**Group No: 8 Date: November 09, 2022**

**Group Members: Borja, Karl Vladimir B.**

**Cayanan, Luis Miguel I.**

**Paldeng, Carl Lester P.**

**Quiambao, Mark Erolle**

**Topic: Catering Services**

**Program Description**

- Our program acts as a shopping application where consumers can select products and order catering services. It is a program that receives user inputs for operations, orders and user information for order processing. This application also has built-in calculations for user orders, including quantity, costs for each order based on quantity, total purchase for each section, reset cart and overall total. It prompts the user input and displays its specific function. If a user's input is not recognized, it asks for a valid input.   The program provides various catering options for users to browse, compare, purchase, display total purchase, and check out.

**Purpose**

* Secured order transaction
* Saves time, money and effort
* Ease of Access
* Less physical contact
* No pressure on ordering
* View and compare products/services offered
* Convenient shopping

**Major Features**

* Options for each dedicated menu
* Input validation (display specific functions according to user input such as viewing of main options, carts, packages, order, quantity etc.)
* If user input is not recognized, it reminds the users and prompts for a valid input
* Built in calculation for user orders such as orders and their corresponding total based on their quantity, total of each section (party food, rentals and party needs, catering services) and overall total
* Provide a main cart which provides the list of items ordered from different sections along with their quantity, price, an option to checkout, reset all orders, return to main menu and total price.
* Obtain user information and details for order processing and display total amount, total to pay and various information.

**Program Flow**

**Program Guide**

By the time you open the program, you are presented with 6 options, each with their own dedicated menu and functions. The program will then ask for your input.

Text

Description automatically generated

To order food, the user must simply choose within the given options. For example, the options in Party Food. The program also asks if the user wants to order or go back to the previous page.

Graphical user interface, text

Description automatically generated

If the user wants to order, he/she is then asked to input a valid package code and quantity.

Text

Description automatically generated

If the user wishes to view the cart, the items ordered will be provided. It also asks if the user wants to return to the previous page or reset his/her cart. If user wants to reset the cart, it is cleared and will display no current items if viewed again.

**Text

Description automatically generated**

Once the user has placed his/her orders, it would be shown with their order name, quantity, price and total price on the main menu [5] cart.

**Text

Description automatically generated**

The program also asks if the user wants to checkout, reset orders or return to previous page. If the user input 2, the program resets all user orders. If the user checkouts, he/she is asked for user details. Once done, the program displays user details for confirmation. The user is also asked if he/she wants to reset the details provided. If so, the program will ask for an updated version of the user details.

Text

Description automatically generated

Once user details have been confirmed, the user total order is displayed, as well as the amount for order processing. The program also asks if the user wants to create another order or not.

Text

Description automatically generated

If the user wants to order again, then the program will return to the main options and reset all previous orders.

**Text

Description automatically generated**

In any case the user input is not recognized, the program displays an invalid input and ask for another input.

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

In addition, if the user wants to end the program, he/she must simply input the assigned value to end the program at the right section.

**Program Flowchart**

INputad

INputad

INputad

False

Start

MENU ()

Get user input and assign to main\_opt

main\_option = check\_main(main\_opt)

if main\_option is 1

information\_options()

Get user input and assign to select

opt = four\_option(select)

if opt is 1

if opt is 2

if opt is 3

Break the loop

if opt is 4

about\_us()

contact\_us()

terms\_of\_payment()

BACK()

Display invalid input

BACK()

BACK()

party\_option = check\_party\_option(answer)

party\_options()

if party\_option is 3

Break the loop

if main\_option is 2

get user input an assign to answer

if party\_option is 1

if party\_option is 2

Display invalid input

PARTY\_PACKAGES()

ask\_order\_q = ask\_if\_order\_q()

If ask\_order\_q is O

party\_code = pack\_option()

show\_cart\_partypacks()

ask\_resets = ask\_reset()

if ask\_reset is R

Set all party pack variables to 0 exept party pack prices.

while True

while

True

True

True

True

False

False

False

True

False

False

True

False

False

False

True

True

False

True

False

True

Display successfully cleared your party cart

party\_pack1,party\_pack2,party\_pack3 = 0,0,0 #NUMBER OF PARTY PACKS ORDERED

price\_pp1,price\_pp2,price\_pp3 = 10261.00, 11707.00, 12404.00 #PRICES OF EACH PARTY PACKS

pp1,pp2,pp3 = 0.00,0.00,0.00 #DISPLAYS THE TOTAL OF EACH ITEM ORDERED IN PARTY PACKS

total\_party\_pack = 0.00 #TOTAL IN PARTY PACKS

rental\_pack1,rental\_pack2,rental\_pack3 = 0,0,0 #NUMBER OF RENTAL PACKS ORDERED

price\_rp1,price\_rp2,price\_rp3 = 100.00, 130.00, 160.00 #PRICES OF EACH EQUIPMENT PACKS

item1,item2,item3,item4,item5,item6 = 0,0,0,0,0,0 #NUMBER OF RENTAL ITEMS ORDERED

price\_i1,price\_i2,price\_i3,price\_i4,price\_i5,price\_i6=100.00,100.00,20.00,50.00,15.00,50.00 #PRICES OF EACH EQUIPMENT ITEMS

p1,p2,p3,p4,p5,p6,p7,p8,p9 = 0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00 #DISPLAYS THE TOTAL OF EACH ITEM ORDERED IN EQUIPMENT RENTAL

total\_rentals = 0.00 #TOTAL IN RENTAL AND EQUIPMENT PACKS/ITEMS

wed\_pack1,wed\_pack2,wed\_pack3 = 0,0,0 #NUMBER OF WEDDING PACKS ORDERED

price\_wp1,price\_wp2,price\_wp3 = 24000.00, 38000.00, 75000.00 #PRICES OF EACH WEDDING PACKS

b\_pack1,b\_pack2,b\_pack3,b\_pack4,b\_pack5,b\_pack6 = 0,0,0,0,0,0 #NUMBER OF BIRHTDAY PACKS ORDERED

price\_bp1,price\_bp2,price\_bp3,price\_bp4,price\_bp5,price\_bp6=18000.00,28000.00,55000.00,13000.00,21000.00,38000.00 #PRICES OF EACH BIRHTDAY PACKS

c1,c2,c3,c4,c5,c6,c7,c8,c9 = 0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00 #DISPLAYS THE TOTAL OF EACH ITEM ORDERED IN CATERING SERVICES

total\_catering\_packs = 0.00 #TOTAL IN CATERING SERVICES PACKS

quantity = numberOfItems()

If party\_code is 1

If party\_code is 2

total\_party\_pack += price\_pp1\*quantity

party\_pack1 += quantity

pp1 = party\_pack1\*price\_pp1

total\_party\_pack += price\_pp2\*quantity

party\_pack1 += quantity

pp2 = party\_pack2\*price\_pp2

total\_party\_pack += price\_pp3\*quantity

party\_pack1 += quantity

pp3 = party\_pack2\*price\_pp3

Display successfully added to cart

True

True

False

False

A1

B1

while

True

**SET UP VARIABLES**

INputad

INputad

INputad

B1

A1

if main\_option is 3

rental\_options()

get user input and assign to select

rental\_option = four\_option(select)

if rental\_option is 1

if rental\_option is 2

if rental\_option is 3

if rental\_option is 4

Display invalid input

rentalMenuPacks()

ask\_order\_q = ask\_if\_order\_q()

if ask\_order\_q is O

pack\_code = pack\_option()

quantity = numberOfItems()

if pack\_code is 1

if pack\_code is 2

total\_rentals += price\_rp1\*quantity

rental\_pack1 += quantity

p1 = rental\_pack1\*price\_rp1

total\_rentals += price\_rp2\*quantity

rental\_pack2 += quantity

p2 = rental\_pack2\*price\_rp2

total\_rentals += price\_rp3\*quantity

rental\_pack3 += quantity

p3 = rental\_pack3\*price\_rp3

Display successfully added to cart

Display successfully added to cart

rentalMenuItems()

ask\_order\_q = ask\_if\_order\_q()

if ask\_order\_q is O

quantity = numberOfItems()

item\_code = six\_option()

