### IMG_256

**Computer Project**



**Topic: Pizza Delivery Service**

### 

Emily Eldose

Class XII A

(2017-2018)



## COMPUTER SCIENCE PROJECT

**GRADE XII – CBSE**

#### REGISTER NO: ……………………

This is certified to be the investigatory project undertaken by Miss / Master……….…………………………………..…………..……..   
in Computer Science (Python) during the year 2017-2018.

The topic chosen is………..………….…………………………………...

This project has been brought to completion under the guidance of Mrs. Neenu Bobby.

Principal Teacher- in- charge

Submitted for the Practical Examination on \_\_/ \_\_ / 2018 held at GEMS Our Own Indian School, Dubai.

Examiners

1) …………………………………..…….

2)……………………………..………….

#### Dubai

2017-18

### **ACKNOWLEDGEMENT**

It is with immense respect and gratitude, I express my thanks to all those who have inspired, motivated and guided me during the completion of the project work.

First and foremost, I am grateful to the Almighty for helping me to complete this project.

I express my sincere gratitude to the Principal, Mrs. Lalitha Suresh, Headmistress, Mrs. Usha Balachandran and the Supervisior,Mrs. Bharati Bakshi, for having provided me an excellent institution with well qualified staff.

I extend my sincere thanks to my computer teacher,   
Mrs. Neenu Bobby for her invaluable contributions towards improving this project.

Further, I express my gratitude to my beloved family and friends, without whom I would not have been able to complete this project.

Although I have taken utmost care in order to remove any discrepancy that must have crept in and have ensured that it works to the best of its capabilities, suggestions for improvement are always welcome from your end.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Topic** | **Page No.** |
| **1.** | **Certificate** | **1** |
| **2.** | **Acknowledgement** | **2** |
| **3.** | **Introduction** | **4** |
| **4.** | **Aim** | **10** |
| **5.** | **Project Description** | **11** |
| **6.** | **Technical Description** | **12** |
| **7.** | **User Documentation** | **15** |
| **8.** | **Flow Chart** | **18** |
| **9.** | **Source Code** | **19** |
| **10.** | **Sample Outputs** | **47** |
| **11.** | **Possible improvements** | **50** |
| **12.** | **Bibliography** | **51** |

### **INTRODUCTION**

* **Introduction to Python :**
* **Python** is a widely used general purpose, high level programming language. Its design philosophy emphasises code readability, and its syntax allows programmers to express concepts in fewer lines of code than would be possible in languages such as C++, Java, etc.
* The language provides constructs intended to enable clear programs on both a small and large scale. Python supports multiple programming paradigms, including object-oriented, imperative and functional programming or procedural styles. It features a dynamic type system and automatic memory management and has a large and comprehensive standard library.



* Python interpreters are available for installation on many operating systems, allowing python code execution on a wide variety of systems. Python code can be packaged into stand-alone executable programs for some of the most popular operating systems, allowing the distribution of python-based

software for use on those environments without requiring the installation of a python interpreter.

* Python was conceived in the late 1980s and its implementation was started in December 1989 by Guido Van Rossum at CWI in Netherlands as a successor to the ABC language capable of execution handling and interfacing with the Amoeba OS. Python 2.0 was released on October 2000 and had many major new features, including a cycle-detecting language garbage collector and support for Unicode.
* With this release the development process was changed and became more transparent and community backed. Python 3.0 a major, backwards-incompatible release was released on 3 December 2008 after a long period of testing. Many of its major features have been ported to the backwards-compatible 2.6 and 2.7
* Type constraints are not checked at compile time; rather at operations on an object may fail signifying that the given object is not of suitable type. Despite being dynamically typed, Python is strongly typed, forbidding operations that are not well-defined (for example adding a number to a string ) rather than silently attempting to make sense of them. Python allows programmers to define their own types using classes , which are most often used for object-oriented programming.
* **Popular features of Python :**

1. **Simple :** Python is a simple and minimalistic language. Reading a good Python program feels almost like reading English, although very strict English! This pseudo code nature of Python is one of its greatest strengths. It allows you to concentrate on the solution to the problem rather than the language itself.
2. **Easy to learn :** As you will see, Python is extremely easy to   
   get started with. Python has an extraordinarily simple syntax.
3. **Read and Open structure :** In simple terms,you can freely distribute copies of this software, read its source code, make changes to it, use pieces of it in new free programs, and that

you know you can do these things. This is one of the reasons why Python is so goo-it has been created and is constantly improved by a community who just wants to see a better Python.

