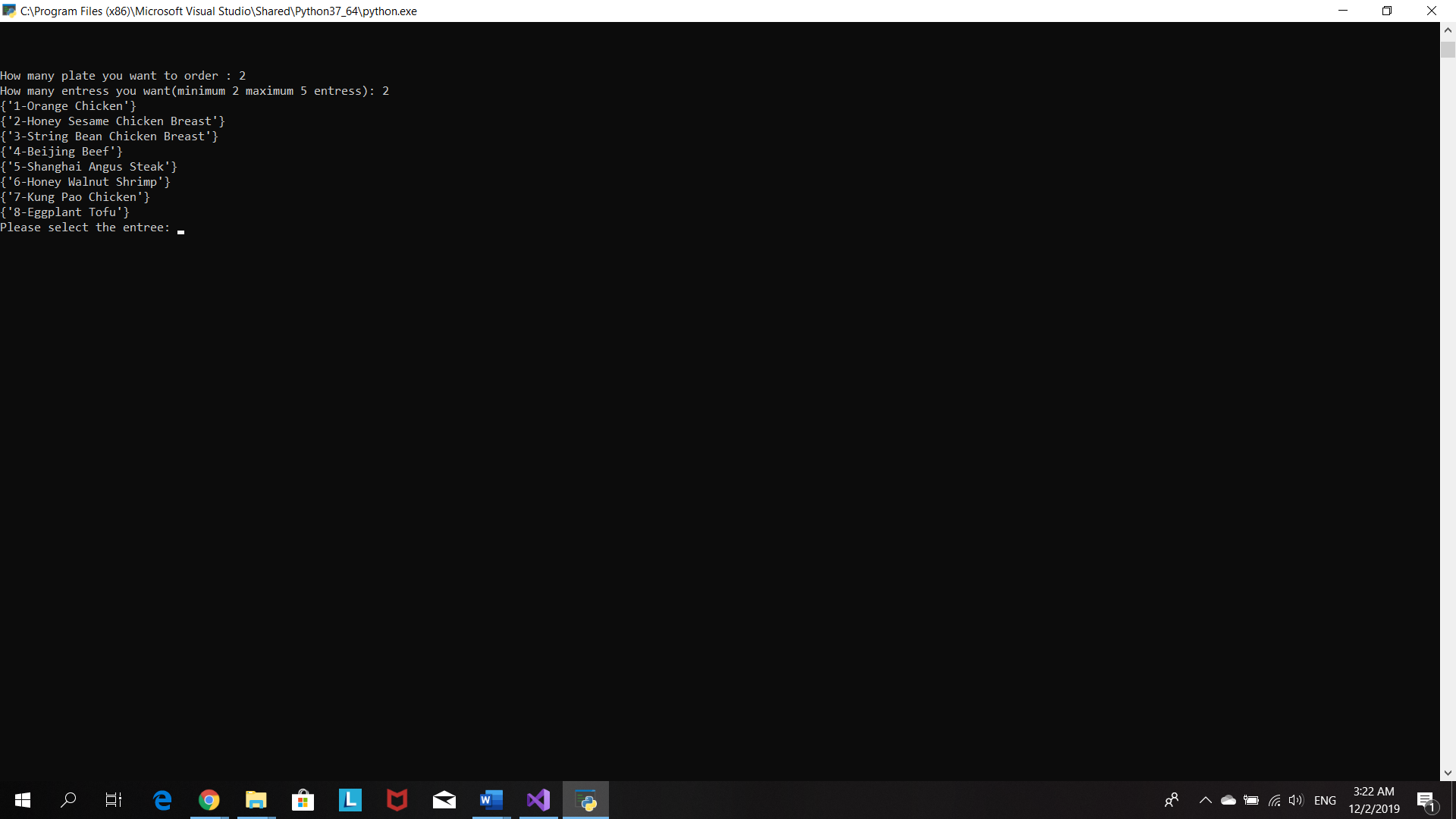
Program print out delivery details and ask costumer to confirm them.



