Final Project Documentation

Project: Food Delivery System

Course: Basics of programming 2

Responsible instructor: Dr. Dunaev Dmitriy

Laboratory instructor: Mr. Mohammad Saleem

Student name: Ahmed Borchani

Neptun code: EFKTV2

1)Problem statement and the task description:

My task consists on making a restaurant system that supports ordering food, managing food stock and delivery.

2)Solution design:

I tried to implement as many C++ and object-oriented programming techniques in order to create this system.

I used:

-an object-oriented approach to programming,

-dynamic memory management,

-exception handling,

-file management.

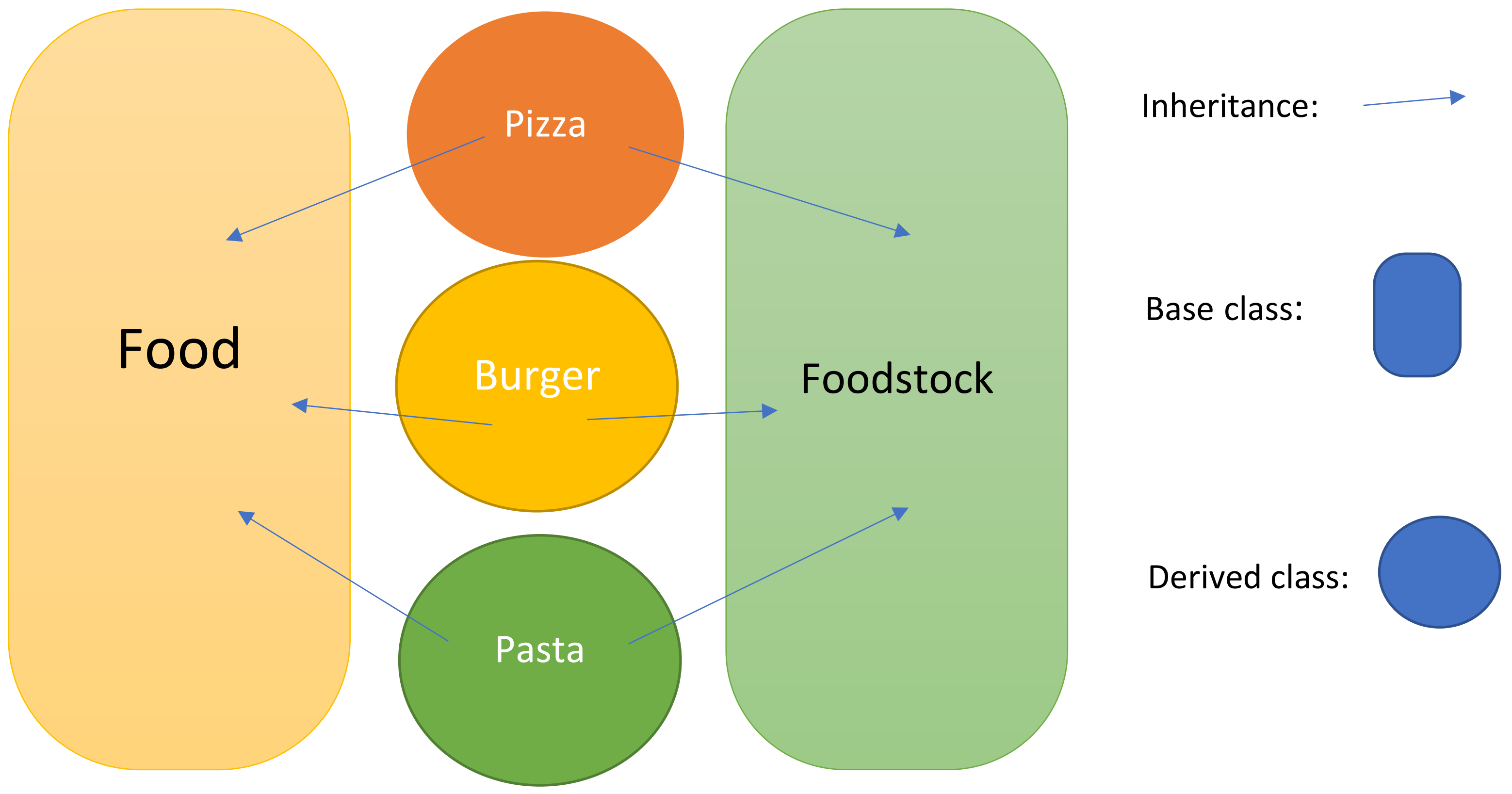
-Inheritance (base and derived classes).