If item\_code is 1

If item\_code is 2

If item\_code is 3

If item\_code is 4

If item\_code is 5

total\_rentals += price\_i1\*quantity

item1 += quantity

p4 = item1\*price\_i1

total\_rentals += price\_i2\*quantity

item2 += quantity

p5 = item2\*price\_i2

total\_rentals += price\_i3\*quantity

item3 += quantity

p6 = item3\*price\_i3

total\_rentals += price\_i4\*quantity

item4 += quantity

p7 = item4\*price\_i4

total\_rentals += price\_i5\*quantity

item5 += quantity

p8 = item5\*price\_i5

total\_rentals += price\_i6\*quantity

item6 += quantity

p9 = item6\*price\_i6

Break the loop

show\_cart\_rentals()

ask\_resets = ask\_reset()

if ask\_reset is R

Set all rental packs variables to 0 except rental pack and items prices

Display successfully cleared your party cart

False

False

True

True

True

True

True

False

False

False

False

True

False

False

False

False

True

True

True

False

True

True

True

True

True

False

False

False

False

False

A2

B2

while

True

INputad

INputad

INputad

B2

A2

if main\_option is 4

catering\_options()

get user input and assign to response

catering\_option = four\_option(response)

if catering\_option is 1

Display invalid input

WEDDING\_PACKAGES()

ask\_order\_q = ask\_if\_order\_q()

if ask\_order\_q is O

wed\_pack\_code = pack\_option()

quantity = numberOfItems()

if wed\_pack\_code is 1

total\_catering\_packs += price\_wp1\*quantity

wed\_pack1 += quantity

c1 = wed\_pack1\*price\_wp1

Display successfully added to cart

Display successfully added to cart

BIRTHDAY\_PACKAGES()

ask\_order\_q = ask\_if\_order\_q()

if ask\_order\_q is O

quantity = numberOfItems()

bday\_pack = six\_option()

If bday\_pack is 1

total\_catering\_packs += price\_bp1\*quantity

b\_pack1 += quantity

c4 = b\_pack1\*price\_bp1

Break the loop

show\_cart\_catering()

ask\_resets = ask\_reset()

if ask\_reset is R

Set all catering packs variables to 0 except catering pack prices such as wedding and birthday packs

Display successfully cleared your party cart

False

False

True

True

True

True

True

False

False

False

False

True

False

False

False

False

True

True

True

False

True

True

True

True

True

False

False

False

False

False

A3

B3

if catering\_option is 2

if catering\_option is 3

if catering\_option is 4

if wed\_pack\_code is 2

total\_catering\_packs += price\_wp2\*quantity

wed\_pack2 += quantity

c2 = wed\_pack2\*price\_wp2

total\_catering\_packs += price\_wp3\*quantity

wed\_pack3 += quantity

c3 = wed\_pack3\*price\_wp3

If bday\_pack is 2

If bday\_pack is 3

If bday\_pack is 4

If bday\_pack is 5

total\_catering\_packs += price\_bp2\*quantity

b\_pack2 += quantity

c5 = b\_pack2\*price\_bp2

total\_catering\_packs += price\_bp3\*quantity

b\_pack3 += quantity

c6 = b\_pack3\*price\_bp3

total\_catering\_packs += price\_bp4\*quantity

b\_pack4 += quantity

c7 = b\_pack4\*price\_bp4

total\_catering\_packs += price\_bp5\*quantity

b\_pack5 += quantity

c8 = b\_pack5\*price\_bp5

total\_catering\_packs += price\_bp6\*quantity

b\_pack6 += quantity

c9 = b\_pack6\*price\_bp6

while

True

A3

B3

INputad

INputad

INputad

INputad

D

E

D

E

D

INputad

E

End

True

False

Break the loop

while

True

Break the loop

True

True

True

False

True

False

while

True

False

False

True

False

True

False

if main\_option is 5

if main\_option is 0

False

False

False

Display invalid input

True

False

if total of party pack, party rentals and catering packs is 0

Display no current orders

BACK()

total\_cart = total\_party\_pack + total\_rentals + total\_catering\_packs

Display your cart and design

if total\_party\_pack != 0

if total\_rentals != 0

if total\_catering\_packs!= 0

show\_cart\_partypacks()

show\_cart\_rentals()

show\_cart\_catering()

Display design and current total

checkout=confirm\_checkout()

if checkout is 1

if checkout is 2

name = details()

confirmation = final\_confirmation()

if confirmation is 1

final\_output()

Break the loop

order\_again\_program = order\_again()

if order\_again\_program is 1

Set all catering packs, rental packs/items and party pack variables to 0 except their prices.

Set all catering packs, rental packs/items and party pack variables to 0 except their prices.

Display successfully cleared all orders

Display Outputs

**Flowchart for Functions**

**Main Functions**

def six\_option()

Get user input and assign to six\_option

if six\_option is “1”

if six\_option is “2”

if six\_option is “3”

if six\_option is “4”

if six\_option is “5”

if six\_option is “6”

Display Invalid input

Break the loop

return int(six\_option)

True

False

True

False

True

False

True

False

True

False

True

False

while True

Function six\_option()

def pack\_option()

while True

get pack code and assign to pack\_code

if pack\_code is “1”

if pack\_code is “2”

if pack\_code is “3”

Display Invalid Input

Break the loop

return int(pack\_code)

True

False

True

False

True

False

Function pack\_option()

def four\_option(option)

return option

int(option)

if option is “1”

if option is “2”

if option is “3”

if option is “4”

True

False

True

False

True

False

True

False

Function four\_option(option)

def BACK()

display design

get user input and assign to back

If back is 0

False

while True

True

break the loop

end

display invalid input

Function BACK()

def check\_main(x)

if x is “1”

if x is “2”

if x is “3”

if x is “4”

if x is “5”

if x is “6”

convert x to int

return x

True

False

False

False

False

False

False

True

True

True

True

True

Function check\_main(x)

def MENU()

end

Display design

Display design and title

Display menu options

Function MENU()

def numberOfItems()

True

False

Display input must be > 0

while True

get quantity and assign to quantity

except ValueError

Display input must be an integer

if quantity > 0

break the loop

return quantity

True

False

Function numberOfItems()

**Read More Functions**

display design and title

display terms of payment contents

def terms\_of\_payment()

end

Function terms\_of\_payment()

display design

display contact us contents

def contact\_us()

end

Function contact\_us()

display design and title

display about us contents

def about\_us()

end

Function about\_us()

display design and title

display Read More options

display design

def information\_options()

end

Function information\_options()

True

False

def ask\_reset()

Display options

get user input and assign to ask\_reset

if ask\_reset is R

if ask\_reset is B

Display Invalid Input

break the loop

return ask\_reset

while True

True

False

Function ask\_reset()

True

False

def ask\_if\_order\_q()

Display options

get user input and assign to ask\_order\_q

if ask\_order\_q is O

Display Invalid Input

break the loop

return ask\_order\_q

while True

True

False

if ask\_order\_q is B

Function ask\_if\_order\_q()

display title and design

display design

display rental menu packs

def rentalMenuPacks()

end

Function rentalMenuPacks()

**Party Food Functions**

display title and design

display design

display rental menu items

def rentalMenuItems()

end

Function rentalMenuItems ()

def rental\_options()

display title and design

display design

display rental options

end

Function rental\_options()

**Rentals and Party Needs Functions**

def PARTY\_PACKAGES()

display party packages

end

Function PARTY\_PACKAGES()

display design and title

display party options

display design

def party\_options()

end

Function party\_options()

def show\_cart\_partypacks()