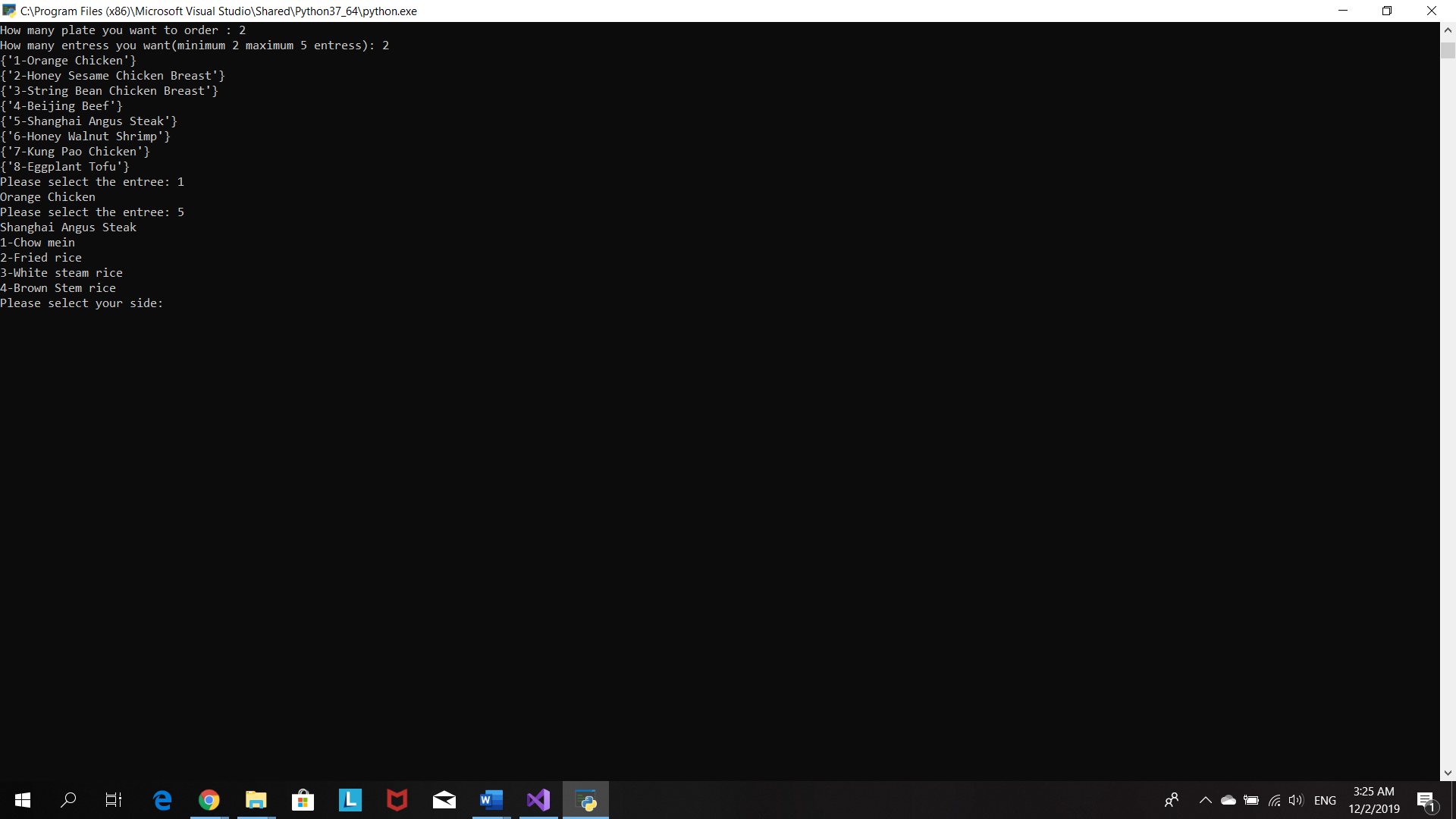
Program ask costumer how many plate he/she wants to order