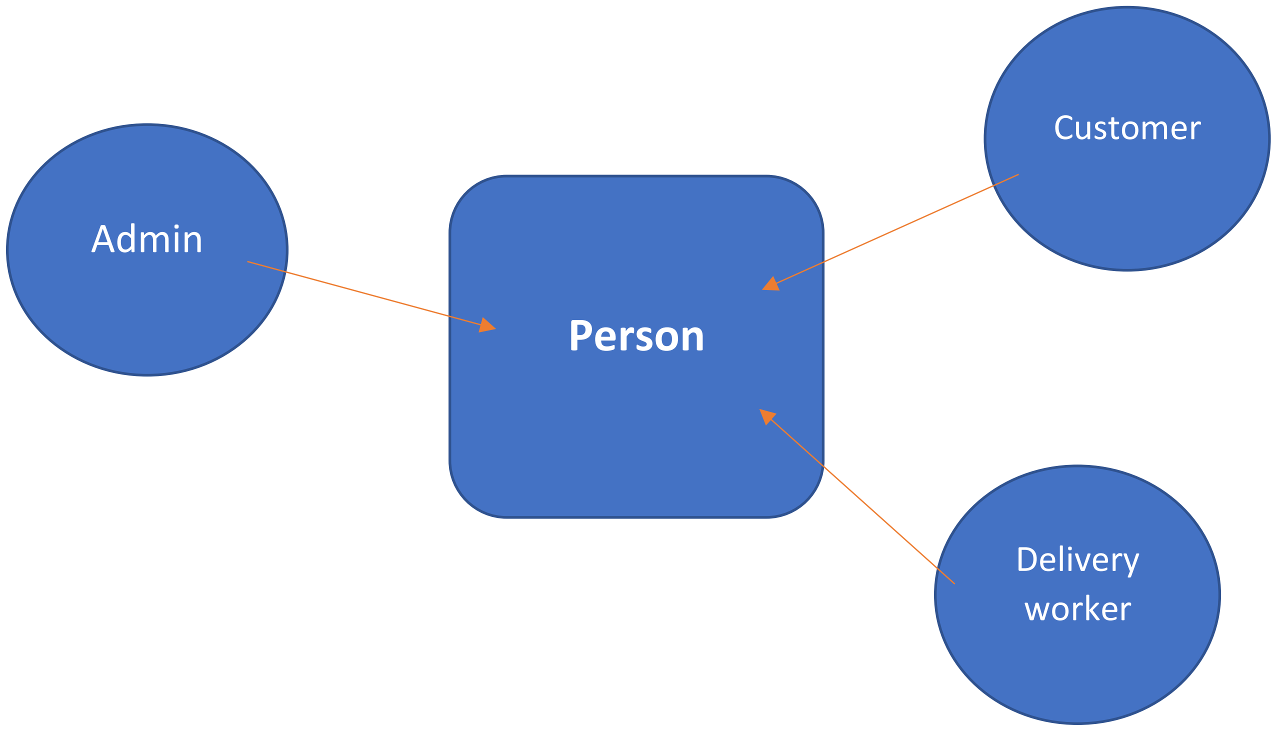
-Polymorphism: abstract classes and virtual functions.

-Operator overloading with local and global operations.

-Multiple inheritance.

3)Solution:


Food
Pizza
Burger
Pasta
Foodstock
Inheritance:
Base class:
Derived class:


Admin
Person
Customer
Delivery worker

Main classes:

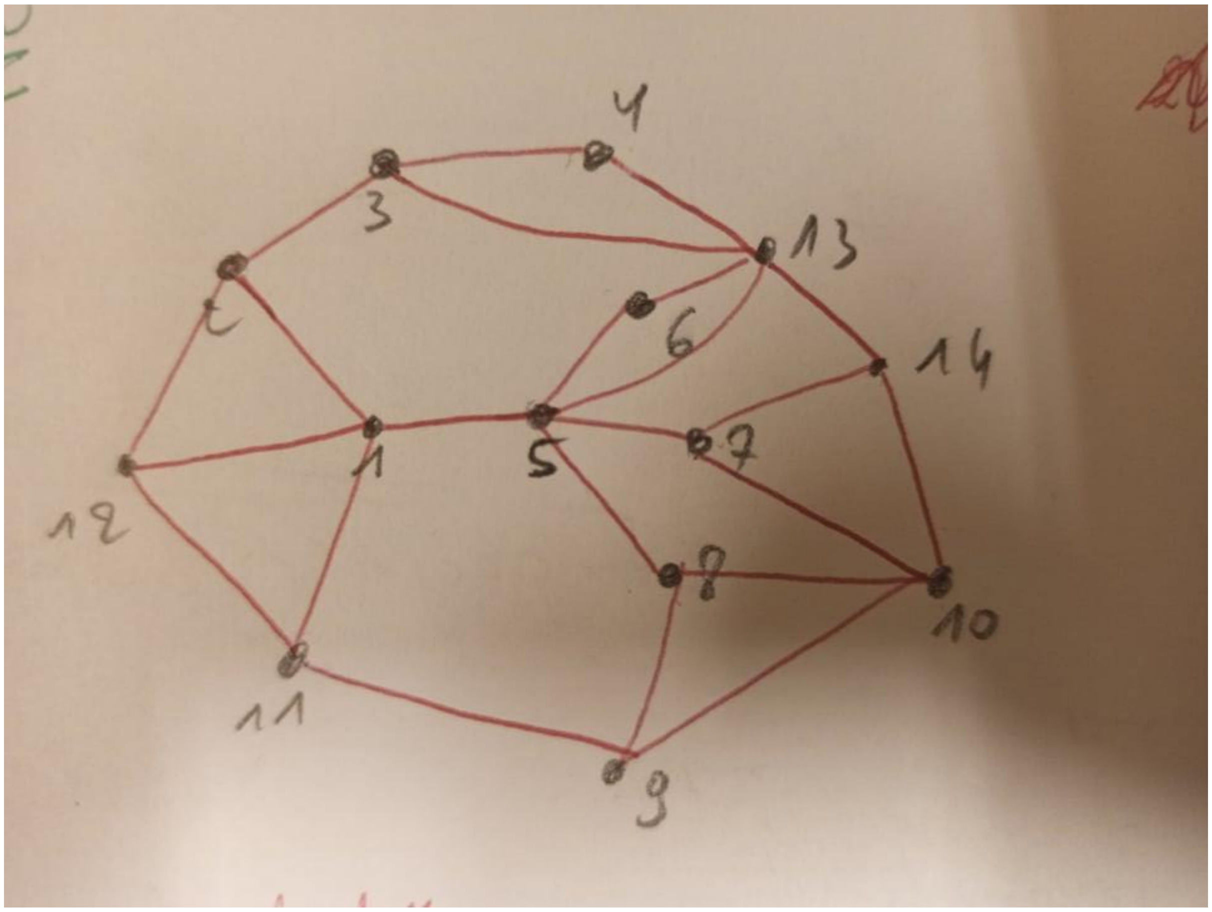
1. Person: its subclasses are Administrator, customer and delivery worker. For the login of these 3 I used file handling as their names and passwords are stored. The login process uses exception handling in case a wrong option was entered.

   a. Administrator: can fill the food stock, check it and add a delivery worker.
      He can only sign in (to add an admin it has been done in the file directly, for security reasons)

   b. Customer: can order multiple types of food
      If the vector of delivery guys is empty the delivery will be stored in a file for later use, when delivery is possible. If it's not empty, the customer will be assigned to the delivery guy available.

   c. Delivery worker: when he signs in, he gets a list of customers that he has to go to their district (an easier approach to the real address), as long as he has a place for more packets in his vehicle. He can only sign in (to add an admin it has been done in the file directly, or by the admin for security reasons)
      This class has a map of the first 14th districts of Budapest in form of graph (matrix form). I used Dijkstra algorithm to find the shortest root to each district from the source (district 1: the place of the restaurant). The graph is unweighted and

undirected (1 means that the two districts are connected. 0 means they're not). The graph and matrix implementation are my own work, however I got the Dijkstra algorithm from geekforgeeks website. This is the graph of the districts:



2. food, foodstock: pizza, burger and pasta inherit from both of these classes. They inherit from the Foodstock to be able to change the values of elements when an

object is created. They inherit from Food because they have many attributes in common.

- food.h is the header file for both food, foodstock
- foodstock.cpp is the source file for both food, foodstock

3.exception class for exception handling

4.Main class has the main login functions and exception handling functions.

