Project on the Implementation   
of an Ice Cream Price Calculator   
for Yummy Gelataria

By

**Younes Azamiyan (20017620)**

24 May 2020

**Project 2 Solution**

700047 Programming Design

Western Sydney University | The College

Contents

[1.0 Structure Chart 3](#_Toc41219083)

[2.0 Flowcharts 4](#_Toc41219084)

[2.1 Main 4](#_Toc41219085)

[2.2 display\_instructions() 5](#_Toc41219086)

[2.3 get\_tub\_num(question) 5](#_Toc41219087)

[2.4 get\_tub\_size (question) 6](#_Toc41219088)

[2.5 get\_extra\_topping(question) 6](#_Toc41219089)

[2.6 get\_coupon\_code(question) 7](#_Toc41219090)

[3.0 Python Code 8](#_Toc41219091)

[3.1 Main 8](#_Toc41219092)

[3.2 myModules 9](#_Toc41219093)

[4.0 IO Table (Test Data) 10](#_Toc41219094)

[5.0 Test Results 11](#_Toc41219095)

[5.1 Valid Data (Positive Testing): 11](#_Toc41219096)

[5.2 Invalid Data (Negative Testing): 12](#_Toc41219097)

# 1.0 Structure Chart

# 2.0 Flowcharts

## 2.1 Main

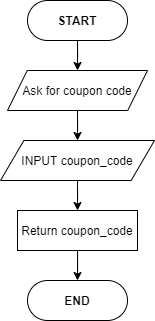
## 2.2 display\_instructions()

## 2.3 get\_tub\_num(question)

## 2.4 get\_tub\_size (question)

## 2.5 get\_extra\_topping(question)

## 2.6 get\_coupon\_code(question)



# 3.0 Python Code

## 3.1 Main

from myModules import \*

small\_pack\_price, regular\_pack\_price, party\_pack\_price = float(5.50), float(15.00), float(35.00)

xtopping\_small\_price, xtopping\_regular\_price, xtopping\_party\_price = float(2.00), float(4.50), float(8.50)

display\_instructions()

while tub\_num not in range(1, 11):

tub\_num = get\_tub\_num("Enter the number of ice cream tubs (1 - 10): ")

# The loop asks for the size of the tubs and extra topping

# for as many times as the number of ice cream tubs.

for count in range(1, tub\_num + 1):

while tub\_size not in size\_list:

tub\_size = get\_tub\_size("Enter the ice cream tub size (Small/Regular/Party): ")

while extra\_topping not in topping\_list:

extra\_topping = get\_extra\_topping("Do you need extra topping (Y/N): ")

if tub\_size == "Small" and extra\_topping == "Y":

order\_cost += small\_pack\_price + xtopping\_small\_price

elif tub\_size == "Small" and extra\_topping == "N":

order\_cost += small\_pack\_price

elif tub\_size == "Regular" and extra\_topping == "Y":

order\_cost += regular\_pack\_price + xtopping\_regular\_price

elif tub\_size == "Regular" and extra\_topping == "N":

order\_cost += regular\_pack\_price

elif tub\_size == "Party" and extra\_topping == "Y":

order\_cost += party\_pack\_price + xtopping\_party\_price

else:

order\_cost += party\_pack\_price

# After the cost for one ice cream tub is calculated,

# the variables bellow will be reset to zero, so the

# loop starts calculating the price for the next tub.

tub\_size, extra\_topping = 0, 0

while coupon\_code not in coupon\_list:

coupon\_code = get\_coupon\_code("Please, enter your coupon code here (Press enter key for no code): ")

if coupon\_code in coupon\_list:

order\_cost -= 2

elif coupon\_code == '':

break

else:

print("The code entered does not exist! Please, try again.")

print("The cost of the ice cream is $", order\_cost)

