WORKSHOP #5 - A LA CARTE ENTERTAINMENT

# Group 4 Members:

* **Viktor Makarov - Parts 1, 2, 6**
* **Luca Novello - Parts 3, 4, 5**

# Workshop Overview

The objective of this activity is to emphasize how much easier it is to provide solutions to larger problems by first isolating and focusing on smaller discrete parts of the problem individually and then piece those solutions together at the end to provide an overall solution to the larger problem. This approach will provide a modular solution where each smaller part works independently from other parts and performs only a very specific task. Each “part” or scenario detailed in the next section guides you on this journey by isolating one specific smaller part of the problem. Incrementally, with each scenario, a new smaller problem will be solved and by the end of the 6th scenario, an overall solution will have been developed.

# Workshop Details

A La Carte Entertainment is a place where people can see stage productions (theatrical) or musicals. Each type of production has a different ticket cost, and each production can be seen during the day (Matinée) or in the evening. The time of day the production is run influences the ticket cost. To gain access to the venue, a cover charge is applied. Discounts are offered based on the number of people in a group. As optional services, A La Carte Entertainment also offers a small-scale snack bar and valet parking. An online mobile application system is needed to track the costs generated by each patron and provide an overall itemized receipt of costs incurred for the time spent at A La Carte Entertainment. Use the pricelist described below later in this document to extract the possible options for each part of the system. The online mobile application system should implement a pre-ordering feature where most of the costs can be predetermined prior to arriving to the complex. To secure the reservation, a deposit of $50 will be required and a QR code is generated to uniquely identify the order. As you go through the computational thinking approach to problem solving this workshop, keep in mind the following:

* Data Structure definition something that will track all required information (hint: Reverse engineer what you need to display in a detailed receipt)
* Create a variable that is the data type of the data structure defined above! This will be used to "send" and "update" as it is passed to other sub-processes (functions)
* Call sub-processes (functions) in a logical sequence as it should occur (these will be the 6 main parts to the workshop!)

# DATA STRUCTURE

Order

* ticketCost
* coverCost
* customerTotal
* customerAges
* discount
* valetCost
* snackCost

# 

# PSEUDOCODE

## Overall Pseudocode:

1. Start
2. Customer creates an account
3. Create and store Order
4. Ask customer how many patrons and store in Order
5. Part 1
6. Part 2
7. Part 3
8. Part 4
9. Process $50 deposit from customer credit card
10. Part 5
11. Does the customer want to leave a valet tip?
    1. Yes: add tip to Order.valvetCosts: continue step
12. Wait customer to make a payment
13. Scan QR to confirm payment
14. Did the customer make a payment in full?
    1. No: go to step: 12
15. Part 6
16. Permit customer to leave
17. End

## 

## Logic 1 - Pseudocode:

### Part-5 (Valet Parking Charges)

#### Pseudocode:

1. Start/Previous Step
2. Import and store **Order**
3. Does the customer want to purchase valet parking?
   1. *Yes: continue (step: 4)*
   2. *No: add 0 to* Order.valetCosts*: continue (step: 7)*
4. Does the customer want “Matinée Standard” parking?
   1. *Yes: add 10 to* Order.valetCosts*: continue (step: 7)*
   2. *No: continue (step: 5)*
5. Does the customer want “Matinée VIP” parking?
   1. *Yes: add 15 to* Order.valetCosts*: continue (step: 7)*
   2. *No: continue (step: 6)*
6. Does the customer want “Evening Standard” parking?
   1. *Yes: add 20 to* Order.valetCosts*: continue (step: 7)*
   2. *No: add 30 to* Order.valetCosts*: continue (step: 7)*
7. Output **Order**
8. End/Next Step

#### Tests:

| **Conditions** | **Test** |
| --- | --- |
| Standard tier valet service during the matinée time frame with a $4.75 tip. | *Start → Matinée Standard parking?* ***Yes:*** *add 10 to Order.valetCosts**→ Leave a tip?* ***Yes - $4.75****→ Output Order.valetCosts → End* |
| Standard tier valet service during the evening time frame without a tip. | *Start → Matinée Standard parking?* ***No*** *→ Matinée VIP parking?* ***No*** *→ Evening Standard parking?* ***Yes:*** *add 20 to Order.valetCosts**→ Leave a tip?* ***No*** *→ Output Order.valetCosts → End* |
| VIP tier valet service during the matinée time frame with a $2.50 tip | *Start → Matinée Standard parking?* ***No*** *→ Matinée VIP parking?* ***Yes:*** *add 15 to Order.valetCosts**→ Leave a tip?* ***Yes - $2.50*** *→ Output Order.valetCosts → End* |
| VIP tier valet service during the evening time frame without a tip. | *Start → Matinée Standard parking?* ***No*** *→ Matinée VIP parking?* ***No*** *→ Evening Standard parking?* ***No*** *→ Evening VIP parking? add 30 to Order.valetCosts**→ Leave a tip?* ***No*** *→ Output Order.valetCosts → End* |