4)External solutions:

Dijkstra algorithm from geeksforgeeks

5)Testing:

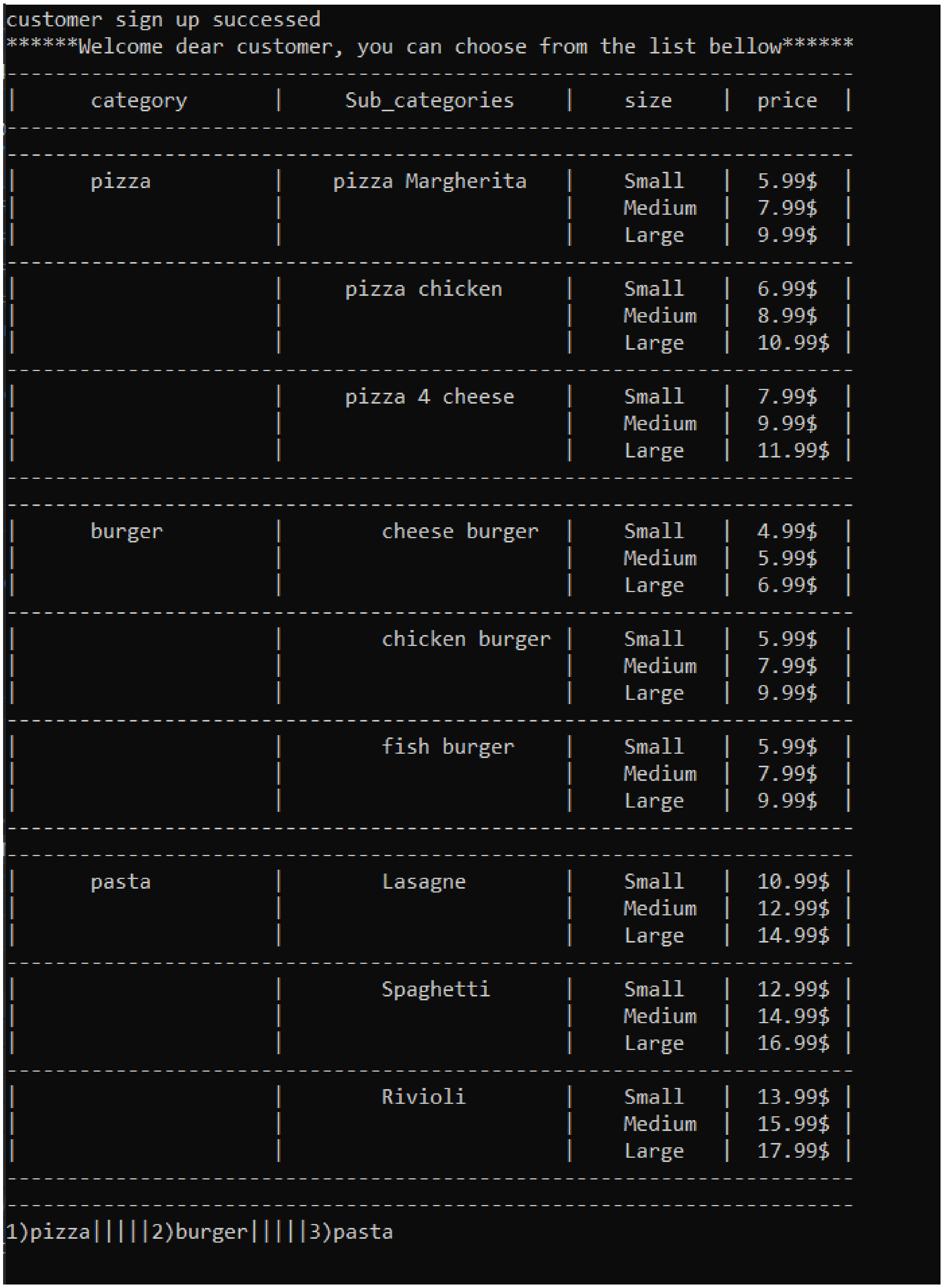
For testing:

Admin login: please use for name “test” and the password “1234”

Delivery guy login: please use for name “dell” and the password “5678”