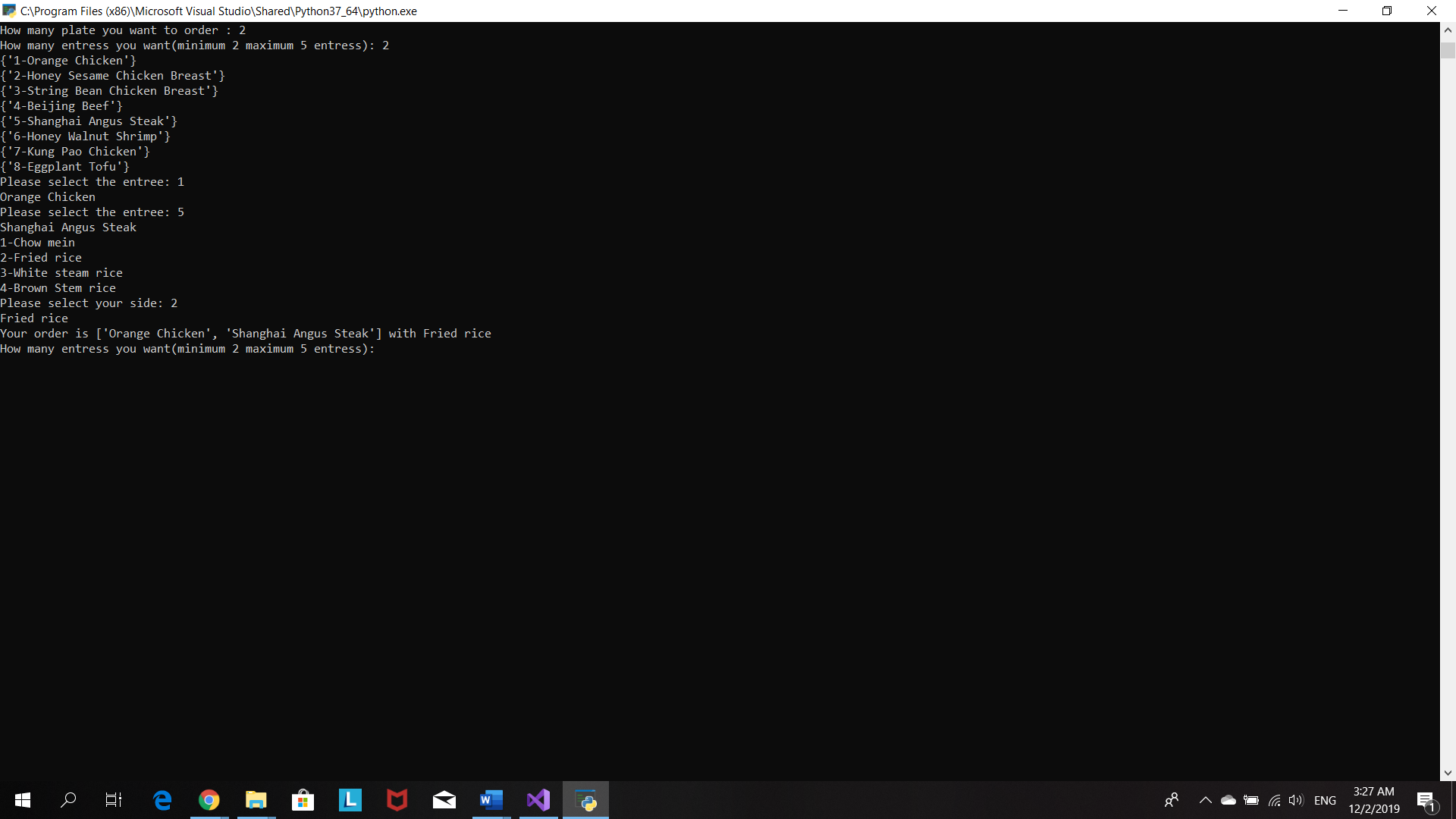
Then program ask costumer to how many entrée he/she wants and entrée menu display



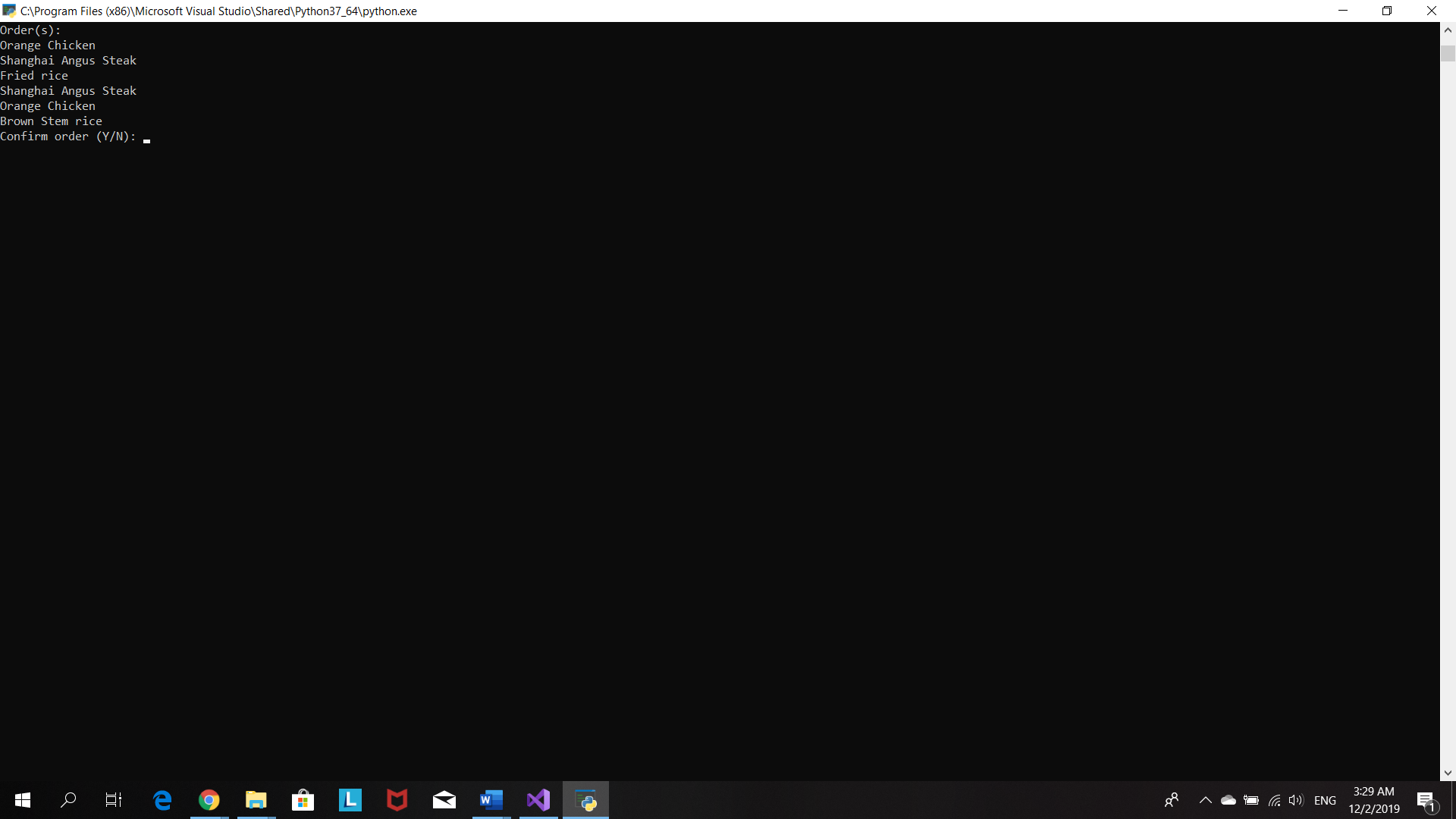
After costumer chooses entrees then program will display side menu and let costumer chooses