### 

### Part-6 (Display Mobile Receipt: Itemized Listing of All Charges)

#### Pseudocode:

1. Start.
2. Get order data by QR
3. Add cover charge to Receive
4. Is there a snack?
   1. Yes: add snack to Receive
5. Is there a Valet charge?
   1. Yes: add Valet charge to Receive
6. Is there a discount?
   1. Yes: add discount to Receive
7. Print Recieve
8. End

#### Tests:

| **Test** | **Cover Charges** | **Ticket Charges** | **Snack Charge** | **Valet Charges** | **Discounts** |
| --- | --- | --- | --- | --- | --- |
| 1 | Yes/Show | Yes/Show | - | - | - |
| 2 | Yes/Show | Yes/Show | Yes/Show | - | Yes/Show |
| 3 | Yes/Show | Yes/Show | - | Yes/Show | - |
| 4 | Yes/Show | Yes/Show | Yes/Show | Yes/Show | Yes/Show |

## 

## 

## Logic 2 - Pseudocode:

### Part-1 (Cover Charges) Pseudocode:

1. Start
2. Import and store “Order”
3. Is there another person?
   1. *Yes: continue (step: 5)*
   2. *No: continue (step: 7)*
4. Is he < 18?
   1. *Yes: Add 0.75 to “ Order.coverCost”: continue (step: 4)*
   2. *No: continue (step: 6)*
5. Is he < 65?
   1. *Yes: Add 4 to “Order.coverCost”: continue (step: 4)*
   2. *No: Add 1.5 to “Order.coverCost”: continue (step: 4)*
6. Return “*Order.coverCost*” / Output “*Order.coverCost*”
7. End

#### Tests:

| **Conditions** | **Test** |
| --- | --- |
| Three seniors, one adult, and two children enter the venue paying the necessary cover charges. | *Start -> Import -> Create -> Yes -> No -> No -> add 1.5 -> Yes -> No -> No -> add 1.5-> Yes -> No -> No -> add 1.5 -> Yes -> No -> Yes -> add 4 -> Yes -> Yes -> add 0.75 -> Yes -> Yes -> add 0.75 -> No -> output Cover Charge (10) -> End.* |
| One adult enters the venue paying the necessary cover charge. | *Start -> Import -> Create Cover Charge -> Yes -> No -> Yes -> add 4 -> No -> output Cover Charge (10) -> End.* |

### 

### Part-2 (Ticket Charges)

#### Pseudocode:

1. Start
2. Import and store “Order”
3. the event in the Matinee?
   1. *Yes: continue (step: 5)*
   2. *No: continue (step: 6)*
4. Is it a musical event?
   1. *Yes: Add 70 to “Order.ticketCost”: continue (step: 7)*
   2. *No: Add 55 to “Order.ticketCost”: continue (step: 7)*
5. Is it a musical event?
   1. *Yes: Add 150 to “Order.ticketCost”: continue (step: 7)*
   2. *No: Add 115 to “Order.ticketCost”: continue (step: 7)*
6. Return *Order*

#### Tests:

| **Conditions** | **Test** |
| --- | --- |
| One senior, two adults, and one child go to the A La Carte venue at 8:00 PM to see the musical “Trump Squeaks”. Using your process determine how much the ticket charges cost. | *Start -> Import -> Create Ticket Charge - > No ->Yes -> add 150 -> Cover Charge = 150 \* 4 -> output Cover Charge 600 -> End.* |
| One senior, two adults, and one child go to the A La Carte venue at 3:00 PM to see the musical “Doug Ford Barks”. Using your process determine how much the ticket charges cost. | *Start -> Import -> Create Ticket Charge - > Yes ->Yes -> add 70 -> Cover Charge = 70 \* 4 -> output Cover Charge 280 -> End.* |
| One senior, two adults, and one child go to the A La Carte venue at 8:00 PM to see the theater production of “Saw 1 – Where it Began”. Using your process determine how much the ticket charges cost. | *Start -> Import -> Create Ticket Charge - > No ->No -> add 115 -> Cover Charge = 115 \* 4 -> output Cover Charge 460 -> End.* |
| One senior, two adults, and one child go to the A La Carte venue at 2:00 PM to see the theater production of “Jaws 1 – Swim with Me”. Using your process determine how much the ticket charges cost. | *One senior, two adults, and one child go to the A La Carte venue at 2:00 PM to see the theater production of “Jaws 1 – Swim with Me”. Using your process determine how much the ticket charges cost.* |