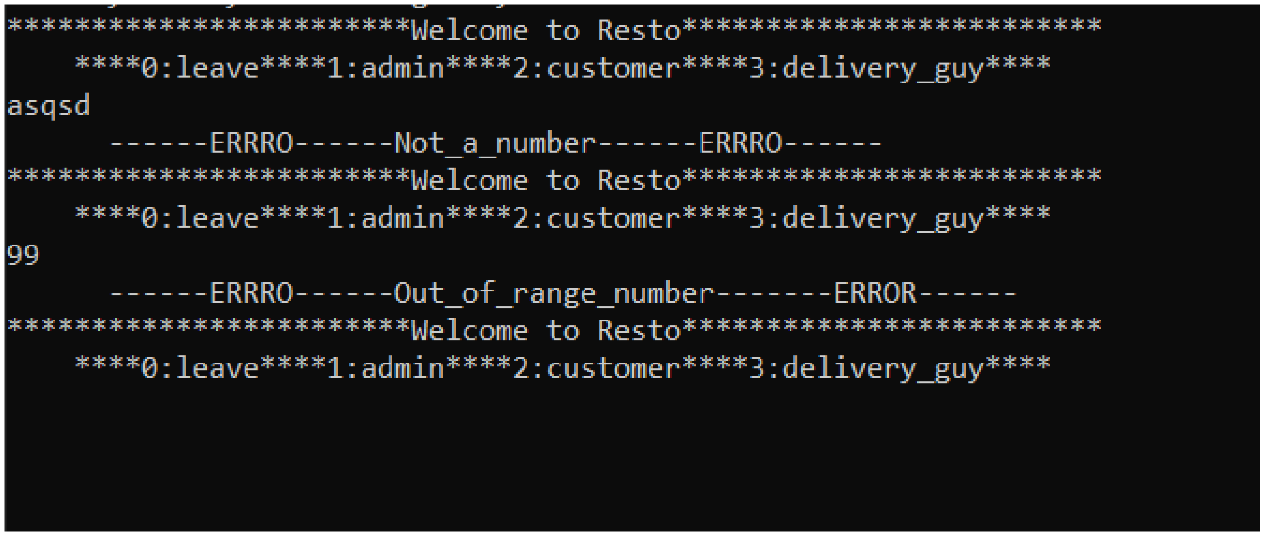
The program can be easily tested through using the functions of admin, customer and delivery worker.

This is the menu that the customer can choose from:



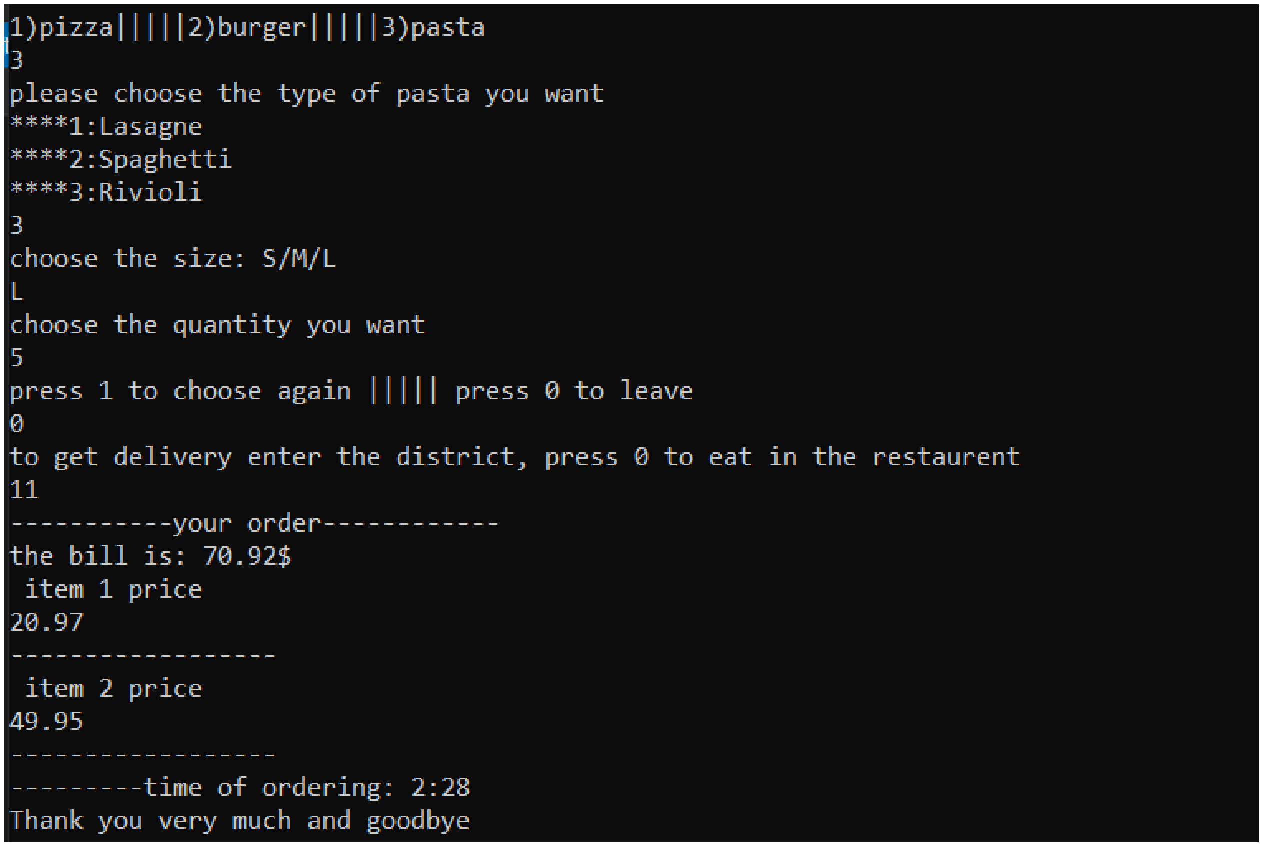
```
customer sign up successed
******Welcome dear customer, you can choose from the list bellow******
----------------------------------------------------------------------
|      category       |      Sub_categories     |    size    |  price  |
----------------------------------------------------------------------
----------------------------------------------------------------------
|      pizza          |    pizza Margherita     |    Small   |  5.99$  |
|                     |                         |    Medium  |  7.99$  |
|                     |                         |    Large   |  9.99$  |
----------------------------------------------------------------------
|                     |     pizza chicken       |    Small   |  6.99$  |
|                     |                         |    Medium  |  8.99$  |
|                     |                         |    Large   |  10.99$ |
----------------------------------------------------------------------
|                     |     pizza 4 cheese      |    Small   |  7.99$  |
|                     |                         |    Medium  |  9.99$  |
|                     |                         |    Large   |  11.99$ |
----------------------------------------------------------------------
----------------------------------------------------------------------
|      burger         |        cheese burger    |    Small   |  4.99$  |
|                     |                         |    Medium  |  5.99$  |
|                     |                         |    Large   |  6.99$  |
----------------------------------------------------------------------
|                     |        chicken burger   |    Small   |  5.99$  |
|                     |                         |    Medium  |  7.99$  |
|                     |                         |    Large   |  9.99$  |
----------------------------------------------------------------------
|                     |        fish burger      |    Small   |  5.99$  |
|                     |                         |    Medium  |  7.99$  |
|                     |                         |    Large   |  9.99$  |
----------------------------------------------------------------------
----------------------------------------------------------------------
|      pasta          |        Lasagne          |    Small   |  10.99$ |
|                     |                         |    Medium  |  12.99$ |
|                     |                         |    Large   |  14.99$ |
----------------------------------------------------------------------
|                     |        Spaghetti        |    Small   |  12.99$ |
|                     |                         |    Medium  |  14.99$ |
|                     |                         |    Large   |  16.99$ |
----------------------------------------------------------------------
|                     |        Rivioli          |    Small   |  13.99$ |
|                     |                         |    Medium  |  15.99$ |
|                     |                         |    Large   |  17.99$ |
----------------------------------------------------------------------
----------------------------------------------------------------------
1)pizza|||||2)burger|||||3)pasta
```

This is how exception handling throws an exception according to the input:



```
************************Welcome to Resto*************************
    ****0:leave****1:admin****2:customer****3:delivery_guy****
asqsd
      ------ERRRO------Not_a_number------ERRRO------
************************Welcome to Resto*************************
    ****0:leave****1:admin****2:customer****3:delivery_guy****
99
      ------ERRRO------Out_of_range_number-------ERROR------
************************Welcome to Resto*************************
    ****0:leave****1:admin****2:customer****3:delivery_guy****
```

This is the bill after the customer choose his order:



```
1)pizza|||||2)burger|||||3)pasta
3
please choose the type of pasta you want
****1:Lasagne
****2:Spaghetti
****3:Rivioli
3
choose the size: S/M/L
L
choose the quantity you want
5
press 1 to choose again ||||| press 0 to leave
0
to get delivery enter the district, press 0 to eat in the restaurent
11
-----------your order------------
the bill is: 70.92$
 item 1 price
20.97
------------------
 item 2 price
49.95
------------------
---------time of ordering: 2:28
Thank you very much and goodbye
```

Acknowledgment

I would like to thank Dr. Dunaev Dmitriy and Mr. Mohammad Saleem for their efforts and great-quality teaching and assistance in both the real and online classes.