1. **High level language :** When you write programs in Python you never need to bother about the low level details such as managing memory used by your program etc.
2. **Portable :** Due to its open source nature, Python has been ported ( i.e. changed to make it work on ) to different platforms. All your Python programs can work on any of these platforms without requiring any changes at all if you are careful enough to avoid any system-dependent features. You can use Python on Linux, Windows, Free BSD, Macintosh, Solaris, Amiga, AROS, Palm OS, QNX, VMS, Psion, Playstation, PocketPC etc.
3. **Embeddable :** You can embed Python within your C/C++ programs to give ‘scripting’ capabilities for your program’s users.
4. **Interpreted :** A program written in a compiled language like   
   C or C++ is converted from the source language into a language that is spoken by the computer using a compiler with

various options and flags. When you run the program, the linker/loader software copies the program from hard disk to memory and starts running it. Python on the other hand, does not need compilation to binary. You just run the program directly form the source code. Internally, Python converts the source code into an intermediate form called byte   
codes and then translates into native language of the computer   
and then runs it. All this, actually make using Python   
much easier since you don’t have to worry about compiling the program, making sure that the proper libraries are linked and loaded etc. This also makes Python programs more portable, since you can just copy your Python program   
onto another computer and it still works.

1. **Object-Oriented :** Python supports procedure-oriented programming as well as object-oriented programming. In procedure-oriented languages the program is built around procedures or functions which are nothing but reusable pieces

of programs. In object-oriented languages the program is built around the objects which combine data and functionality. Python has a very powerful but simplistic way of doing OOP especially when compared to big languages like Java or C++.

1. **Extensible :** If you need a critical piece of code to run very fast want to have some piece of algorithm not to be open, you can code that part of your program in C or C++ and then use them form your Python program.
2. **Extensive Libraries :** The python Standard library is huge. It helps us do various things involving regular expressions, documentation generation, unit testing, threading, databases, web browsers, CGI, ftp, email, XML, XML-RPC, HTML,   
   WAV files, Cryptography, GUI (graphical user interface), and other system dependent stuff. Remember, all this is available when Python is installed. This is called ‘Batteries Included’ Philosophy of Python.

* **Mode of Python Shell Used :**

**Interactive mode:** Allows us to interact with OS

**Script mode:**  Create and edit a python source file.

**AIM**

* The objective of this program is to develop a Python project   
  on Pizza Delivery Service i.e ordering pizzas from the   
  nearest restaurant or pizzeria so as to be delivered to the customer. It is based on Online Food Ordering Services.
* The project primarily focuses on:
* Ordering pizza / (s) of choice
* Accepting the customer’s details
* Modify/Cancel the order
* Calculate the bill
* Display Receipt

**PROJECT DESCIRPTION**

* The program is made using Python 2.7
* Firstly, the Customer has to select the pizza of his/her choice -

1. From a variety of selected pizzas OR
2. Custom-made i.e choose a pizza with toppings of their selection.

* Secondly, the Customer is required to create an account-permanent or temporary( as Guest ). The details of the Customer are stored for delivery purpose, send updates regarding deals and discounts, announce offers etc.
* The Customer is given a choice of

1. Cancel the complete order
2. Create changes or modify the original order
3. Proceed to billing without any change

* Finally, the Receipt is displayed with all the details of the order.