## 

## Logic 3 - Pseudocode:

### Part-3 (Discounts)

#### Pseudocode:

1. Start/Previous Step.
2. Import and store Order
3. Is “Order.customerTotal” < 2?
   1. *Yes: add 0 to* **Order.discountTotal** *: continue (step: 7)*
   2. *No: continue (step: 4)*
4. Is “Order.customerTotal” < 3?
   1. *Yes: add (ticketTotal x 0.05) to* ***Order.discountTotal****: continue (step: 7*
   2. *No: continue (step: 5)*
5. Is “Order.customerTotal” < 4?
   1. *Yes: add (ticketTotal x 0.15) to* **Order.discountTotal***: continue (step: 7)*
   2. *No: continue (step: 6)*
6. Is “Order.customerTotal” < 5?
   1. *Yes: add (ticketTotal x 0.2) to* **Order.discountTotal***: continue (step: 7)*
   2. *No: add (ticketTotal x 0.3) to* **Order.discountTotal***: continue (step: 7)*
7. Output “discount”
8. End/Next Step

#### Tests:

| **Conditions** | **Test** |
| --- | --- |
| One senior goes to the A La Carte venue at 8:00 PM to see the musical “The Walking Dead”. Using your process determine the discount. | *Start → Number Of Customers < 2?* ***Yes: Create a new Charge, set name to “discount” and amount to 0*** *→ Output discount → End* |
| One senior, and two adults go to the A La Carte venue at 3:00 PM to see the theater production of “Daenerys Rides Dragons”. Using your process determine the discount. | *Start → Number Of Customers < 2?* ***No*** *→ Number Of Customers < 3?* ***No*** *→ Number Of Customers < 4?* ***Yes: Create new Charge, set name to “discount” and amount to (ticketTotal x 0.15)*** *→ Output discount → End* |
| Two adults, and five children go to the A La Carte venue at 8:00 PM to see the musical “APS145 Puts Logic to Music!” Using your process determine the discount. | *Start → Number Of Customers < 2?* ***No*** *→ Number Of Customers < 3?* ***No*** *→ Number Of Customers < 4?* ***No*** *→ Number Of Customers < 5?* ***No: Create new Charge, set name to “discount” and amount to (ticketTotal x 0.3)*** *→ Output discount → End* |

### 

### **Part-4 (Snack Charges)**

#### Pseudocode:

1. Start/Previous Step
2. Import and store Order
3. Does the customer want to purchase a snack?
   1. *Yes: continue (step: 4)*
   2. *No: add 0 to Order.snackCosts: continue (step: 7)*
4. Are customer account details imported and stored?
   1. *Yes: continue (step: 5)*
   2. No: Verify **QR code**: continue **(step: 5)**
5. Did the customer order a "Non-Alcoholic Beverage"?
   1. *Yes: add* ***2.25*** *to* **Order.snackCosts***: continue (step: 3)*
   2. *No: continue (step: 6)*
6. Did the customer order a "Alcoholic Beverage"?
   1. *Yes: add 7.5 to* **Order.snackCosts***: continue (step: 3)*
   2. *No: add 1.75 to* **Order.snackCosts***: continue (step: 3)*
7. Output **Order.snackCosts**
8. End/Next Step