True

False

display title and design

if total\_party\_pack != 0

if party\_pack1 != 0

if party\_pack2 != 0

if party\_pack3 != 0

Display party pack 1 orders

Display party pack 2 orders

Display party pack 3 orders

display titles

display no current items

Display party pack total

end

True

False

True

False

True

False

Function show\_cart\_partypacks()

def check\_party\_option (party\_option)

if party\_option is “1”

if party\_option is “2”

if party\_option is “3”

int(party\_option)

return party\_option

True

False

True

False

True

False

Function check\_party\_option (party\_option)

def show\_cart\_rentals()

display title and design

if total\_rentals != 0

display titles

display no current items

end

True

False

Display rental pack 1 orders

Display rental cart total

Display rental pack 2 orders

Display rental pack 3 orders

Display rental item 1 orders

Display rental item 2 orders

Display rental item 3 orders

Display rental item 4 orders

Display rental item 5 orders

Display rental item 6 orders

if rental\_pack1 != 0

if rental\_pack2 != 0

if rental\_pack3 != 0

if item1 != 0

if item2 != 0

if item3 != 0

if item4 != 0

if item5 != 0

if item6 != 0

True

False

True

False

True

False

True

False

True

False

True

False

True

False

True

False

True

False

Function show\_cart\_rentals()

**Catering Services Functions**

end

def BIRTHDAY\_PACKAGES()

Display Birthday Packages

Function BIRTHDAY\_PACKAGES()

end

def WEDDING\_PACKAGES()

Display Wedding Packages

Function WEDDING\_PACKAGES()

def catering\_options()

display title and design

display catering options

display design

end

Function catering\_options()

display title and design

end

True

False

def show\_cart\_catering()

display no current items

if total\_catering\_packs != 0

display titles

Display wedding pack 1 orders

Display wedding pack 2 orders

Display wedding pack 3 orders

Display birthday pack 1 orders

Display birthday pack 2 orders

Display birthday pack 3 orders

Display birthday pack 4 orders

Display birthday pack 5 orders

Display birthday pack 6 orders

Display catering packs total

if wed\_pack1 != 0

if wed\_pack2 != 0

if wed\_pack3 != 0

if b\_pack1 != 0

if b\_pack2 != 0

if b\_pack3 != 0

if b\_pack4 != 0

if b\_pack5 != 0

if b\_pack6 != 0

True

False

True

False

True

False

True

False

True

False

True

False

True

False

True

False

True

False

Function show\_cart\_catering()

**All Cart Conclusion Functions**

Display info and design

get user name, number, address, date\_order, date\_req, guest\_count and payment

Display user details and ask if data is accurate. Display design also

return name

def details()

Function details()

def confirm\_checkout()

display design

True

False

display options

get user input and assign to checkout

if checkout is “1”

if checkout is “2”

if checkout is “3”

display invalid input

break the loop

return int(checkout)

while True

True

False

True

False

Function confirm\_checkout()

def final\_confirmation()

Display confirmation

Display

design

get user data and assign to confirmation

if confirmation is “1”

if confirmation is “2”

Display

invalid option

break the loop

return int(confirmation)

while True

False

True

True

False

Function final\_confirmation()

def final\_output()

display username, total bill, how much to pay, other information and design

end

Function final\_output()

def order\_again()

display design and ask if user want to create another order

get user input and assign

to order\_again

if order\_again is “1”

if order\_again is “2”

display invalid option

break the loop

return int(order\_again)

while True

False

True

False

True

Function order\_again()

**Definition Of Terms**

|  |  |
| --- | --- |
| Variables used in the main loop | |
| quantity | **-** used to identify the number of packages/items the user wants to order |
| ask\_resets | - used to identify if user wants to reset his/her order |
| ask\_order\_q | **-** variable used to ask if user want to order or back (return to previous page) |
| main\_option | **-** option for main options |
| opt | **-** option for information options |
| party\_option | **-** options for party food |
| party\_code | **-** code of the party pack user wants to order |
| rental\_options | **-** options for rental and party needs |
| pack\_code | **-** code of the rental pack user wants to order |
| item\_code | **-** code of the rental item user wants to order |
| catering\_option | **-** option for catering services |
| wed\_pack\_code | **-** used to identify the code of the wedding pack user wants to order |
| bday\_pack | **-** used to identify the code of the birthday pack user wants to order |
| checkout | **-** used to identify if user wants to checkout, reset order or back (return to previous page) |
| name | **-** used to display the user name |
| confirmation | **-** confirmation of user details |
| order\_again\_program | **-** identify if user wants to order again or not |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Party Pack Variables | | | | | |
| party\_pack1 | Quantity of Party Packs Ordered in | Party pack 1 | **price\_pp1** | Prices of | Party pack 1 |
| party\_pack2 | Party pack 2 | **price\_pp2** | Party pack 2 |
| party\_pack3 | Party pack 3 | **price\_pp3** | Party pack 3 |
| pp1 | Display the total of each pack ordered in | Party pack 1 | **total\_party\_pack** | Total Costs in Party Pack | |
| pp2 | Party pack 2 |
| pp3 | Party pack 3 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Rental Pack Variables | | | | | |
| rental\_pack1 | Quantity of Rental Packs Ordered in | Rental pack 1 | **price\_rp1** | Prices of | Rental pack 1 |
| rental\_pack2 | Rental pack 2 | **price\_rp2** | Rental pack 2 |
| rental\_pack3 | Rental pack 3 | **price\_rp3** | Rental pack 3 |
| item1 | **Quantity of Rental Items Ordered in** | Item 1 | **price\_i1** | Prices of | Item 1 |
| item2 | Item 2 | **price\_i2** | Item 2 |
| item3 | Item 3 | **price\_i3** | Item 3 |
| item4 | Item 4 | **price\_i4** | Item 4 |
| item5 | Item 5 | **price\_i5** | Item 5 |
| item6 | Item 6 | **price\_i6** | Item 6 |
| p1 | Display the total of each pack/item ordered in | Rental pack 1 | **p6** | Display the total of item ordered in | Item 3 |
| p2 | Rental pack 2 | **p7** | Item 4 |
| p3 | Rental pack 3 | **p8** | Item 5 |
| p4 | Item 1 | **p9** | Item 6 |
| p5 | Item 2 | **total\_rentals** | Total cost in rental and equipment packs/items | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Catering Services Variables | | | | | |
| wed\_pack1 | Quantity of Wedding Packs Ordered in | Wedding Package 1 | **price\_wp1** | Prices of | Wedding Package 1 |
| wed\_pack2 | Wedding Package 2 | **price\_wp2** | Wedding Package 2 |
| wed\_pack3 | Wedding Package 3 | **price\_wp3** | Wedding Package 3 |
| b\_pack1 | **Quantity of Rental Items Ordered in** | Birthday Package 1 | **price\_bp1** | Prices of | Birthday Package 1 |
| b\_pack2 | Birthday Package 2 | **price\_bp2** | Birthday Package 2 |
| b\_pack3 | Birthday Package 3 | **price\_bp3** | Birthday Package 3 |
| b\_pack4 | Birthday Package 4 | **price\_bp4** | Birthday Package 4 |
| b\_pack5 | Birthday Package 5 | **price\_bp5** | Birthday Package 5 |
| b\_pack6 | Birthday Package 6 | **price\_bp6** | Birthday Package 6 |
| c1 | Display the total of each pack ordered in | Wedding Package 1 | **c6** | Display the total of each pack ordered in | Birthday Package 3 |
| c2 | Wedding Package 2 | **c7** | Birthday Package 4 |
| c3 | Wedding Package 3 | **c8** | Birthday Package 5 |
| c4 | Birthday Package 1 | **c9** | Birthday Package 6 |
| c5 | Birthday Package 2 | **total\_catering\_packs** | Total cost in catering packs | |