## 3.2 myModules

# The next four lines are variables and lists, required

# to enter the loops, and verify the user's input

tub\_num, tub\_size, extra\_topping, order\_cost, count, coupon\_code = 0, 0, 0, 0, 0, 0

size\_list = ["Small", "Regular", "Party"]

topping\_list = ["Y", "N"]

coupon\_list = ["VIP"]

def display\_instructions():

print("This program calculates the cost of an order based on standard Gelato prices.\nTub sizes are 250 mL for a small pack, 1 Litre for a regular pack, and 3.5 Litre for a party pack.\nExtra topping includes diced almonds, mini marshmallows, and 100s & 1000s.\nTo calculate the cost of your order please answer below statements.\n")

def get\_tub\_num(question):

# Asks for the number of tubs, and shows an

# error message for invalid inputs.

try:

tub\_num = int(input(question))

if tub\_num not in range(1, 11):

print("Out of range value! Please, try again.")

except ValueError:

print("Value error! Please, try again with an integer.")

else:

return tub\_num

def get\_tub\_size(question):

# Asks for the ice cream tub size, and shows an

# error message for invalid inputs.

tub\_size = input(question)

if tub\_size not in size\_list:

print("Incorrect input! Please, specify the tub size again.")

return tub\_size

def get\_extra\_topping(question):

# A boolean function that asks whether the user

# wants any extra toppings.

extra\_topping = input(question)

if extra\_topping not in topping\_list:

print("Incorrect input! Please, state the topping again.")

return extra\_topping

def get\_coupon\_code(question):

coupon\_code = input(question)