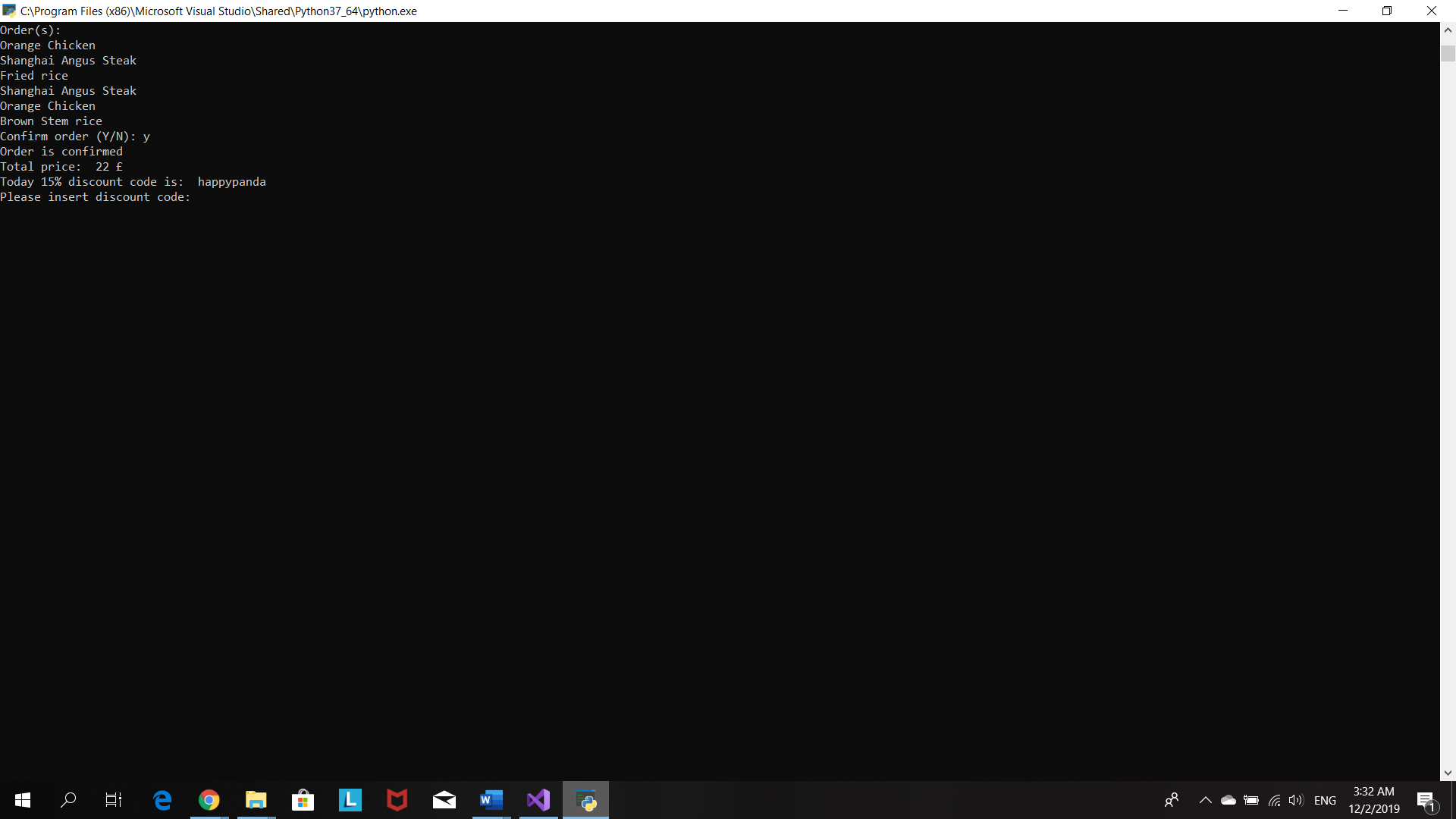
Costumer wants to order 2 plates so program let costumer chooses second plate