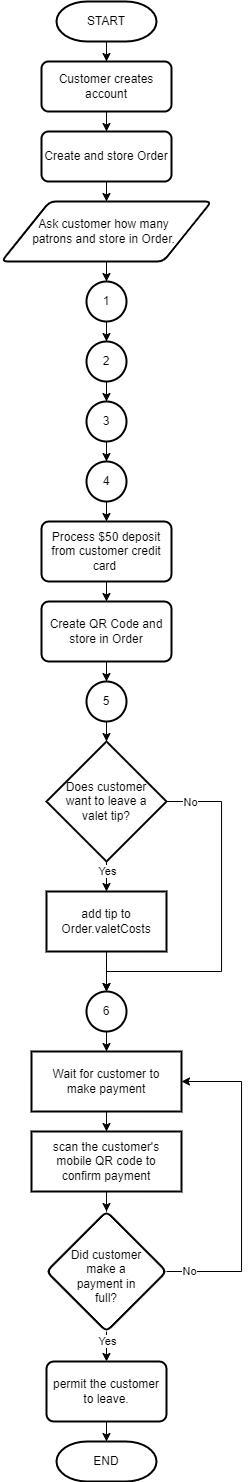
#### Tests:

| **Conditions** | **Test** |
| --- | --- |
| Purchase of no snacks. | *Start → purchase a snack?* ***No:*** *add 0 to Order.snackCosts**→ Output snackCosts:* ***0*** *→ End* |
| Purchase of 4 soft-drinks (Coke, Pepsi, Ginger Ale and Fruitopia). Using your process, determine the cost. | *Start → purchase a snack?* ***Yes*** *→ "Non-Alcoholic Beverage"?* ***Yes:*** *add* ***2.25*** *to* **Order.snackCosts***→ done ordering?* ***No*** *→ "Non-Alcoholic Beverage"?* ***Yes:*** *add* ***2.25*** *to* **Order.snackCosts***→ done ordering?* ***No*** *→ "Non-Alcoholic Beverage"?* ***Yes:*** *add* ***2.25*** *to* **Order.snackCosts***→ done ordering?* ***No*** *→ "Non-Alcoholic Beverage"?* ***Yes:*** *add* ***2.25*** *to* **Order.snackCosts***→ done ordering?* ***No*** *→ Output snackCosts:* ***4 x 2.25 = 9*** *→ End* |
| Purchase of 2 soft-drinks, 2 beers, and 2 wines. Using your process, determine the cost. | *Start → purchase a snack?* ***Yes*** *→ "Non-Alcoholic Beverage"?* ***Yes: add 2.25 to Order.snackCosts*** *→ done ordering?* ***No*** *→ "Non-Alcoholic Beverage"?* ***Yes: add 2.25 to Order.snackCosts*** *→ done ordering?* ***No*** *→ "Non-Alcoholic Beverage"?* ***No*** *→ "Alcoholic Beverage"?* ***Yes:*** *add 7.5 to* **Order.snackCosts***→ done ordering?* ***No*** *→ "Non-Alcoholic Beverage"?* ***No*** *→ "Alcoholic Beverage"?* ***Yes:*** *add 7.5 to* **Order.snackCosts***→ done ordering?* ***No*** *→ "Non-Alcoholic Beverage"?* ***No*** *→ "Alcoholic Beverage"?* ***Yes:*** *add 7.5 to* **Order.snackCosts***→ done ordering?* ***No*** *→ "Non-Alcoholic Beverage"?* ***No*** *→ "Alcoholic Beverage"?* ***Yes:*** *add 7.5 to* **Order.snackCosts***→ done ordering?* ***Yes*** *→ Output snackCosts:* ***(2 x 2.25) + (4 x 7.5) = 34.5*** *→ End* |
| Purchase of 1 wine and 4 chocolate bars. | *Start → purchase a snack?* ***Yes*** *→ "Non-Alcoholic Beverage"?* ***No*** *→ "Alcoholic Beverage"?* ***Yes:*** *add 7.5 to* **Order.snackCosts***→ done ordering?* ***No*** *→ "Non-Alcoholic Beverage"?* ***No*** *→ "Alcoholic Beverage"?* ***No*** *→ "Misc. Candy Item"?* ***Yes:*** *add 1.75 to* **Order.snackCosts** *done ordering?* ***No*** *"Non-Alcoholic Beverage"?* ***No*** *→ "Alcoholic Beverage"?* ***No*** *→ "Misc. Candy Item"?* ***Yes:*** *add 1.75 to* **Order.snackCosts** *done ordering?* ***No*** *"Non-Alcoholic Beverage"?* ***No*** *→ "Alcoholic Beverage"?* ***No*** *→ "Misc. Candy Item"?* ***Yes:*** *add 1.75 to* **Order.snackCosts** *done ordering?* ***No*** *"Non-Alcoholic Beverage"?* ***No*** *→ "Alcoholic Beverage"?* ***No*** *→ "Misc. Candy Item"?* ***Yes:*** *add 1.75 to* **Order.snackCosts** *done ordering?* ***Yes****→ Output snackCosts:* ***(1 x 7.5) + (4 x 1.75) = 14.5*** *→ End* |