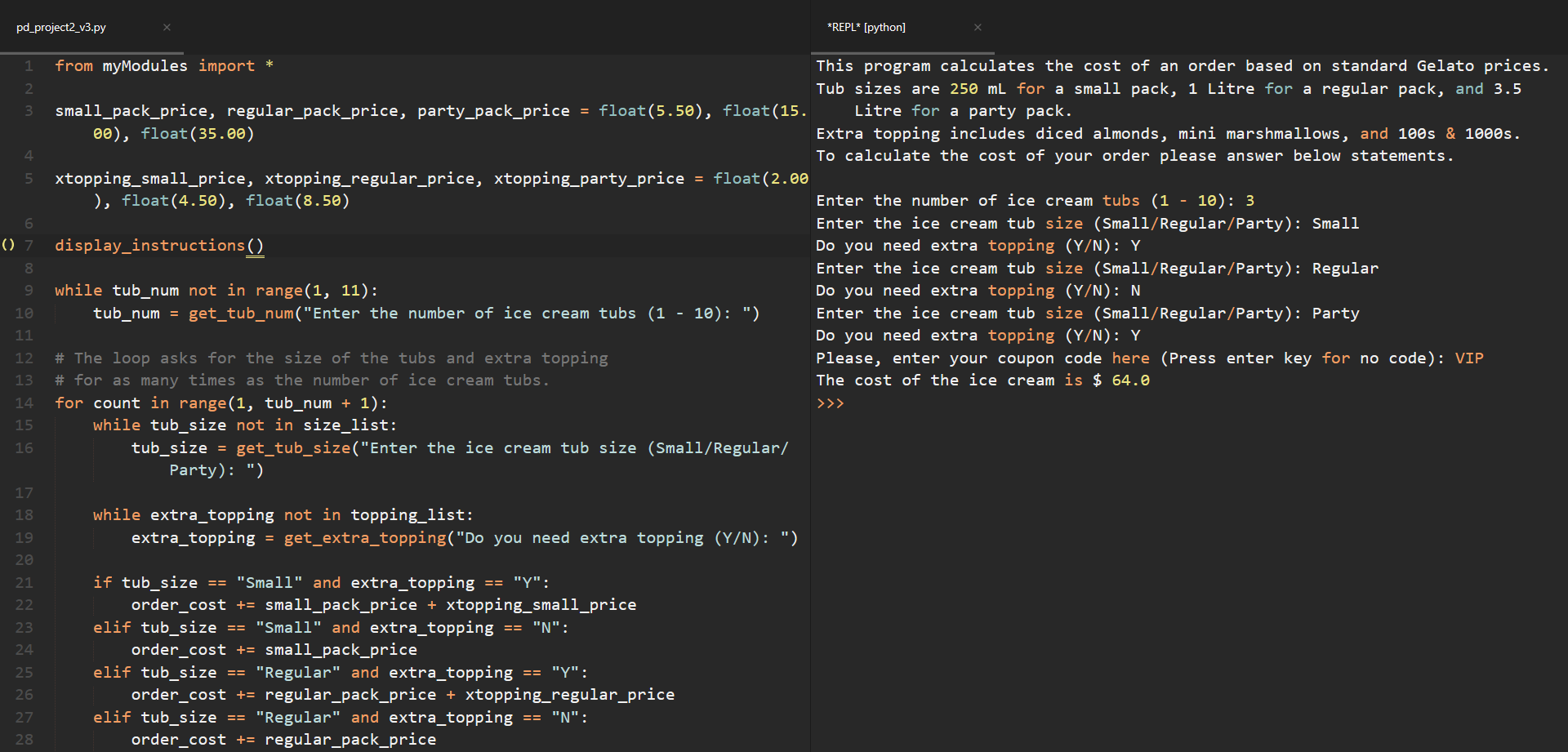
return coupon\_code

# 4.0 IO Table (Test Data)

|  |  |  |
| --- | --- | --- |
| **Dataset** | **Input** | **Expected Output** |
| 1 | Valid Data   1. 3, 2. Small, 3. Y, 4. Regular, 5. N, 6. Party, 7. Y, 8. VIP | This program calculates the cost of an order based on standard Gelato prices.  Tub sizes are 250 mL for a small pack, 1 Litre for a regular pack, and 3.5 Litre for a party pack.  Extra topping includes diced almonds, mini marshmallows, and 100s & 1000s.  To calculate the cost of your order please answer below statements.   1. Enter the number of ice cream tubs (1 - 10): 2. Enter the ice cream tub size (Small/Regular/Party): 3. Do you need extra topping (Y/N): 4. Enter the ice cream tub size (Small/Regular/Party): 5. Do you need extra topping (Y/N): 6. Enter the ice cream tub size (Small/Regular/Party): 7. Do you need extra topping (Y/N): 8. Please, enter your coupon code here (Press enter key for no code):   The cost of the ice cream is $ 64.0 |
| 2 | Invalid Data   1. 11, 2. Three, 3. 3, 4. Small, 5. Yes, 6. Y 7. Medium, 8. Regular, 9. 0, 10. N, 11. 1.1, 12. Party, 13. Y, 14. DISCOUNT 15. (Enter Key) | This program calculates the cost of an order based on standard Gelato prices.  Tub sizes are 250 mL for a small pack, 1 Litre for a regular pack, and 3.5 Litre for a party pack.  Extra topping includes diced almonds, mini marshmallows, and 100s & 1000s.  To calculate the cost of your order please answer below statements.   1. Enter the number of ice cream tubs (1 - 10):    1. Out of range value! Please, try again. 2. Enter the number of ice cream tubs (1 - 10):    1. Value error! Please, try again with an integer. 3. Enter the number of ice cream tubs (1 - 10): 4. Enter the ice cream tub size (Small/Regular/Party): 5. Do you need extra topping (Y/N):    1. Incorrect input! Please, state the topping again. 6. Do you need extra topping (Y/N): 7. Enter the ice cream tub size (Small/Regular/Party):    1. Incorrect input! Please, specify the tub size again. 8. Enter the ice cream tub size (Small/Regular/Party): 9. Do you need extra topping (Y/N):    1. Incorrect input! Please, state the topping again. 10. Do you need extra topping (Y/N): 11. Enter the ice cream tub size (Small/Regular/Party):     1. Incorrect input! Please, specify the tub size again. 12. Enter the ice cream tub size (Small/Regular/Party): 13. Do you need extra topping (Y/N): 14. Please, enter your coupon code here (Press enter key for no code):     1. The code entered does not exist! Please, try again. 15. Please, enter your coupon code here (Press enter key for no code):   The cost of the ice cream is $ 66.0 |

# 5.0 Test Results

## 5.1 Valid Data (Positive Testing):



## 5.2 Invalid Data (Negative Testing):