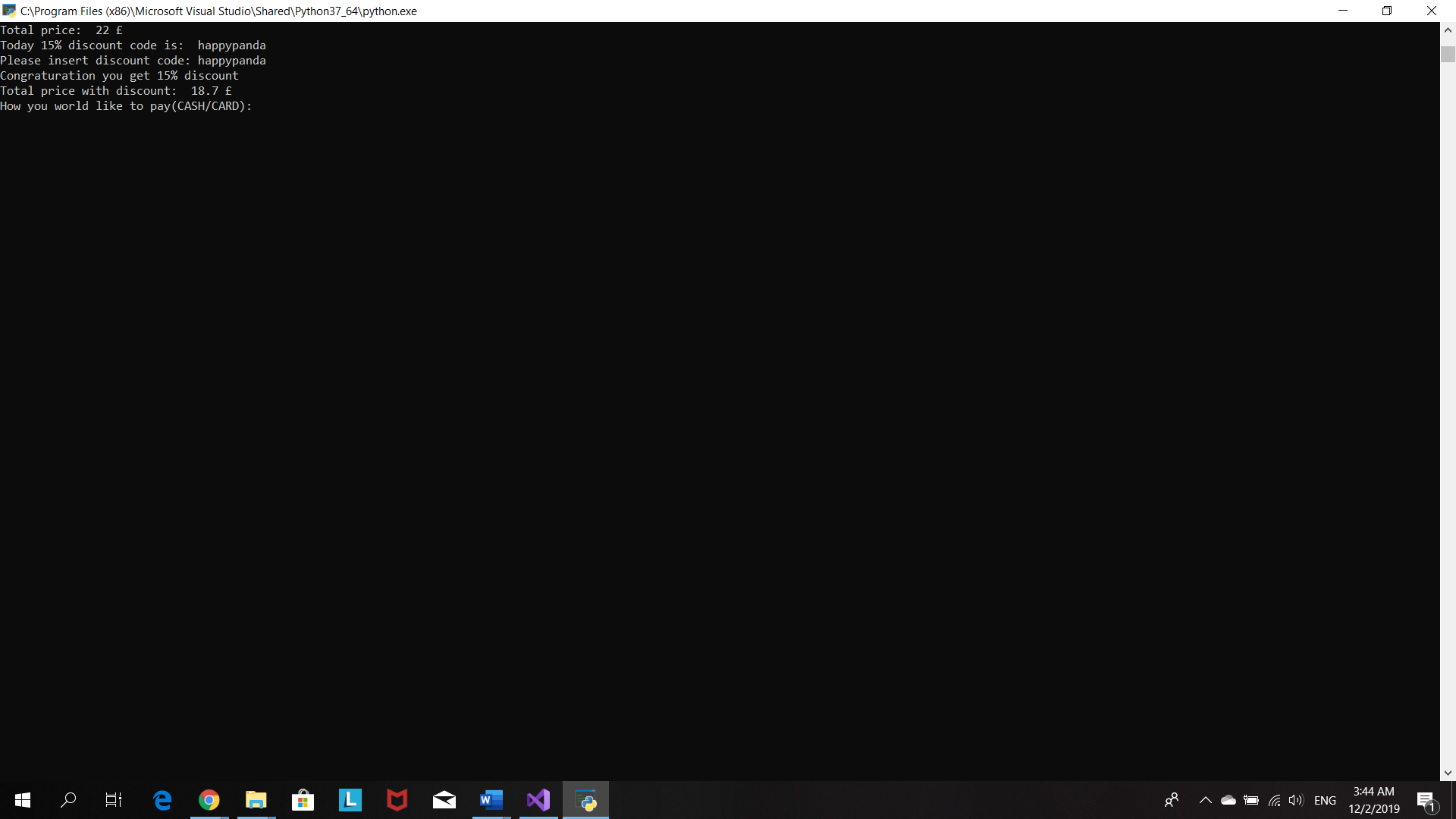
Program print order details and ask costumer to confirm them