|  |  |  |  |
| --- | --- | --- | --- |
| Main Functions with their respective purpose and variables | | | |
| MENU() | - Display the main options of the program and program design | | |
| check\_main(x) | – Check if user input is within the given choices and convert it into an integer and return the value. If not, return x. | **x** | **-** used to identify user input |
| four\_option (option) | - Check if user input is within the given choices and convert it into an integer and return the value. If not, return option. | **option** | - used to identify user input |
| BACK() | - Display design and enter loop. Ask the user to input 0 if he wants to back to the previous page. If user input is not recognized, display invalid input and loop back. | **back** | - determine user input if he wants to go back to the previous page |
| six\_option() | - Enter loop and ask user for item/pack code with 6 options. If user input is not within the choices, display invalid input and ask again. If user input is within the choices, exit the loop and return user input as an integer. | **pack\_code** | - stores user item/pack code |
| numberOfItems() | – Enter loop and ask for user input of item/pack quantity. If user input is not an integer, display input must be an integer and ask again. If user input is an integer but less than or equal to 0, display input must be greater than 0 and ask again. Else, break the loop and return the quantity. | **quantity** | **-** number of items/packs user wants to order |
| ask\_reset() | - Display choices and enter loop. Ask if user wants to reset order or not. If user input is not within the choices, display invalid input and ask again. Else, break the loop and return user input | **ask\_reset** | - stores user input if he/she wants to reset orders |
| ask\_if\_order\_q() | - Display choices and enter loop. Ask if user wants order or return to previous page. If user input is not within the choices, display invalid input and ask again. Else, break the loop and return user input | **ask\_order\_q** | **-** stores user input if he/she wants to order or go back to previous page |

|  |  |
| --- | --- |
| Read More Functions with their respective purpose | |
| information\_options() | – Display information options and design |
| about\_us() | – Display BURI catering about us section and design |
| contact\_us() | – Display contacts of BURI catering and design |
| terms\_of\_payment() | – Display information about BURI catering terms of payment and design |

|  |  |  |  |
| --- | --- | --- | --- |
| Party Food Functions with their respective purpose and variables | | | |
| party\_options() | - Display party options and design | | |
| check\_party\_option(party\_option) | - Check if user input is within the given choices and convert it into an integer and return the value. If not, return user input. | **party\_option** | **-** used if user input is within the choices |
| PARTY\_PACKAGES () | - Display party packages and codes and design | | |
| show\_cart\_partypacks() | – Display design and check if total of party pack != 0. If it’s true, validate conditions, display party pack orders and their total. Else, display no current items in the shopping list. | | |

|  |  |
| --- | --- |
| Rental and Party Needs Functions with their respective purpose | |
| rental\_options() | – Display rental options and design |
| rentalMenuPacks() | – Display rental menu on packages and design |
| rentalMenuItems() | – Display rental menu on items and design |
| show\_cart\_rentals() | – Display design and check if total of total\_rentals != 0. If its true, validate conditions, display rental orders and their total. Else, display no current items in the shopping list. |

|  |  |
| --- | --- |
| Catering Services Functions with their respective purpose | |
| catering\_options() | – Display catering services options and design |
| WEDDING\_PACKAGES() | – Display wedding packages, codes and design |
| BIRTHDAY\_PACKAGES() | – Display birthday packages, codes and design |
| show\_cart\_catering() | – Display design and check if total of total\_catering\_packs != 0. If its true, validate conditions, display catering packs orders and their total. Else, display no current items in the shopping list. |

|  |  |  |  |
| --- | --- | --- | --- |
| All Cart Conclusion Functions with their respective purpose and variables | | | |
| confirm\_checkout() | – Display checkout options, design and enter loop. Get user input and validate it. If user input is not within the choices, display invalid input and ask again. Else, convert user input to integer and return it. | **checkout** | **-** use to get user input if he wants to checkout, reset or return to previous page |
| details() | – Get user details such as name, number etc., display it back to the user and ask to check it wisely. Finally, program return user input on name | **option** | - used to identify user input |
| **name** | **-** get user name |
| **number** | **-** get user number |
| **address** | **-** get user venue address |
| **date\_order** | **-** the date which order is created |
| **date\_req** | **-** the date of order delivery |
| **guest\_count** | **-** the number of guests |
| **payment** | **-** user mode of payment |
| final\_confirmation() | – Display user confirmation of details, design and enter loop. Ask user for confirmation and if user input is not recognized, display invalid option and ask again. Else, break the loop and return user input as an integer | **confirmation** | – get user input on confirmation of details |
| final\_output() | – Display final outputs, such as name, total bill, how much to pay and all other details. | | |
| order\_again() | – Display order again options, design and enter loop. Get user input and validate it. If user input is within the choices, break the loop and return user input as an integer. If not, display invalid option and ask again. | **order\_again** | - get user input about creating another order or not |

**Program Code**

#--------------------MAIN FUNCTIONS------------------------------------#

def MENU():

print("O"+"="\*37+"\*\*\*BURI CATERING SERVICE\*\*\*"+"="\*38+"O")

print("""\t\t\t\t[1] Read More

\t\t\t\t[2] Party Food

\t\t\t\t[3] Rentals and Party Needs

\t\t\t\t[4] Catering Services

\t\t\t\t[5] View your cart

\t\t\t\t[0] End Program""")

print("O"+"="\*102+"O")

def check\_main(x):

match x:

case "1"|"2"|"3"|"4"|"5"|"0":

x = int(x)

return x

def BACK():

print("O"+"="\*102+"O")

while True:

back = input("Press [0] to go back: ")

if back == "0":

break

else:

print("Invalid Input")

def four\_option(option):

match option:

case "1"|"2"|"3"|"4":

option = int(option)

return option

def six\_option():

while True:

six\_option = input("Input Code [1-6]: ")

match six\_option:

case "1"|"2"|"3"|"4"|"5"|"6":

break

case \_:

print("Invalid Input")

return int(six\_option)

def pack\_option():

while True:

pack\_code = input("Input Pack Code [1-3]: ")

match pack\_code:

case "1"|"2"|"3":

break

case \_:

print("Invalid Input")

return int(pack\_code)

def numberOfItems():

while True:

try:

quantity = int(input("Quantity: "))

if quantity > 0:

break

else:

print("Input must be greater than 0")

except ValueError:

print("Input must be an integer")

return quantity

def ask\_reset():

print("\t\t\t\t[R] Reset Order\t[B] Back")

while True:

ask\_reset = input("Input Operation[R/B]: ")

if ask\_reset == "R" or ask\_reset == "B":

break

else:

print("Invalid Input")

return ask\_reset

def ask\_if\_order\_q():

print("\t\t\t\t[O] Order\t[B] Back")

while True:

ask\_order\_q = input("Input Operation[O/B]: ")

if ask\_order\_q == "O" or ask\_order\_q == "B":

break

else:

print("Invallid Input")

return ask\_order\_q

#--------------------------------READ MORE FUNCTIONS--------------------------------#

def information\_options():

print("O"+"="\*42+"\*\*\*Regarding Us\*\*\*"+"="\*42+"O")

print("""\t\t\t\t[1] About Us

\t\t\t\t[2] Contact Us

\t\t\t\t[3] Terms of Payment

\t\t\t\t[4] Back""")

print("O"+"="\*102+"O")

def about\_us():

print("O"+"="\*44+"\*\*\*About Us\*\*\*"+"="\*44+"O")

print("""\n Since 2018, we at BURI Catering have been serving up joyful occasions. Curiosity led

to the founding of our current catering company, and as we gathered customers and

culinary expertise, we endeavored to enhance our offerings. Carl and Vladimir are the

founders; they know caterers who have studied culinary arts, so they can offer you

with a delectable dinner whenever you need their services. Mark organizes the menus for

celebrations such as weddings and birthdays. He has catering and event organizing experience.

In five years, BURI Catering has established itself as one of the city's

caterers (San Fernando, Pampanga) with the most rapid expansion and highest demand. With

30 people that are accessible for weddings and birthdays, we have become one of the most

well-known catering businesses. Since the beginning of our careers, we have been recognized

as the catering industry's most dependable brand.

The catering business and restaurant continue to expand in popularity, and the

restaurant's dedicated team of professionals and personnel give all of its presentations

5-star district-wide appeal. Both large and small catering companies offer social meals and

floral and décor arrangements for birthdays and weddings. Remember about BURI Catering.

Call +63123456789 and let us know what we can do for you.\n""")

def contact\_us():

print("O"+"="\*102+"O")

print("""\tNatakam ka ba? Hindi mo pa na try? Contact us here!