**TECHNICAL DESCIRPTION**

* **System Requirements :**

|  |
| --- |
| * Windows XP and above |
| * Mac OS X 10.6 and higher |
| * 16-bit ARM Processor |
| * 16 MB or higher memory |
| * 640x480 resolution with 8-bit colour |
| * Python VERSION: Python 2.7.6 |



* **Main Functions Used :**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Name of Function** | **Purpose** |
| 1. | **order\_pizza()** | **Choose an assorted,special or customised pizza** |
| 2. | **allpizza\_main()** | **Choose the pizza from ALL-PIZZAS (an assortment)** |
| 3. | **custompizza\_main()** | **To choose size, crust and toppings of customised pizza** |
| 4. | **size() and crust()** | **To select the size and crust** |
| 5. | **meat() and veg()** | **Select the non-veg/veg toppings of customised pizza** |
| 6. | **toppings()** | **Select the toppings of a custom-made pizza** |
| 7. | **todayspl\_main()** | **Choose the pizza from TODAY 's SPECIAL** |
| 8.. | **account\_main()** | **Choose whether to create an account ( temporary or permanent) or login** |
| 9. | **inputdata()** | **Accept details from Customer** |
| 10.. | **confirmdata()** | **Confirm the details given** |
| 11. | **createaccount()** | **Internally creates an object that stores details** |
| 12. | **login()** | **Log in to account to proceed** |
| 13. | **guest()** | **To sign in with temporary account, ask details** |
| 14. | **cancel\_order()** | **Cancel the complete order** |
| 15. | **get\_info()** | **Obtain personal info including the address of customer** |
| 16. | **change\_order()** | **Give options of cancel/modify/proceed** |
| 17.. | **billing\_main()** | **Calculates the total bill** |
| 18.. | **display\_info()** | **Displays the final Receipt** |

**USER DOCUMENTATION**

1. **SELECT SIZE AND CRUST :**

* Select a choice from the list and enter the corresponding number to it. Ex: To order a pizza-SMALL, Enter ‘1’ in screen.
* This applies to selection of a crust too.To order a pizza-PAN, Enter ‘1’ in screen.

1. **SELECT PIZZA FROM ALL PIZZAS :**

* Enter ‘1’ in main menu
* Enter desired size and crust of pizza
* Select a choice from the list of pizzas’ names and enter the corresponding number to it. Ex: 1.BBQ, 2.Margaritas
* Enter 1/(2) in screen to select BBQ/ (Margaritas)

1. **SELECT PIZZA FROM CUSTOMIZE :**

* Enter ‘2’ in main menu.
* Enter desired size and crust of pizza.
* Meat and Veggie toppings are provided. Select a choice from the list and enter the corresponding number to it. A minimum of 3 toppings is provided for each (Meat & Veggie).
* If you do no wish to add more toppings, simply press ‘ Enter ‘ or any key other than the choice numbers.

1. **SELECT PIZZA FROM TODAY’S SPECIAL :**

* Enter ‘3’ in main menu
* Enter desired size and crust of pizza
* Select a choice from the list of pizzas’ names and enter the corresponding number to it.

1. **TO CREATE AN ACCOUNT :**

* Enter ‘1’ in Account Details MENU
* Enter name, user name, password, contact number, address etc. and other details.
* Confirm the details.

1. **LOGIN TO AN ACCOUNT :**

* Enter ‘2’ in Account Details MENU
* Enter user name and password

1. **TO SIGN UP AS GUEST :**

* Enter ‘3’ in Account Details MENU
* Enter name, contact number, city, address and area of residence.

1. **CANCEL OR MODIFY THE ORDER :**

* Enter ‘1’ to cancel the complete order and exit.
* Enter ‘2’ to make changes to your order.
* Press ‘Enter’ to ensure that there are no changes to your order.

**FLOWCHART**

**ACCOUNT DETAILS**

**ALL PIZZA**

**OVERALL MENU**

**CUSTOMIZE**

**GUEST**

**CREATE ACCOUNT**

**TODAY’S SPECIAL**

**LOGIN**

**CHANGE ORDER**

**BILLING**

**MODIFY**

**CANCEL**

**NO CHANGE**

**DISPLAY RECEIPT**

**EXIT**

**SOURCE CODE**

import os

import pickle

print '-'\*50

print "PROJECT: PIZZERIA "

print "Mangia bene"

print '-'\*50,'\n\n'

print "We are the finest of all."