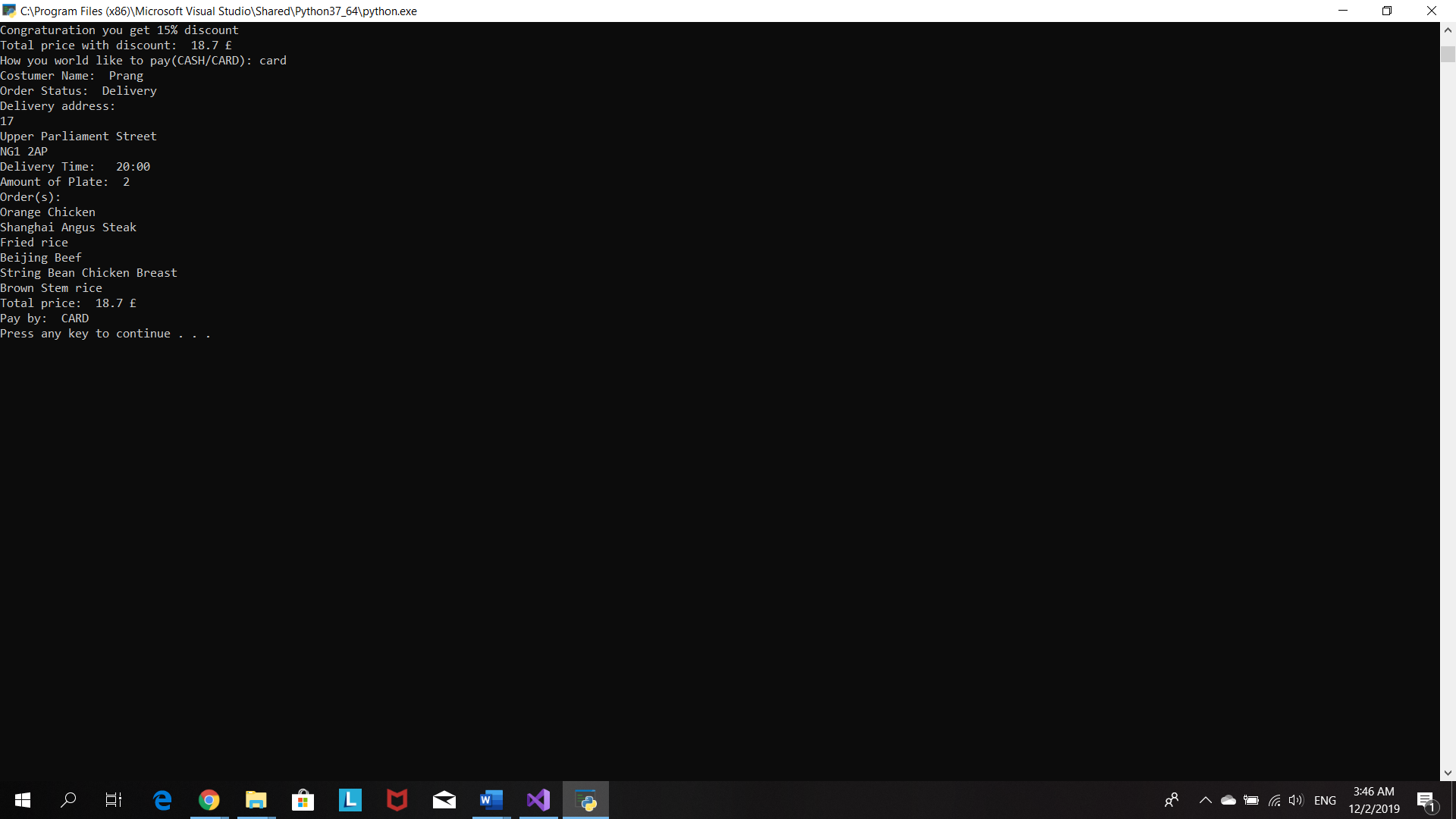
Program display a total price and inform costumer about the discount code



After costumer insert a correct code, program will re-calculate total price and ask how he/she wants to pay for order.



