Planning sheet

Task 1: Describe the problem

I am developing a program for people to order pizza from Heavenly Pizza, a local pizzeria who wants to operate without contact during the Covid-19 pandemic. Customers order their pizza through this program, and can either pick up the pizza from the store or have it delivered to their home.

Task 2: Identify the input information

What information will the user have to enter? Copy and complete this table.

|  |  |  |  |
| --- | --- | --- | --- |
| **Variable name** | **Scope** | **Data type** | **Purpose of variable** |
| name | Local? | string | The customer’s name for delivery and pickups |
| address | Local? | string | The customer’s delivery address (not necessary if they’re picking up) |
| order\_type | Global? | String or int? | Whether the customer is picking up their order or getting it delivered (e.g. 1 for delivery 2 for pickup) |
| pizza\_order | Local? | int | Which pizza the customer is choosing, would relate to a pizza id of sorts e.g. Hawaiian - 5 |
| stuff\_crust | local | string | Y or N, yes means stuffed crust is added to the pizza, n means it will not be added. |

Task 3: Identify other variable information

What information will the program need to store? Also include constants and derived values as they will maximise the flexibility and robustness of your plan. Your chosen scope should match the way the variable will be used in your program.

|  |  |  |  |
| --- | --- | --- | --- |
| **Variable name** | **Scope** | **Data type** | **Purpose of variable and where it comes from.** |
| PIZZA\_RESTRICTION | Local? | int | This controls the maximum amount of pizzas that the user is allowed to order, it is a constant because it may be subject to change later. |
| GOURMET\_PRICE | Local? | int | The price of the pizzas in the gourmet menu |
| REGULAR\_PRICE | Local? | int | The price of the pizzas in the regular menu |
| price | Global? | int? | Every time they order a pizza the price gets added on, delivery costs get added here too, as well as stuffed crust price. |

Task 4: Identify the indexed data (list) structure

What information will be stored in a list?

|  |  |  |  |
| --- | --- | --- | --- |
| **List name** | **Scope** | **Data type** | **Purpose** |
| regular\_menu | Global? | 2D list, each list: [string, string, int] | Each list in the list corresponds to a pizza that can be ordered. The first string is the pizza’s name, the second string is the description, and the int is its “id” that the user will order off of. |
| gourmet\_menu | Global? | 2D list, each list: [string, string, int] | Same as above except for the gourmet pizzas |
| order | Local? | ints | Appending the id of the pizzas that the users have ordered. |
| crust\_types | Local? | strings | Appending the y or n from each pizza when the user is asked whether they want stuffed crust or not. |

Task 5: Identify the output information

What information will the program need to print out?

|  |  |  |
| --- | --- | --- |
| **Output** | **When it happens** | **Details** |
| Introduction/welcome message | When the program starts | Welcomes the user, tells them (briefly) how the program works, and tells them important information like delivery costs, restrictions on how many pizzas they can order, etc. |
| The menus | Every time the user wants to order a pizza, they can see the menus. | Will print through the 2D lists in a format somewhat like ‘[pizza name] ([pizza id]) – [pizza description]’ |
| The user’s order | After they have finished ordering, or possibly during the process when they want? | Print out the pizzas that they have ordered so far (and possibly whether they have stuffed crust or not?). |
| The user’s relevant information | At the end before they confirm their order | The user’s name, their address if they are getting their pizza delivered, maybe phone number or email? |
| Farewell message | Once they have confirmed their order | Something like “See you soon, enjoy your meal”? or along the lines of that. |

Task 6: Write the algorithm

* [output the welcome message]
* Would you like to pick up or deliver?
  + Store in variable
    - If pick-up, get their name
    - If deliver, get their address too and add eight dollars to the price
* [repeat for as many pizzas as the user wants (3 or below, though)]
  + Print out the pizzas
  + Input – what pizza do you want (enter the id)
    - Append this number to the order list
  + Ask them if they want to stuff their crust.
    - If y – append y to the list and add the extra cost to the price
    - If n – append n to the list, no extra cost
* Display the user’s order
* Confirmation – is this right?
  + If not – redo the order process
  + If it is – carry on
* Display relevant customer information
* Confirmation – is this right?
  + If not – redo the customer information input process
  + If it is – carry on
* [output farewell message]

Task 7: Testing Plan

Include a testing schedule with input cases that ensures that your program will work correctly on all inputs – **expected, boundary, and exceptional input**.

Step through your algorithm showing the outputs that are generated. If you find problems with your algorithm make changes to it and run through your algorithm again until it works.

|  |  |
| --- | --- |
| **TEST** | **RESULT** |
| **Delivery or Pickup - expected** | |
| 1 (or whatever relates to delivery) | Program stores this and asks for relevant information, as well as adding extra cost to the price variable |
| 2 (or whatever relates to pick up) | Program stores this and asks for relevant information |
| **Delivery or Pickup - unexpected** | |
| “hello”, “asflkdsajflkagj;s”, “deliver” (any string that’s not expected) | Error message, will ask the user to enter again |
| 7, 3.4, unexpected numbers | Error message, will ask the user to enter again |
| [enter] – no input | Error message, will ask the user to enter again |
| **Pizza order - expected** | |
| 3 | Will order one Garlic pizza, and add the regular price to the price variable as well as appending the pizza id to the order list |
| 10 | Will order one Big BBQ Bacon pizza, and add the gourmet price to the price variable as well as appending the pizza id to the order list |
| **Pizza order – boundary expected** | |
| 1 | Will order one Margherita pizza, and add the regular price to the price variable as well as appending the pizza id to the order list |
| 12 | Will order one Meatlovers pizza, and add the gourmet price to the price variable as well as appending the pizza id to the order list |
| **Pizza order – boundary unexpected** | |
| 0 | Will ask the user to enter a pizza id that exists |
| 13 | Will ask the user to enter a pizza id that exists |
| **Pizza order – exceptional unexpected** | |
| -24, 500, any int that isn’t covered | Will ask the user to enter a pizza id that exists |
| 3.2, 7.1, any float | Will come up with an error message and ask the user to enter the whole number that correspond to the pizza they want |
| “hello”, “cheese”, any string | Will come up with an error message and ask the user to enter the whole number that correspond to the pizza they want |
| [enter], no input | Will come up with an error message and ask the user to enter the whole number that correspond to the pizza they want |
| **Y/N input – stuffed crust, order confirmation, etc. - expected** | |
| Y | Will do the “yes” result – e.g. for stuffed crust that would be adding on the extra price and recording that they want stuffed crust with that pizza. |
| N | Will do the “no” result – e.g. for stuffed order confirmation they’d repeat their order |
| **Y/N input – stuffed crust, order confirmation, etc. - unexpected** | |
| “hello”, “asflkdsajflkagj;s”, “deliver” (any string that’s not expected) | Error message, will ask the user to enter again |
| 7, 3.4, unexpected numbers | Error message, will ask the user to enter again |
| [enter] – no input | Error message, will ask the user to enter again |