class allpizza :

def \_\_init\_\_(self):

self.apizzachoice=''

self.asize=''

self.acrust=''

def allpizza\_main(self):

print '-'\*30

print 'All Pizzas-Your favourites coming right up'

print '-'\*30

print "First, Select the size and crust of your choice (ALL PIZZAS)", '\n'

self.asize=self.size()

self.acrust=self.crust()

print '-'\*30

print 'PIZZA CHOICE:'

print '-'\*30

print "1.Italy delicio",'\n',"2.Hot Barbeque",'\n',"3.Mangia Special",'\n',"4.Seafood",'\n',"5.Rasta-pout",'\n',"6.Italiano Classic" ,'\n\n'

while True:

ch=raw\_input("Enter choice(ALL PIZZAS): (1/2/3/4/5/6)")

print

if ch in ['1','2','3','4','5','6']:

break

else:

print "Invalid! Try again"

print

if ch=='1':

print "You chose PIZZA:Italy delicio"

print '-'\*30,'\n'

pi1=1

elif ch=='2':

print "You chose PIZZA:Hot Barbeque"

print '-'\*30,'\n'

pi1=2

elif ch=='3':

print "You chose PIZZA:Mangia Special"

print '-'\*30,'\n'

pi1=3

elif ch=='4':

print "You chose PIZZA:Seafood"

print '-'\*30,'\n'

pi1=4

elif ch=='5':

print "You chose PIZZA:Rasta-pout"

print '-'\*30,'\n'

pi1=5

elif ch=='6':

print "You chose PIZZA:Italiano Classic"

print '-'\*30,'\n'

pi1=6

d={1:"Italy delicio",2:"Hot Barbeque",3:"Mangia Special",4:"Seafood",5:"Rasta-pout",6:"Italian Classic"}

for i in d:

if i==pi1:

self.apizzachoice=d[i]

class custompizza:

def \_\_init\_\_(self):

self.pcrust=''

self.psize=''

self.ptop1=''

self.ptop2=''

self.ptop3=''

self.ptop4=''

self.ptop5=''

self.ptop6=''

def custompizza\_main(self)

print '-'\*30

print 'CUSTOM PIZZA'

print '-'\*30

print "You must select your choices for the below fields:"

print "1.Size"

print "2.Crust"

print "3.Toppings"

self.psize=self.size()

self.pcrust=self.crust()

self.toppings()

def size(self):

print '-'\*30

print "SIZE:"

print '-'\*30,'\n'

print "1.Small",'\n',"2.Medium",'\n',"3.Large",'\n'

while True:

ch=raw\_input("Enter choice(SIZE): (1/2/3)")

print

if ch in ['1','2','3']:

break

else:

print "Invalid! Try again",'\n'

if ch=='1':

print "You chose SIZE: SMALL"

print '-'\*30,'\n'

return 'Small'

elif ch=='2':

print "You chose SIZE: MEDIUM"

print '-'\*30,'\n'

return 'Medium'

elif ch=='3':

print "You chose SIZE: LARGE"

print '-'\*30,'\n'

return 'Large'

def crust(self):

print '-'\*30

print "Crust"

print '-'\*30,'\n'

print "1.Pan", '\n', "2.Classic" ,'\n' ,"3.Stuffed Crust",'\n'

while True:

ch=raw\_input("Enter choice(CRUST): (1/2/3)")

print

if ch in ['1','2','3']:

break

else:

print "Invalid! Try again",'\n'

if ch=='1':

print "You chose CRUST: PAN"

print '-'\*30,'\n'

return 'Pan'

elif ch=='2':

print "You chose CRUST: CLASSIC"

print '-'\*30,'\n'

return 'Classic'

elif ch=='3':

print "You chose CRUST: STUFFED CRUST"

print '-'\*30,'\n'

return 'Stuffed Crust'

def meat(self):

while True:

print

ch=raw\_input("Enter one choice: (1/2/3/4/5)")

print

if ch in ['1','2','3','4','5','']:

break

else:

print "Invalid! Try again"

if ch=='1':

print "You chose MEAT TOP': PEPPERONI"

return '1'

elif ch=='2':

print "You chose MEAT TOP': CHICKEN FAJITA"

return '2'

elif ch=='3':

print "You chose MEAT TOP': GRILLED CHICKEN"

return '3'

elif ch=='4':

print "You chose MEAT TOP': SMOKED TURKEY"

return '4'

elif ch=='5':