Program display all the details



# Source code

#Program 1: Food Delivery Program

#The purpose of the program is creat a food delivery app for local chinese resturant

#Name:Prang Kongthongluck

#Date:25/10/2019

#Version:1.0

#set up variable

price=()

totalPrice=()

payMethod=()

orderPrice=[]

userOrder=[]

menu=[

{"1-Orange Chicken"},

{"2-Honey Sesame Chicken Breast"},

{"3-String Bean Chicken Breast"},

{"4-Beijing Beef"},

{"5-Shanghai Angus Steak"},

{"6-Honey Walnut Shrimp"},

{"7-Kung Pao Chicken"},

{"8-Eggplant Tofu"}]

#printMenu is menu wittout number. It will be ues to print out the order for customer

printMenu=["Orange Chicken",

"Honey Sesame Chicken Breast",

"String Bean Chicken Breast",

"Beijing Beef",

"Shanghai Angus Steak",

"Honey Walnut Shrimp",

"Kung Pao Chicken",

"Eggplant Tofu"]

sideMenu=[

"1-Chow mein",

"2-Fried rice",

"3-White steam rice",

"4-Brown Stem rice"

]

#printsideMenu is sidemenu wittout number. It will be ues to print out the order for customer

printsideMenu=[

"Chow mein",

"Fried rice",

"White steam rice",

"Brown Stem rice"

]

userOrder=[]

#list locate inside for loop

printOrder=[]

address=[]

orderStatus=()

costumerName=()

deliveryTime=()

userentreeChoice=()

entree=()

amount=()

entreelist=[]

sidelist=[]

#---------------------------------------------------------------------------------------------------

print("welcome to Panda express food delivery app")

print("Today promotion: Free delivey  ")

#create a pickupDelivery function

def pickupDelivery():

#set some valiables as globle so it can be use outside their function

global customerName

global orderStatus

global deliveryTime

#ask customer what they prefer pick up or delivery

delivery=input("Please type 'P' for pick and 'D' for delivery: ")

#change input from delivery to be uppercase

delivery=delivery.upper()

#ask for customer name

customerName=input("Costumer Name: ")

#make customerName begin with capital letter

customerName=customerName.capitalize()

#when customer choose delivery option

if delivery== "D":

orderStatus='Delivery'

#loop to confirm deliver address

while True:

houseNumber=input("Please enter you house number: ")

# add houseNumber in to addres list

address.append(houseNumber)

street=input("Please enter street: ")

# add street in to addres list

address.append(street)

postcode=input("Please enter postcode: ")

# add postcode in to addres list

address.append(postcode)

deliveryTimeAsk=input("Do you want your order to deliver in specific time? (Type Y for yes and N for no): ")#make input in deliveryTimeAsk to be uppercase

delivertTimeAsk=deliveryTimeAsk.upper()

#if customer what their order to deliver in specific time

if (deliveryTimeAsk=="Y"):

deliveryTime=input("What time do you want your order to be deliver")#if customer dont what their order to deliver in specific time. Their order will be delivered as soon as possible

else:

deliveryTime="ASAP"

print("Your order wil be deliver as soon as possible")

#print out delivery details

print("Customer name:",costumerName)

#print address list vertically

for a in (address):

print (a)

print("Delivery time: ",deliveryTime)

#make customer to re checkt their delivery details

confirmAddress=input("Confirm Information (Type Y for yes and N for no):")

#make input of confirmAddress alwasy uppercase

confirmAddress=confirmAddress.upper()

#costumer dont confirm delivery details

if confirmAddress=="N":

#remove the data in the address list when constumer dont confirm the divery adress

address.remove(houseNumber)

address.remove(street)

address.remove(postcode)

print("Your delivery address has been cancel")

print("Please enter your delivery adress")

#customer confirm delivery details

elif confirmAddress=="Y":

print("your delivery address is confirmed")

break

else:

print("wrong input. Please try again")

#customer choose to pick up order themselves

if delivery=="P":

orderStatus='Pick Up'

print("Customer Name: ",customerName)

print("Customer will pick up the order")

#call pickupDelivery function

pickupDelivery()

print(

"""

    ------------------------------------

          !Creat your own plate!

     any 1 side + 2 entrees         £6

     any 1 side + 3 entrees         £7

     additional entrees         +   £1.25

   ------------------------------------

    """

)

#create order function

def order():

#set some valiables as globle so it can be use outside their function

global entree

global amount

#the purpose of this loop is to confirm order details

while True:

amount=int(input("How many plate you want to order : "))

#make a loop repeat respectively to amount of plate costumer choosed

for i in range(amount):

#the purpose of this loop is to make sure number of entree costumer choose is between 2 and 5

while True:

entree=int(input("How many entress you want(minimum 2 entrees and maximum 5 entress): "))

#make sure number of entree costumer choose is between 2 and 5

if (entree<=1) or (entree>=5):

print("INCORRECT INPUT.Please try again")

else:

break

#this will help to calculate the price

if (entree==2):

price=6

if (entree==3):

price=7

if (entree==4):

price=8.25

if (entree==5):

price=9.5

#add price in orderPrice list

orderPrice.append(price)

#display menu vertically

for i in (menu):

print (i)