#### 

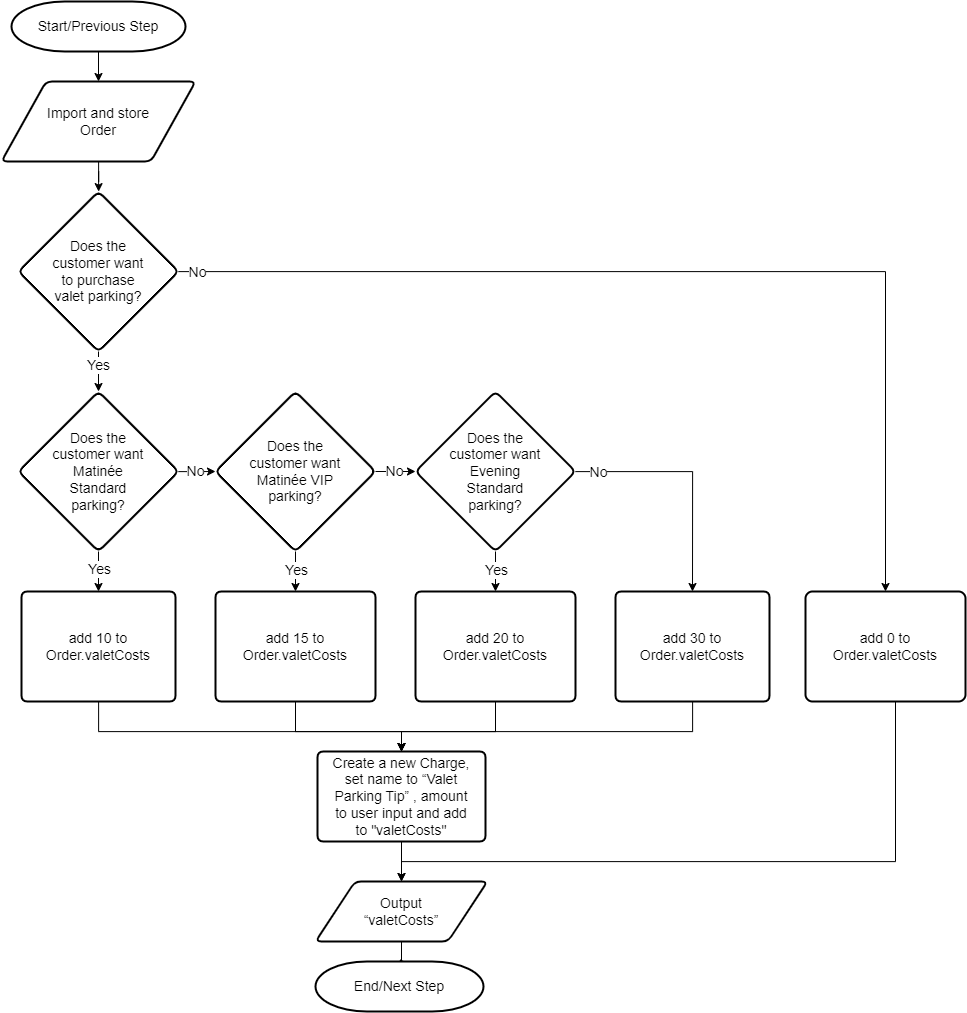
# FLOWCHARTS:

## Overall Flowchart:



## Logic 1 Flowcharts:

### Part-5 (Valet Parking Charges) Flowcharts:

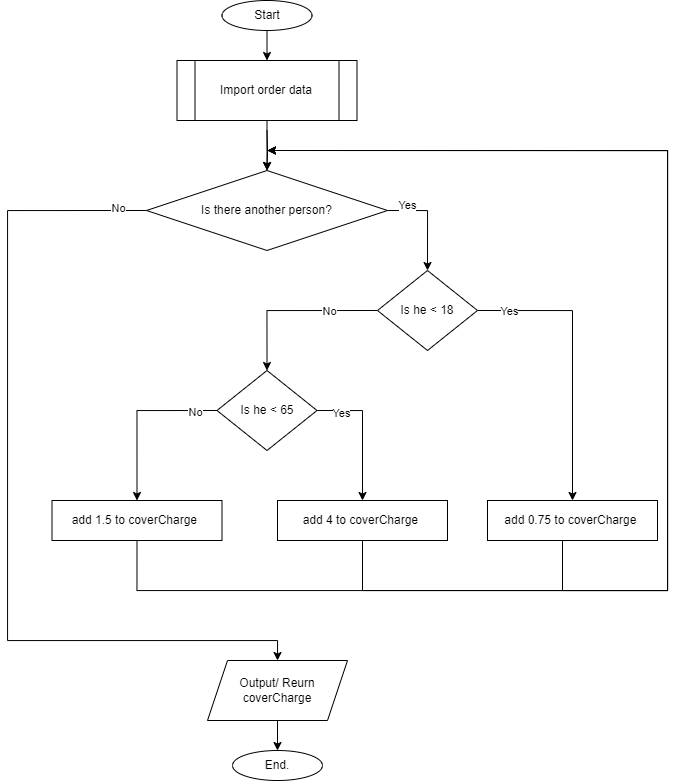


### Part-6 (Display Mobile Receipt: Itemized Listing of All Charges) Flowcharts:

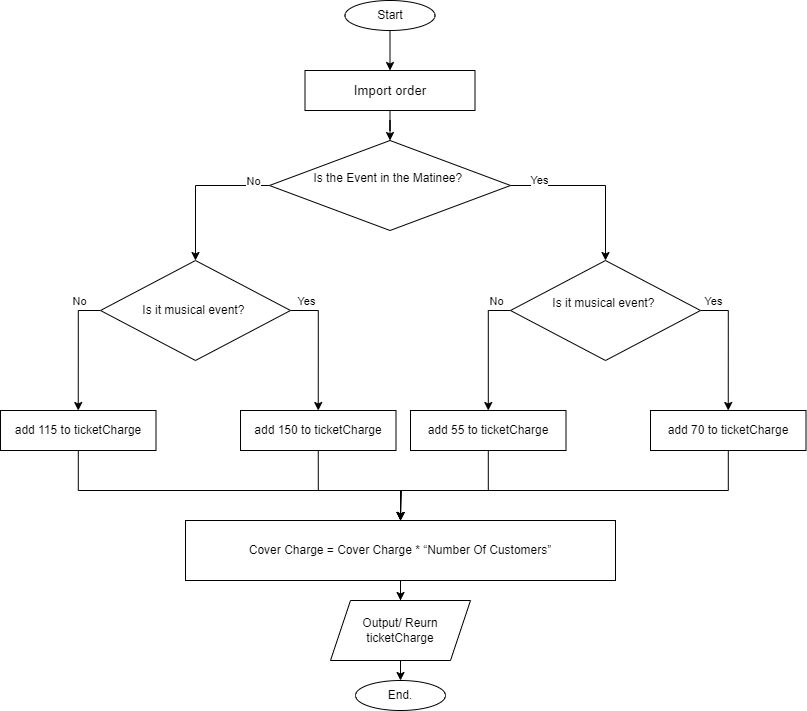
### 

## Logic 2 Flowcharts:

### Part-1 (Cover Charges) Flowcharts:



### Part-2 (Ticket Charges) Flowcharts:



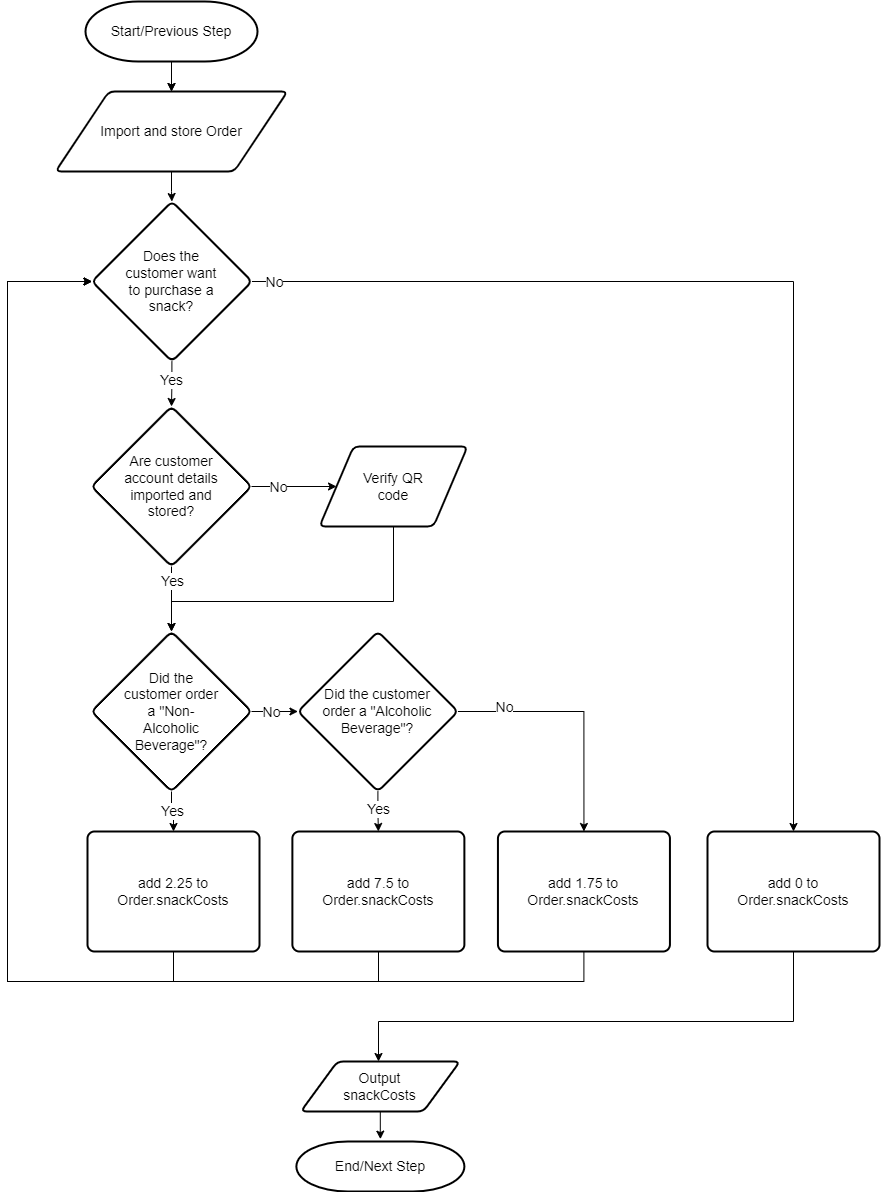
# 

## Logic 3 Flowcharts:

### Part-3 (Discounts) Flowchart:

### 

### Part-4 (Snack Charges) Flowchart:



# TEST

#### 

| **Conditions** | **Test** |
| --- | --- |
| One senior, two adults, and one child go to see the theater production of “Chewbacca Chew’s the Furniture” at 9:00 PM. They opt for the VIP valet treatment (tipping $7.50)and at intermission, indulge themselves with three wines, four bags of chips, and threeSprite soft drinks. What was the total cost? | *Start →*Customer creates an account*→*Create and store Order  *→ 4 → Part 1 return $10.25 → Part 2 $460→ Part 3 return -$92 → Part 4 return 16 →* Process $50 deposit from customer credit card*→ Part 5 return $30→* Order.valvetCosts += 37.5 *→ wait→ scan QR → yes→ Part 6 printed total of $431.75 →* Permit customer to leave*→ End* |
| Two adults go to see the musical “Yoda Sings” at 2:00 PM. They don’t use the valet,but at intermission, indulge themselves with two wines and two bags of popcorn. What was the total cost? | *Start →Customer creates an account →Create and store Order → 2 Adults → Part 1 add $8.00 → Part 2 add $140.00 → Part 3 add -$7.00 → Part 4 add $18.50 → Process $50 deposit from customer credit card→ Part 5 add $0→ wait→ scan QR → yes→ Part 6 printed total of $159.5→ Permit customer to leave→ End* |

# FINAL DESCRIPTION

In this case, we created a data structure (bunch of classes) to optimize work with data. There is a main data type called “***Order***”. It contains all data about the order that was asked for during the subprocess, including charges for tickets,covers, parking, and snacks. Moreover, “***Order***” has a ***QR*** code. The logic is based on dividing programs into small logics. Each of them works with order sets, gets, or calculates data from there. Only one moment not covered by functions is the tip charge. It is in the main program between functions. Finally, the last function gets all the data and prints the final result. Once the customer has made payment in full they are able to leave the premises.