print "You chose MEAT TOP': ROASTED BEEF"

return '5'

elif ch=='':

print "You chose NONE"

def veggie(self):

while True:

print

ch=raw\_input("Enter choice: (6/7/8/9/10)")

print

if ch in ['6','7','8','9','10','']:

break

else:

print "Invalid! Try again"

print

if ch=='6':

print "You chose VEGGIE TOP': MOZZARELLA"

return '6'

elif ch=='7':

print "You chose VEGGIE TOP': FRESH TOMATOES"

return '7'

elif ch=='8':

print "You chose VEGGIE TOP': SLICED BLACK OLIVES"

return '8'

elif ch=='9':

print "You chose VEGGIE TOP': FRESHLY CUT ONIONS"

return '9'

elif ch=='10':

print "You chose VEGGIE TOP': GARLIC"

return '10'

elif ch=='':

print "You chose NONE",'\n'

def toppings(self):

print '-'\*30

print "Toppings"

print '-'\*30,'\n’

print "1.MEAT","2.VEGGIE",'\n'

print '-'\*30

print "MEAT TOPPINGS"

print '-'\*30,'\n'

print "You must select any 3 toppings. If you select the same topping twice,it will be added as extra"

print "For example:If you select Pepperoni twice, EXTRA PEPPERONI on your pizza." ,'\n'

print "IF YOU DO NOT WANT A TOPPING, PRESS ENTER",'\n'

print "1.Pepperoni",'\n',"2.Chicken Fajita",'\n',"3.Grilled Chicken",'\n',"4.Smoked turkey",'\n',"5.Roasted Beef",'\n'

ch1=self.meat()

ch2=self.meat()

ch3=self.meat()

print '\n\n'

print '-'\*30

print "VEGGIE TOPPINGS"

print '-'\*30,'\n'

print "You can select any 3 toppings",'\n'

print "6.Mozzarella",'\n',"7.Fresh tomatoes",'\n',"8.Sliced Black Olives",'\n',"9.Freshly Cut onions",'\n', "10.Garlic",'\n'

ch4=self.veggie()

ch5=self.veggie()

ch6=self.veggie()

print '\n'

l1=[['1',"Pepperoni"],['2',"Chicken Fajita"],['3',"Grilled Chicken"],['4',"Smoked turkey"],['5',"Roasted Beef"],['6',"Mozzarella"],['7',"Fresh Tomatoes"],['8',"Sliced black olives"],['9',"Freshly Cut onions"], ['10',"Garlic"]]

for i in l1:

if i[0]==ch1:

self.ptop1=i[1]

elif i[0]==ch2:

self.ptop2=i[1]

elif i[0]==ch3:

self.ptop3=i[1]

elif i[0]==ch4:

self.ptop4=i[1]

elif i[0]==ch5:

self.ptop5=i[1]

elif i[0]==ch6:

self.ptop6=i[1]

class todayspl:

def \_\_init\_\_(self):

self.tpizzachoice=''

self.tsize=''

self.tcrust=''

def todayspl\_main(self):

print "First, Select the size and crust of your choice (TODAY'S SPECIAL)",'\n'

self.tsize=self.size()

self.tcrust=self.crust()

print '-'\*30

print "Today's special-YOU HAVE COME TO THE RIGHT PLACE!!"

print '-'\*30

print "1.Mi Manchi",'\n',"[ Enjoy this tangy, tasty pizza draped in BBQ sauce with roasted turkey ham, green peppers and shrimps ]",'\n'

print "2.Io amo l'Italia",'\n'," An ultimate mix of chicken pepperoni and juicy grilled chicken strips together with mushrooms and melting mozarella cheese",'\n'

while True:

ch=raw\_input("Enter choice(TODAY'S SPECIAL PIZZAS): (1/2)")

print

if ch in ['1','2']:

break

else:

print "Invalid! Try again",'\n'

if ch=='1':

print "You chose PIZZA: Mi Manchi"

print '-'\*30,'\n'

self.tpizzachoice='Mi Manchi'

elif ch=='2':

print "You chose PIZZA: Io amo l'Italia"

print '-'\*30,'\n'

self.tpizzachoice= "Io amo l'Italia"

class account:

def \_\_init\_\_(self):

self.username=''

self.password=''

self.name=''

self.city=''

self.area=''

self.address=''

self.phno=0

self.emailid=''

def account\_main(self):

print '-' \* 30

print "ACCOUNT DETAILS"

print '-' \* 30

print "1.Create Account"

print "2.Login"

print "3.Guest"

while True:

print

ch=raw\_input("Enter choice(ACCOUNT): (1/2/3)")

print

if ch in ['1','2','3']:

break

else:

print "Invalid! Try again"

if ch=='1':

self.createaccount()

elif ch=='2':

self.login()

elif ch=='3':

self.guest()

def inputdata(self):

print '-'\*30

print "1.USER-INPUT DATA"

print '-'\*30,'\n'

self.username=raw\_input("Enter your username: ")

print

while True:

self.password=raw\_input("Enter your password: ")

self.confirmp=raw\_input("Confirm password: ")

if self.confirmp!=self.password:

print "Password entered incorrect. Try again",'\n'

else:

break

print '\n', "Please provide your details below. ' \*\* ' ones are compulsory"

self.name=raw\_input("\*\* Enter your name: ")

while True:

self.phno=raw\_input("\*\* Enter your mobile no.(10 digits must): ")

if self.phno.isdigit():

break

else:

print "Numbers only accepted",'\n'

continue

print

self.city=raw\_input("\*\* Enter your city name: ")

self.area=raw\_input("\*\* Enter the area of residence: ")

self.address=raw\_input("\*\* Enter your address along with references to Bldg no, flat no etc.: ")

print '\n',"To recieve latest deals,offers and discount vouchers at 'MANGIA BENE,VIVI FELICE,' Please provide your email-id below",'\n'

self.emailid=raw\_input("Enter your emailid: ")

print '\n\n'

def confirmdata(self):

print '-'\*30

print "2.CONFIRM ACCOUNT DETAILS"

print '-'\*30,'\n'

print "NAME:",self.name

print "USERNAME:",self.username

print "PHONE NO.:",self.phno

print "CITY:",self.city

print "ADDRESS:",self.address

print "AREA:",self.area

print "EMAILID:",self.emailid,'\n'

print "Confirm account details?"

while True:

ch=raw\_input("Enter choice (CONFIRMATION): (Y/N) :")

if ch in ['y','Y','n','N']:

break

else:

print "Invalid option. Try again",'\n'

continue

if ch in ['y','Y']:

print "Account Created Successfully!",'\n\n',

self.login()

elif ch in ['n','N']:

print

self.createaccount()

def createaccount(self):

print "-"\*30

print "CREATE ACCOUNT"

print "-"\*30,'\n'

s1=account()

s1.inputdata()

file1=s1.username+'.dat'

myfile=open(file1,"wb+")

pickle.dump(s1,myfile)

myfile.close()

print "Account Created Successfully!",'\n\n',

s1.confirmdata()

def login(self):

print "-"\*30

print "LOGIN TO ACCOUNT"

print "-"\*30,'\n'

import os.path

username=raw\_input("Enter username")

password=raw\_input("enter password")

if os.path.isfile(username+'.dat'):

fileout=open(username+'.dat',"rb+")

try:

s1=pickle.load(fileout)

if (s1.username==username) and (s1.password==password):

print "SUCCESSFULLY LOGGED IN",'\n'

fileout.close()

else:

print "Incorrect username/password.Try again",'\n\n'

self.login()

except EOFError,e:

print e.message

fileout.close()

else:

print "Account does not exist"

print "Would you like to try again?",'\n'

while True:

ch=raw\_input("Enter choice [TRY AGAIN]:(Y/N):")

if ch in ['y','Y']:

print '\n\n'

self.login()

elif ch in ['n','N']:

print "Sign up as Guest",'\n'

self.guest()

else:

print "Invalid option. Try again"

continue

def guest(self):

print "-"\*30

print "GUEST"

print "-"\*30,'\n'

self.name=raw\_input("\*\* Enter your name: ")

while True:

self.phno=raw\_input("\*\* Enter your mobile no.(10 digits must): ")

if self.phno.isdigit():

break

else:

print "Numbers only accepted"