\tFacebook: Buri Catering

\tTwitter: @BuriCatering

\tContact Number: +63917611539

+63123456789

\tInterested at Buri Catering? Please message or look for one of the members below:

\tFounder:

\tVladimir Borja

\tCarl Paldeng

\tCo-founder:

\tMark Erolle Quiambao""")

def terms\_of\_payment():

print("O"+"="\*44+"Terms of Payment"+"="\*42+"O")

print("\n"," "\*42, "Catering Agreement\n")

print(""" Catering Agreement/General Agreement for Buri.ph. The catering agreement will be

created once the detail of the client is provided. The client will be required to describe

the following details: name, date, time, guest count etc. They must also concur with the

following:

\t1. DEPOSIT OF PAYMENT

Fifty percent (50%) of the anticipated cost (COST) is required for order processing. If the

order goes through, the deposit amount will be deducted from the final payment.\n

\t2. FINAL PAYMENT

The final payment is due three to five days before the event. The payment terms must be

adhered to.\n

\t3. PAYMENT METHOD

All orders are payable with GCASH or CASH. GCASH DETAILS.\n

\t4. GUEST COVERAGE

A final headcount or guest list should be provided at the time of order to avoid further

miscalculation.\n

\t5. LEFTOVER

Guests who have leftover foods will be discarded by the Caterer.\n

\t6. BEVERAGES

The guests will be entertained with beverages, or support as agreed upon the agreement.\n

\t7. STRICTLY NO CANCELLATION OF ORDER

All pre-payments and post-payments won’t be refunded if the client cancels the event and

notifies the venue within one to three days prior to the event.\n

\t8. VENUE

The caterer is not responsible for any damage or loss to furnishings, equipment, or clothing

from the said venue. However, damages or loss of our products, equipment and more will be the

customer’s responsibility and must be charged for the damaged items.\n

""")

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

#-----------------------------Party Food----------------------------------#

def party\_options():

print("O"+"="\*38+"\*\*\*PARTY PACK OPTIONS\*\*\*"+"="\*40+"O")

print("""\t\t\t\t[1]Packs

\t\t\t\t[2]Cart

\t\t\t\t[3]Back""")

print("O"+"="\*102+"O")

def check\_party\_option(party\_option):

match party\_option:

case "1"|"2"|"3":

party\_option = int(party\_option)

return party\_option

def PARTY\_PACKAGES():

print("""O======================================================================================================O

| PACKAGE CODE [1] | PACKAGE CODE [2] | PACKAGE CODE [3] |

| Package 1: Occasional Package | Package 2: Party Package Overload | Package 3: Formal Package |

| (Good for 20 People) | (Good for 20 People) | (Good for 20 People) |

| Mixed Veggies and Seafood | Fried Garlic Rice | Roast Beef |

| Spaghetti & Meatballs | Hotdogs | Fried garlic and butter rice |

| Plain Rice | Crispy Chicken Tacos | Chicken Cordon Bleu |

| Garlic Chicken | Pizza Hawaiian | Roast Pork with Oyster Sauce |

| Roast Beef | Chicken Wings | Pescada en Salsa Verde |

| Pork and Chicken BBQ | BBQ Pork Sliders | Baked Spaghetti |

| Fish Fillet | Spaghetti & Meatballs | Mixed Vegetables with Ham |

| Pork Lumpia | Sandwiches Roasted Beef | Buko Fruit Salad |

| Iced Tea Bottomless | Iced Tea Bottomless | Iced Tea Bottomless |

| Ice Mineral Water | Ice Mineral Water | Ice Mineral Water |

| Price: P 10,261 | Price: P 11,707 | Price: P 12,404 |

O======================================================================================================O""")

def show\_cart\_partypacks():

print(" "\*41+"\*\*\*Party Food Cart\*\*\*")

if total\_party\_pack != 0:

print("Package Ordered", "\t\t\t\t\tQuantity","\t\t\tPrice")

if party\_pack1 != 0:

print("Party Package 1 [Occasional Package]:","\t\t\t",party\_pack1, "\t\t\t\tP",format(pp1, ',.2f'))

if party\_pack2 != 0:

print("Party Package 2 [Party Package Overload]:","\t\t",party\_pack2, "\t\t\t\tP",format(pp2, ',.2f'))

if party\_pack3 != 0:

print("Party Package 3 [Formal Package]:", "\t\t\t",party\_pack3, "\t\t\t\tP",format(pp3, ',.2f'))

print("\t "\*8,"Party Pack Total: P", format(total\_party\_pack, ',.2f'))

else:

print("\t\t\tThere are no current items in your shopping cart.\n")

party\_pack1,party\_pack2,party\_pack3 = 0,0,0 #NUMBER OF PARTY PACKS ORDERED

price\_pp1,price\_pp2,price\_pp3 = 10261.00, 11707.00, 12404.00 #PRICES OF EACH PARTY PACKS

pp1,pp2,pp3 = 0.00,0.00,0.00 #DISPLAYS THE TOTAL OF EACH ITEM ORDERED IN EQUIPMENT RENTAL

total\_party\_pack = 0.00 #TOTAL IN PARTY PACKS

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

#------------------------RENTALS AND PARTY NEEDS FUNCTIONS---------------#

def rental\_options():

print("O"+"="\*33+"\*\*\*RENTAL AND EQUIPMENT OPTIONS\*\*\*"+"="\*35+"O")

print("""\t\t\t\t[1] Packs

\t\t\t\t[2] Individal Items

\t\t\t\t[3] Cart

\t\t\t\t[4] Back""")

print("O"+"="\*102+"O")

def rentalMenuPacks():

print("O"+"="\*38+"\*\*\*RENTAL PACK OPTIONS\*\*\*"+"="\*39+"O")

print("""\t\t[1]Complete Catering Equipment Rental Package - 100/person

\t\t[2]Complete Catering Equipment Rental Package with Waiters - 130/person

\t\t[3]Complete Catering Equipment with Flower Arrangement and Waiters - 160/person""")

print("O"+"="\*102+"O")

def rentalMenuItems():

print("O"+"="\*38+"\*\*\*RENTAL ITEM OPTIONS\*\*\*"+"="\*39+"O")

print("""\t\t[1]Tents - 100/pc

\t\t[2]Tables - 100/pc

\t\t[3]Chairs - 20/pc

\t\t[4]Table Cloth Linen - 50/pc

\t\t[5]Ribbons - 15/pc

\t\t[6]Complete Utensils and Silverware - 50/pc""")

print("O"+"="\*102+"O")

def show\_cart\_rentals():

print(" "\*39+"\*\*\*Equipment Rental Cart\*\*\*")

if total\_rentals != 0:

print("Package Ordered", "\t\t\t\t\tQuantity","\t\t\tPrice")

if rental\_pack1 != 0:

print("Complete Catering Eq. Rental Package:", "\t\t\t",rental\_pack1, "\t\t\t\tP", format(p1, ',.2f'))

if rental\_pack2 != 0:

print("Complete Catering Eq. Rental Package w/ Waiters:","\t",rental\_pack2, "\t\t\t\tP", format(p2, ',.2f'))

if rental\_pack3 != 0:

print("Complete Catering Eq. w/ Flower Arrangement & Waiters:", "\t",rental\_pack3, "\t\t\t\tP", format(p3, ',.2f'))

if item1 != 0:

print("Tents:","\t\t\t\t\t\t\t", item1, "\t\t\t\tP", format(p4, ',.2f'))

if item2 != 0:

print("Tables:","\t\t\t\t\t\t", item2, "\t\t\t\tP", format(p5, ',.2f'))

if item3 != 0:

print("Chairs:","\t\t\t\t\t\t", item3, "\t\t\t\tP", format(p6, ',.2f'))

if item4 != 0:

print("Table Cloth Linen:","\t\t\t\t\t", item4, "\t\t\t\tP", format(p7, ',.2f'))

if item5 != 0:

print("Ribbons:", "\t\t\t\t\t\t",item5, "\t\t\t\tP", format(p8, ',.2f'))

if item6 != 0:

print("Complete Utensils and Silverware:", "\t\t\t",item6, "\t\t\t\tP", format(p9, ',.2f'))

print("\t "\*8,"Equipment Total: P", format(total\_rentals,',.2f'))

else:

print("\t\t\tThere are no current items in your shopping cart.\n")

rental\_pack1,rental\_pack2,rental\_pack3 = 0,0,0 #NUMBER OF RENTAL PACKS ORDERED

price\_rp1,price\_rp2,price\_rp3 = 100.00, 130.00, 160.00 #PRICES OF EACH EQUIPMENT PACKS

item1,item2,item3,item4,item5,item6 = 0,0,0,0,0,0 #NUMBER OF RENTAL ITEMS ORDERED

price\_i1,price\_i2,price\_i3,price\_i4,price\_i5,price\_i6=100.00,100.00,20.00,50.00,15.00,50.00 #PRICES OF EACH EQUIPMENT ITEMS

p1,p2,p3,p4,p5,p6,p7,p8,p9 = 0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00 #DISPLAYS THE TOTAL OF EACH ITEM ORDERED IN EQUIPMENT RENTAL

total\_rentals = 0.00 #TOTAL IN RENTAL AND EQUIPMENT PACKS/ITEMS

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

#-----------------------------Catering Services---------------------------#

def catering\_options():

print("O"+"="\*40+"\*\*\*CATERING SERVICES\*\*\*"+"="\*39+"O")

print("""\t\t\t\t[1] Wedding Packages

\t\t\t\t[2] Birthday Packages

\t\t\t\t[3] Cart

\t\t\t\t[4] Back""")

print("O"+"="\*102+"O")

def WEDDING\_PACKAGES():

print("""O======================================================================================================O

| WEDDING PACKAGE | Set-up and Props Inclusion | Wedding Package |

| 1 Rice Menu | Buffet Set-up | |

| 1 Fish Menu | Formal Uniform for Caterers | Package [1] - P 24,000 |

| 1 Chicken Menu | Complete Wedding Equipments | Good for 30 People |

| 1 Beef Menu | Entrance Arch , Red Carpet, etc. | |

| 1 Vegetable Menu | Invitation Cards & A Special Gift | Package [2] - P38,000 |

| 1 Dessert Menu | Complete Equipment for | Good for 50 People |

| 1 Soup Menu | Catering (Utensils) | |

| Round Bottomless Softdrinks | Complete Catering Equipments | Package [3] - P75,000 |

| Purified Water With Tube Ice | Wedding Party Host and Co-Host | Good for 100 People |

O======================================================================================================O""")

def BIRTHDAY\_PACKAGES():

print("""O======================================================================================================O

| BIRTHDAY PACKAGE | Set-up and Props Inclusion | Birthday Packages |

| Plain Rice | Buffet Set-up | |

| Garlic Rice | Formal Uniform for Caterers | Good for N People |

| 1 Beef Menu | Presentable Backdrop with | |

| 1 Chicken Menu | Color Motive | Package [1] - P 18,000 |

| 1 Fish Menu | Complete Equipment for | 30 People |

| 1 Vegetable Menu | Catering (Utensils) | |

| 1 Appetizer Dish | Complete Tables and Chairs | Package [2] - P 28,000 |

| 1 Dessert | with Presentable Center Piece | 50 People |

| 1 Soup | and Color Motive | |

| Purified Water W/Tube Ice | Presentable Cake Table/Stand | Package [3] - P 55,000 |

| Bottomless Softdrinks | Set-up and Photobooth | 100 People |

O======================================================================================================O

| BIRTHDAY PACKAGE - KIDDIE | Set-up and Props Inclusion | Birthday Packages - Kiddie |

| Plain Rice | Loot Bags and Accent Lights | |

| 1 Beef Menu | Buffet Set-up | Package [4] - P 13,000 |

| 1 Vegetable Menu | Formal Uniform for Caterers | Good for 30 People |

| Spaghetti | Party Props (Hats, Lights, etc.) | |

| Hotdog | Complete Equipment for | Package [5] - P 21,000 |

| Cotton Candy Station | Catering (Utensils) | Good for 50 People |

| Kiddie Cake | Complete Tables and Chairs | |

| 1 Round Bottomless Softdrinks | with Presentable Center Piece | Package [6] - P 38,000 |

| Purified Water With Tube Ice | and Color Motive | Good for 100 People |

O======================================================================================================O""")

def show\_cart\_catering():

print(" "\*42+"\*\*\*Catering Cart\*\*\*")

if total\_catering\_packs != 0:

print("Package Ordered", "\t\t\t\t\tQuantity","\t\t\tPrice")

if wed\_pack1 != 0:

print("Wedding Package 1:\t\t\t\t\t", wed\_pack1, "\t\t\t\tP", format(c1, ',.2f'))

if wed\_pack2 != 0:

print("Wedding Package 2:\t\t\t\t\t", wed\_pack2, "\t\t\t\tP", format(c2, ',.2f'))

if wed\_pack3 != 0:

print("Wedding Package 3:\t\t\t\t\t", wed\_pack3, "\t\t\t\tP", format(c3, ',.2f'))

if b\_pack1 != 0:

print("Birthday Package 1:\t\t\t\t\t", b\_pack1, "\t\t\t\tP", format(c4, ',.2f'))

if b\_pack2 != 0:

print("Birthday Package 2:\t\t\t\t\t", b\_pack2, "\t\t\t\tP", format(c5, ',.2f'))

if b\_pack3 != 0:

print("Birthday Package 3:\t\t\t\t\t", b\_pack3, "\t\t\t\tP", format(c6, ',.2f'))

if b\_pack4 != 0:

print("Birthday Package 4 - Kiddie:\t\t\t\t", b\_pack4, "\t\t\t\tP", format(c7, ',.2f'))

if b\_pack5 != 0:

print("Birthday Package 5 - Kiddie:\t\t\t\t", b\_pack5, "\t\t\t\tP", format(c8, ',.2f'))

if b\_pack6 != 0:

print("Birthday Package 6 - Kiddie:\t\t\t\t", b\_pack6, "\t\t\t\tP", format(c9, ',.2f'))

print("\t"\*8,"Catering Pack Total: P", format(total\_catering\_packs, ',.2f'))

else:

print("\t\t\tThere are no current items in your shopping cart.\n")

wed\_pack1,wed\_pack2,wed\_pack3 = 0,0,0 #NUMBER OF WEDDING PACKS ORDERED

price\_wp1,price\_wp2,price\_wp3 = 24000.00, 38000.00, 75000.00 #PRICES OF EACH WEDDING PACKS

b\_pack1,b\_pack2,b\_pack3,b\_pack4,b\_pack5,b\_pack6 = 0,0,0,0,0,0 #NUMBER OF BIRHTDAY PACKS ORDERED

price\_bp1,price\_bp2,price\_bp3,price\_bp4,price\_bp5,price\_bp6=18000.00,28000.00,55000.00,13000.00,21000.00,38000.00 #PRICES OF EACH BIRHTDAY PACKS

c1,c2,c3,c4,c5,c6,c7,c8,c9 = 0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00 #DISPLAYS THE TOTAL OF EACH ITEM ORDERED IN CATERING SERVICES

total\_catering\_packs = 0.00 #TOTAL IN CATERING SERVICES PACKS

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

#------------------------------ALL CART CONCLUSION------------------------------------------#

def confirm\_checkout():

print(("\t\t\t\t[1]CHECKOUT\n\t\t\t\t[2]RESET ALL ORDERS\n\t\t\t\t[3]BACK"))

print("O"+"="\*102+"O")

while True:

checkout = input("Input Operation[1-3]: ")

match checkout:

case "1"|"2"|"3":

return int(checkout)

break

case \_:

print("Invalid Input")

def details():

print("O"+"="\*102+"O")

print("Place your details and accurate data. Invalid information will not be processed by the management.")

name = input("Full Name: ")

number = input("Contact Number: ")

address = input("Venue Address: ")

date\_order = input("Input date of order (Ex. January 3,2022): ")

date\_req = input("Input date of order delivery (Ex. January 13, 2022): ")

guest\_count = input("Guest Count: ")

payment = input("Mode of Payment [GCASH/Paymaya]: ")

print("O"+"="\*102+"O")

print("\n\t\t\t\tFull Name:",name,"\n\t\t\t\tContact Number:", number,

"\n\t\t\t\tVenue Address:",address,"\n\t\t\t\tDate of order:",date\_order,

"\n\t\t\t\tDate of order deadline:",date\_req,"\n\t\t\t\tGuest Count:",guest\_count,