#Purpose of this list is to display the order for costumer inside the loop so it can print each order seperatly

printOrder=[]

#make a loop repeat respectively to number of entree costumer choose

for i in range(entree):

#make sure user input right number od entree(1-8)

while True:

entreeChoice=int(input("Please selet the entree: "))

if (entreeChoice<1) or (entreeChoice>8):

print("INVALID INOUT.Please choose number 1-8")

else:

break

#to print out the correct entree choice of costumer

userentreeChoice=printMenu[entreeChoice-1]

print(userentreeChoice)

#insert user's choice of entrees in userOrder list and entreelist

userOrder.append(userentreeChoice)

entreelist.append(userentreeChoice)

#use for loop to print side menu vertically

for x in sideMenu:

print (x)

sideChoice=int(input("Please select your side: "))

#to print out the correct side dish choice of costumer

userSideChoice=printsideMenu[sideChoice-1]

print(userSideChoice)

#insert user's choice of side dish to userOrder list and sidelist

userOrder.append(userSideChoice)

sidelist.append(userSideChoice)

#print out customer choice

def OrderRecipe(choiceE,choiceS):

print("Your order is",choiceE,"with",choiceS)

OrderRecipe(entreelist,sidelist)

#customer order morethen 1 plate

#data in printOrder list will be deleted so that program able to print out each plate choic individually

if (amount>=2):

entreelist.remove(userentreeChoice)

sidelist.remove(userSideChoice)

#display order details for customer to confirm

print("number of plate: ",amount)

print("Order(s): ")

#print order details vertically

for i in (userOrder):

print (i)

confirmOrder=input("Confirm order (Type Y for yes and N for no): ")

confirmOrder.upper()

#allow customer to confirm their order

if (confirmOrder=="N"):

print("Order has been cancel")

print("Please order again")

else:

print("Order is confirmed")

break

#call order function

order()

#create payment function

def payment():

#set some valiables as globle so it can be use outside their function

global totalPrice

global payMethod

#total price is sum of number in orderPrice list

totalPrice=sum(orderPrice)

print("Total price: ",totalPrice,"£")

#inform customer discount code

print("Today 15% discount code is:  happypanda")

#make a loop TRUE

reenter="Y"

#the purpose of this loop is to make sure customer enter the correct discount code

while (reenter=="Y"):

discount=input("Please insert discount code: ")

#make input from discount always be in lower case

discount=discount.lower()

if (discount == "happypanda"):

#totelprice reduce by 15%

totalPrice=totalPrice-(0.15\*totalPrice)

print("Congraturation you get 15% discount")

print("Total price with discount: ",totalPrice,"£")

break

else:

print("Incorret discount code")

#ask customer do they want to re-enter the code

reenter=input("Do you want to re-enter discount code (Type Y for yes and N for no): ")

reenter=reenter.upper()

if (reenter == "N"):

break

else:

print("Incorrect input")

#Ask customer how they would like to pay

payMethod=input("How you world like to pay(CASH/CARD): ")

payMethod=payMethod.upper()

#check whether customer select the collect payMethod

while True:

if (payMethod=="CASH") or (payMethod=="CARD"):

break

else:

print("Incorrect in out. Please select payment method")

#call payment function

payment()

#create receipt function

#allow customer to see all details(delivery,order and payment)

def receipt():

print ("Customer Name: ",customerName)

print("Order Status: ",orderStatus)

if (orderStatus=="Delivery"):

print("Delivery address: ")

for i in address:

print (i)

print("Delivery Time: ",deliveryTime)

print("Amount of Plate: ",amount)

print("Order(s): ")

for i in userOrder:

print(i)

print("Total price: ",totalPrice,"£")

print("Pay by: ",payMethod)

#call receipt function

# receipt()

Reference

101 Computing. (2017). *Italian Takeaway Ordering system*. [online] Available at: https://www.101computing.net/italian-takeaway-ordering-system/

GeeksforGeeks. (2019). *Print lists in Python (4 Different Ways) - GeeksforGeeks*. [online] Available at: https://www.geeksforgeeks.org/print-lists-in-python-4-different-ways/

GeeksforGeeks. (2019). *Python | Remove all characters except letters and numbers - GeeksforGeeks*. [online] Available at: https://www.geeksforgeeks.org/python-remove-all-characters-except-letters-and-numbers/

Harrington, A. (2019). *1.11. Defining Functions of your Own — Hands-on Python Tutorial for Python 3*. [online] Anh.cs.luc.edu. Available at: http://anh.cs.luc.edu/python/hands-on/3.1/handsonHtml/functions.html

Programiz.com. (2019). *Python List remove()*. [online] Available at: https://www.programiz.com/python-programming/methods/list/remove [Accessed 1 Dec. 2019].