print

self.city=raw\_input("\*\* Enter your city name: ")

self.area=raw\_input("Enter the area of residence: ")

self.address=raw\_input("\*\* Enter your address along with references to Bldg no, flat no etc.: ")

print '\n'

print "Proceed to billing"

class billing(custompizza,allpizza,todayspl,account):

def \_\_init\_\_(self):

self.total=0.0

def isempty(self,item):

if item=='':

return True

else:

return False

def billing\_main(self):

self.total=0.0

lst1=[self.asize,self.psize,self.tsize]

for i in lst1:

if self.isempty(i)==False:

if i=='Small':

self.total+=25

elif i=='Medium' :

self.total+=38

elif i=='Large':

self.total+=50

lst2=[self.acrust,self.pcrust,self.tcrust]

for i in lst2:

if self.isempty(i)==False:

if i=='Pan':

self.total+=2

elif i=='Classic' :

self.total+=2

elif i=='Stuffed Crust':

self.total+=3

lst3=[self.ptop1,self.ptop2,self.ptop3,self.ptop4,self.ptop5,self.ptop6]

for i in lst3:

if self.isempty(i)==False:

self.total+=3

if self.isempty(self.tpizzachoice)==False:

self.total+=3

def display\_info(self):

print '-'\*100

print 'RECEIPT:'

print '-'\*100

lst1=[self.asize,self.psize,self.tsize]

s='-'\*10

for i in lst1:

if self.isempty(i)==False:

if i=='Small':

print i,':'

print s,": AED 25",'\n'

elif i=='Medium' :

print i,':'

print s,": AED 38",'\n'

elif i=='Large':

print i,':'

print s,": AED 50",'\n'

lst2=[self.acrust,self.pcrust,self.tcrust]

for i in lst2:

if self.isempty(i)==False:

if i=='Pan':

print i,':'

print s,": AED 2",'\n'

elif i=='Classic' :

print i,':'

print s,": AED 2",'\n'

elif i=='Stuffed Crust':

print i,':'

print s,": AED 3",'\n'

lst3=[self.ptop1,self.ptop2,self.ptop3,self.ptop4,self.ptop5,self.ptop6]

for i in lst3:

if self.isempty(i)==False:

print i,':'

print s,": AED 3",'\n'

if self.isempty(self.tpizzachoice)==False:

print self.tpizzachoice,':'

print s,": AED 3",'\n'

print "TOTAL:",s,self.total,'\n',"Bill must be paid on delivery",'\n'

print "The delivery will be done within 45 min - 1hr to the location"

print "Thank you for trying 'Mangia Bene' "’

#MAIN PROG

class pi(custompizza,allpizza,todayspl,billing,account):

def \_\_init\_\_(self):

self.apizzachoice=''

self.tpizzachoice=''

self.ptop1=''

self.ptop2=''

self.ptop3=''

self.ptop4=''

self.ptop5=''

self.ptop6=''

self.asize=''

self.psize=''

self.tsize=''

self.acrust=''

self.pcrust=''

self.tcrust=''

def order\_pizza(self):

print '\n','-'\*50

temp=raw\_input("Press to continue to explore")

print '\n', "-"\*30

print "OVERALL MENU"

print "-"\*30,'\n'

print "1.All pizzas"

print "2.Customize ( MAKE YOUR OWN PIZZA )"

print "3.Today's Special"

print "\n"

while True:

ch=raw\_input("Enter choice(PIZZA MENU): (1/2/3)")

print

if ch in ['1','2','3']:

break

else:

print "Invalid! Try again",'\n

if ch=='1':

self.allpizza\_main()

elif ch=='2':

self.custompizza\_main()

elif ch=='3':

self.todayspl\_main()

def cancel\_order(self):

#cancel whole order

print '\n\n',"-"\*50

print "Hoping to see you soon-"

print ' '\*10,'-',"'Mangia Bene '"

pass

def get\_info(self):

# obtain personal info including address

self.account\_main()

self.change\_order()

def calc\_amount(self):

# assign money

self.billing\_main()

def display\_receipt(self):

#print receipt

self.display\_info()

def change\_order(self):

#cancel,modify or continue?