"\n\t\t\t\tMode of Payment:",payment,"\n\n\t\t\t\tTotal Payment P:", format(total\_cart, ',.2f'))

print("\n Make sure to print out all details accurately as our management will check the data submitted.")

print(" Any form of innacuracy or invalid data will not be processed. Thank you for your understanding.")

print("O"+"="\*102+"O")

return name

def final\_confirmation():

print("\t\t\t\tHave you confirmed your details?\n\t\t\t\t[1] Yes\t[2] Reset my details")

print("O"+"="\*102+"O")

while True:

confirmation = input("Input Operation: ")

match confirmation:

case "1"|"2":

break

case \_:

print("Invalid Option")

return int(confirmation)

def final\_output():

print("O"+"="\*102+"O")

print("\n\tThank you", name, """for placing this order. Your order will be processed and you will

\tbe contacted by one of our staff within 1-3 business days.""")

print("\n\tYour bill came to a total of P", format(total\_cart, ',.2f'),

".\n\tPlease prepare 50% of your total bill which is" , format(total\_cart/2, ',.2f'),

"php\n\tfor the processing of your order.")

print("""\n\tOnce your order is processed, an agreement will be provided for \n\tservice finalization.\n

Thank you for being our valued customer! We’ll make sure that you will have a blast!\n""")

def order\_again():

print("O"+"="\*102+"O")

print("\t\t\t\tDo you want to create another order?\n\t\t\t\t[1]Yes\t[2] No")

print("O"+"="\*102+"O")

while True:

order\_again = input("Input Operation: ")

match order\_again:

case "1"|"2":

break

case \_:

print("Invalid Option")

return int(order\_again)

#----------------------MAIN CODE WHERE THE MAGIC HAPPENS--------------------#

while True:

MENU()

main\_opt = input("Input Operation[0-5]: ")

main\_option = check\_main(main\_opt)

if main\_option == 1:

while True:

information\_options()

select = input("Input Operation[1-4]: ")

opt = four\_option(select)

if opt == 1:

about\_us()

BACK()

elif opt == 2:

contact\_us()

BACK()

elif opt == 3:

terms\_of\_payment()

BACK()

elif opt == 4:

break

#If opt is 4, break the information options look and go back to MENU()

else:

print("Invalid Input")

#Once done execetuing if-else, loop back to information\_options()

elif main\_option == 2:

while True:

party\_options()

answer = input("Input Operation[1-3]: ")

party\_option = check\_party\_option(answer)

if party\_option == 1:

PARTY\_PACKAGES()

ask\_order\_q = ask\_if\_order\_q()

if ask\_order\_q == "O":

party\_code = pack\_option()

quantity = numberOfItems()

if party\_code == 1:

total\_party\_pack += price\_pp1\*quantity

party\_pack1 += quantity

pp1 = party\_pack1\*price\_pp1

elif party\_code == 2:

total\_party\_pack += price\_pp2\*quantity

party\_pack2 += quantity

pp2 = party\_pack2\*price\_pp2

else:

total\_party\_pack += price\_pp3\*quantity

party\_pack3 += quantity

pp3 = party\_pack3\*price\_pp3

print("\n\t\t\t\t[Successfully added to your cart!!!]\n")

## If ask\_order\_q is B, automatically loop back to party\_options()

elif party\_option == 2:

show\_cart\_partypacks()

ask\_resets = ask\_reset()

if ask\_resets == "R":

#RESET PARTY PACKS

party\_pack1,party\_pack2,party\_pack3 = 0,0,0

pp1,pp2,pp3 = 0.00,0.00,0.00

total\_party\_pack = 0.00

print("\n\t\t\t\tSuccessfully Cleared Your Party Cart!!!]\n")

## If ask\_resets is B, automatically loop back to party\_options()

elif party\_option == 3:

break

## If party\_option is 3, break the party options loop and go back to MENU()

else:

print("Invalid Input")

elif main\_option == 3:

while True:

rental\_options()

select = input("Input Operation[1-4]: ")

rental\_option = four\_option(select)

if rental\_option == 1:

rentalMenuPacks()

ask\_order\_q = ask\_if\_order\_q()

if ask\_order\_q == "O":

pack\_code = pack\_option()

quantity = numberOfItems()

if pack\_code == 1:

total\_rentals += price\_rp1\*quantity

rental\_pack1 += quantity

p1 = rental\_pack1\*price\_rp1

elif pack\_code == 2:

total\_rentals += price\_rp2\*quantity

rental\_pack2 += quantity

p2 = rental\_pack2\*price\_rp2

else:

total\_rentals += price\_rp3\*quantity

rental\_pack3 += quantity

p3 = rental\_pack3\*price\_rp3

print("\n\t\t\t\t[Successfully added to your cart!!!]\n")

##Once done ordering, loop back to rental\_options()

## If ask\_order\_q is B, automatically loop back to rental\_options()

elif rental\_option == 2:

rentalMenuItems()

ask\_order\_q = ask\_if\_order\_q()

if ask\_order\_q == "O":

item\_code = six\_option()

quantity = numberOfItems()

if item\_code == 1:

total\_rentals += price\_i1\*quantity

item1 += quantity

p4 = item1\*price\_i1

elif item\_code == 2:

total\_rentals += price\_i2\*quantity

item2 += quantity

p5 = item2\*price\_i2

elif item\_code == 3:

total\_rentals += price\_i3\*quantity

item3 += quantity

p6 = item3\*price\_i3

elif item\_code == 4:

total\_rentals += price\_i4\*quantity

item4 += quantity

p7 = item4\*price\_i4

elif item\_code == 5:

total\_rentals += price\_i5\*quantity

item5 += quantity

p8 = item5\*price\_i5

else:

total\_rentals += price\_i6\*quantity

item6 += quantity

p9 = item6\*price\_i6

print("\n\t\t\t\t[Successfully added to your cart!!!]\n")

##Once done ordering, loop back to rental\_options()

## If ask\_order\_q is B, automatically loop back to rental\_options()

elif rental\_option == 3:

show\_cart\_rentals()

ask\_resets = ask\_reset()

if ask\_resets == "R":

#RESET RENTAL PACKS

rental\_pack1,rental\_pack2,rental\_pack3 = 0,0,0

item1,item2,item3,item4,item5,item6 = 0,0,0,0,0,0

p1,p2,p3,p4,p5,p6,p7,p8,p9 = 0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00

total\_rentals = 0.00

print("\n\t\t\t\t[Successfully Cleared Your Rental Cart!!!]\n")

##Once done resetting, loop back to rental\_options()

##If ask\_reserts is B, automatically loop back to rental\_options()

elif rental\_option == 4:

break

## If rental\_option is 4, break the rental options loop and go back to MENU()

else:

print("Invalid Input")

elif main\_option == 4:

while True:

catering\_options()

response = input("Input Operation[1-4]: ")

catering\_option = four\_option(response)

if catering\_option == 1:

WEDDING\_PACKAGES()

ask\_order\_q = ask\_if\_order\_q()

if ask\_order\_q == "O":

wed\_pack\_code = pack\_option()

quantity = numberOfItems()

if wed\_pack\_code == 1:

total\_catering\_packs += price\_wp1\*quantity

wed\_pack1 += quantity

c1 = wed\_pack1\*price\_wp1

elif wed\_pack\_code == 2:

total\_catering\_packs += price\_wp2\*quantity

wed\_pack2 += quantity

c2 = wed\_pack2\*price\_wp2

else:

total\_catering\_packs += price\_wp3\*quantity

wed\_pack3 += quantity

c3 = wed\_pack3\*price\_wp3

print("\n\t\t\t\t[Successfully added to your cart!!!]\n")

##Once done ordering, loop back to catering\_options()

## If ask\_order\_q is B, automatically loop back to catering\_options()

elif catering\_option == 2:

BIRTHDAY\_PACKAGES()

ask\_order\_q = ask\_if\_order\_q()

if ask\_order\_q == "O":

bday\_pack = six\_option()

quantity = numberOfItems()

if bday\_pack == 1:

total\_catering\_packs += price\_bp1\*quantity

b\_pack1 += quantity

c4 = b\_pack1\*price\_bp1

elif bday\_pack == 2:

total\_catering\_packs += price\_bp2\*quantity

b\_pack2 += quantity

c5 = b\_pack2\*price\_bp2

elif bday\_pack == 3:

total\_catering\_packs += price\_bp3\*quantity

b\_pack3 += quantity

c6 = b\_pack3\*price\_bp3

elif bday\_pack == 4:

total\_catering\_packs += price\_bp4\*quantity

b\_pack4 += quantity

c7 = b\_pack4\*price\_bp4

elif bday\_pack == 5:

total\_catering\_packs += price\_bp5\*quantity

b\_pack5 += quantity

c8 = b\_pack5\*price\_bp5

else:

total\_catering\_packs += price\_bp6\*quantity

b\_pack6 += quantity

c9 = b\_pack6\*price\_bp6

print("\n\t\t\t\t[Successfully added to your cart!!!]\n")

##Once done ordering, loop back to catering\_options()

## If ask\_order\_q is B, automatically loop back to catering\_options()

elif catering\_option == 3:

show\_cart\_catering()

ask\_resets = ask\_reset()

if ask\_resets == "R":

#RESET CATERING PACKS

wed\_pack1,wed\_pack2,wed\_pack3 = 0,0,0

b\_pack1,b\_pack2,b\_pack3,b\_pack4,b\_pack5,b\_pack6 = 0,0,0,0,0,0

c1,c2,c3,c4,c5,c6,c7,c8,c9 = 0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00

total\_catering\_packs = 0.00

print("\n\t\t\t\t[Successfully Cleared Your Catering Cart!!!]\n")

##Once done resetting, loop back to catering\_options()

##If ask\_reserts is B, automatically loop back to catering\_options()

elif catering\_option == 4:

break

## If catering\_option is 4, break the catering options loop and go back to MENU()

else:

print("Invalid Input")

elif main\_option == 5:

if total\_party\_pack == 0 and total\_rentals == 0 and total\_catering\_packs == 0:

print("\t\t\t\tYou have no current orders")

BACK()

else:

total\_cart = total\_party\_pack + total\_rentals + total\_catering\_packs

print("O"+"="\*43+"\*\*\*Your Cart\*\*\*"+"="\*44+"O"+"\n")

if total\_party\_pack != 0:

show\_cart\_partypacks()

if total\_rentals != 0:

show\_cart\_rentals()

if total\_catering\_packs != 0:

show\_cart\_catering()

print("O"+"="\*45+"\*\*\*Total\*\*\*"+"="\*46+"O")

print("\t"\*8+"Your current total is P "+str(format(total\_cart, ',.2f')))

checkout = confirm\_checkout()

if checkout == 1:

while True:

name = details()

confirmation = final\_confirmation()

if confirmation == 1:

final\_output()

break

#If confirmation is 2, automatically loop back to name=details()

order\_again\_program = order\_again()

if order\_again\_program == 1:

#RESET PARTY PACKS

party\_pack1,party\_pack2,party\_pack3 = 0,0,0

pp1,pp2,pp3 = 0,0,0

total\_party\_pack = 0

#RESET RENTAL PACKS

rental\_pack1,rental\_pack2,rental\_pack3 = 0,0,0

item1,item2,item3,item4,item5,item6 = 0,0,0,0,0,0

p1,p2,p3,p4,p5,p6,p7,p8,p9 = 0,0,0,0,0,0,0,0,0

total\_rentals = 0

#RESET CATERING PACKS

wed\_pack1,wed\_pack2,wed\_pack3 = 0,0,0

b\_pack1,b\_pack2,b\_pack3,b\_pack4,b\_pack5,b\_pack6 = 0,0,0,0,0,0

c1,c2,c3,c4,c5,c6,c7,c8,c9 = 0,0,0,0,0,0,0,0,0

total\_catering\_packs = 0

else:

break

#If an order succeeded and the user answers 2 [No], the main loop breaks

elif checkout == 2:

#RESET PARTY PACKS

party\_pack1,party\_pack2,party\_pack3 = 0,0,0

pp1,pp2,pp3 = 0,0,0

total\_party\_pack = 0

#RESET RENTAL PACKS

rental\_pack1,rental\_pack2,rental\_pack3 = 0,0,0

item1,item2,item3,item4,item5,item6 = 0,0,0,0,0,0

p1,p2,p3,p4,p5,p6,p7,p8,p9 = 0,0,0,0,0,0,0,0,0

total\_rentals = 0

#RESET CATERING PACKS

wed\_pack1,wed\_pack2,wed\_pack3 = 0,0,0

b\_pack1,b\_pack2,b\_pack3,b\_pack4,b\_pack5,b\_pack6 = 0,0,0,0,0,0

c1,c2,c3,c4,c5,c6,c7,c8,c9 = 0,0,0,0,0,0,0,0,0

total\_catering\_packs = 0

print("\n\t\t\t\tSUCCESSFULLY CLEARED ALL OF YOUR ORDERS!!!\n")

##If checkout is 3, automatically loop back to MENU()

elif main\_option == 0:

break

else:

print("Invalid Input")

print("O"+"="\*102+"O")

print("""

\t\t\t\tThanks for using our program.

\t\t\t\tPresented by:

\t\t\t\tKarl Vladimir Borja

\t\t\t\tCarl Lester Paldeng

\t\t\t\tMark Erolle Quiambao

\t\t\t\tLuis Miguel Cayanan

""")

print("O"+"="\*102+"O")

**Roles**

|  |  |
| --- | --- |
| Member Name | Roles and Contribution |
| Borja, Karl Vladimir B. | - Program Content (Party Packages)  - Program Flow  - Program testing  - Contribute various ideas for program creation |
| Cayanan, Luis Miguel I. | - Program Code  - Functions  - Input Validation  - Menu/Options/Cart/Reset  - Program Calculations  - Program Flowchart  - Program Formatting |
| Paldeng, Carl Lester P. | - Program content (Catering Services)  - Created descriptions for Read More (About us, etc.)  - Description checker  - Program Testing  - Contribute ideas on options |
| Quiambao, Mark Erolle | - Aide members in creation of program content such as about us, party packages, etc.  - Aide in program formatting  - Aide in Program Flow  - Program testing and identification of program errors |

**References:**

**Program References**

1. “Jojie's Catering Services Manila.” Jojie's Catering Services Accessed October 29, 2022. <https://www.jojies.com/>

2. “Food Ordering System Using Python with Free Source Code Demo.” YouTube. YouTube, October 9, 2021. Accessed October 29, 2022. <https://www.youtube.com/watch?v=MZCazGDxUpk&t=248s>

3. hobokengar. “Python Try except Valueerror.” YouTube. YouTube, September 14, 2013. Accessed October 29, 2022. <https://www.youtube.com/watch?v=Yg7yxLMAb1U>

**Party Packages References**

1. “Sample Menus · Catering Works.” Catering Works, 2022. Accessed October 29, 2022. <https://www.cateringworks.com/weddings/wedding_food_and_bev/sample_menus/>

2. Egcs. “Birthday Package Diamond.” Executive Gourmet, January 29, 2020. Accessed October 29, 2022. <https://www.executivegourmet.ph/birthday-package-diamond/>

3. “Food Catering for Children Birthday Party - Kids Party Catering: Birthday Party Food: Rivera Events.” Miami Party Event Planner, April 29, 2022. Accessed October 29, 2022. <https://riveraevents.com/catering-services/kids-buffet/>

**Catering Services References**

1. “A&J Catering Services.” Facebook, 2n.d.. Accessed October 29, 2022. <https://www.facebook.com/jovy.larrios.arthur>

2. “Luxury Events, Valunzuela.” Facebook. Accessed October 29, 2022. ‘https://www.facebook.com/yourluxuryevent