print '\n','-'\*50

print "Would you like to make any changes to your order?"

print

print "1.Cancel order"

print "2.Modify order"

print "3.No change i.e. CONTINUE TO BILLING"

while True:

print

order=raw\_input("Enter choice (1/2/3):")

if order in ['1','2','3']:

break

else:

print "Invalid option"

continue

if order=='1':

self.cancel\_order()

elif order=='2':

self.order\_pizza()

elif order=='3':

print '\n'

pass

self.sub()

def sub(self):

self.calc\_amount()

self.display\_receipt()

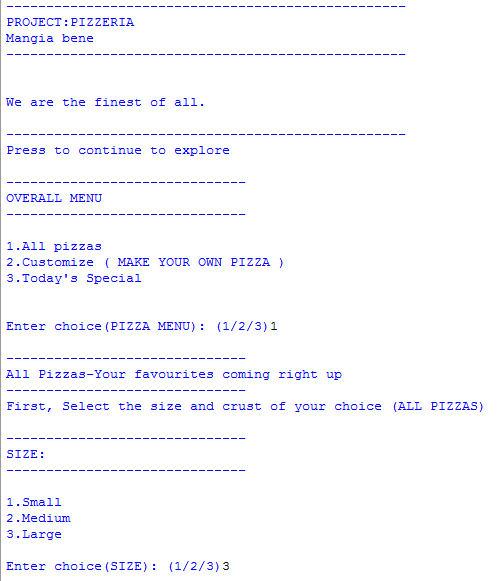
pix=pi()

pix.order\_pizza()

pix.get\_info()

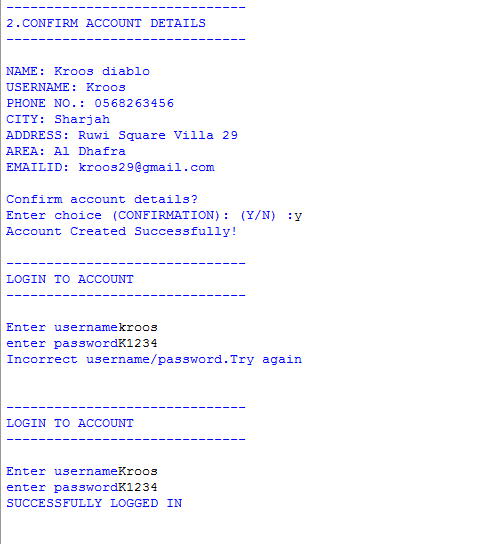
**SAMPLE OUTPUTS**

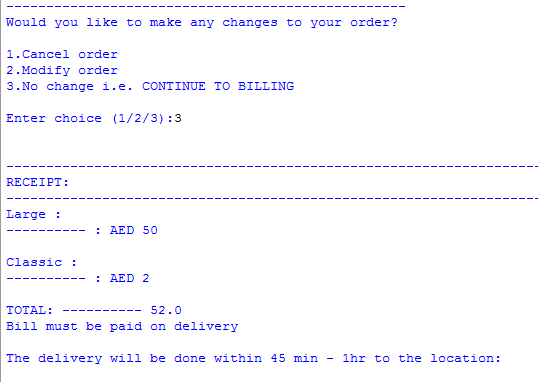
**#1**

****

****

****

****

****

**POSSIBLE IMPROVEMENTS**

* **Use of complex data structures:**

The program can be implemented worth advanced data structures like stacks and queues to reduce memory wastage and increase the program speed and efficiency.

* **Lesser and more compact functions:**

This would ensure less memory wastage. Maintenance and debugging would become simpler.

* **Providing the program online features:**

This will reduce the time occupied by the user and will make the program more advanced.

* **Improving the presentation:**

Readability and understandability by the user can be enhanced by using graphics to make the program more attractive and user-friendly.

**BIBLIOGRAPHY**

* Computer Science Textbook For Grade 12 by Sumita Arora
* [www.wikipedia.org](http://www.wikipedia.org)
* [www.stackoverflow.com](http://www.youtube.com)
* [www.python.o](http://www.livescience.com)rg
* [www.google.c](http://www.learnengineering.org)om
* [www.seoclick.com](http://www.electrical4u.com)