**RESTAURANT MANAGEMENT**

**USING MENU DRIVEN PROGRAM**

**Submitted in partial fulfillment of the requirement for the Internal Contineous Assessment in the Subject “Object Oriented Programming”in Second Year B.Tech(Information Technology) Semester-IV(14pt)**

**Submitted To (12pt)**



**SVKM’s NMIMS,**

**Mukesh Patel School of Technology Management & Engineering,**

**Shirpur Campus (M.H.)**

**Submitted By :**

**RUDRA THORAT [70011118066] –A250**

**ADARSH JHA [70011118026]– A220**

**AAYUSH RAWAT [70011118049]–A240**

**Under The Supervision Of:**

**Ritesh Dhanare Sir**

**(Department of information technology)**

**DEPARTMENT OF INFORMATION TECHNOLOGY (14pt)**

**Mukesh Patel School of Technology Management & Engineering (14pt)**

**SESSION: 2019-20**

### CERTIFICATE

This is to certify that the work embodies in this Mini-Project entitled **“MENU DRIVEN PROGRAM ABOUT A RESTAURANT IN JAVA”** being submitted by

“**Rudra Thorat ” (Roll No.:A250)**

“**Adarsh Jha ” (Roll No.: A220)**

**“Aayush Rawat” (Roll no: A240)**

**--------------------------------------------------------------------------------------**

**for the Internal Continuous Assessment in the Subject “Object Oriented Programming” in Second Year B.Tech(Information Technology) Semester-IV d**uring the academic year 2019-20 **at “Department of Information Technology”, MPSTME, Shirpur (M.H.).** [14pt]

**APPROVED & SUPERVISED BY:(14pt)**

**RITESH DHANARE**

(Department of information technology)

**FORWARDED BY**

# **(Dr. N. S. Choubey)**

H.O.D., IT

MPSTME, Shirpur Campus

**DEPARTMENT OF Information Technology**

**Mukesh Patel School of Technology Management & Engineering (**

### DECLARATION

### We,

### RUDRA THORAT

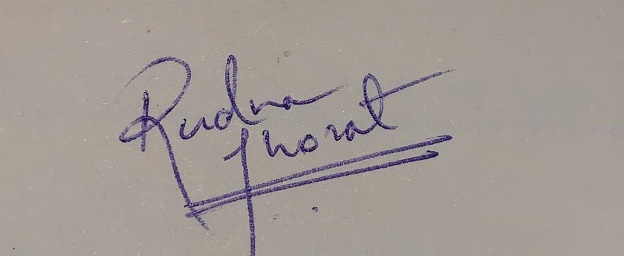
### ADARSH JHA

### AAYUSH RAWAT

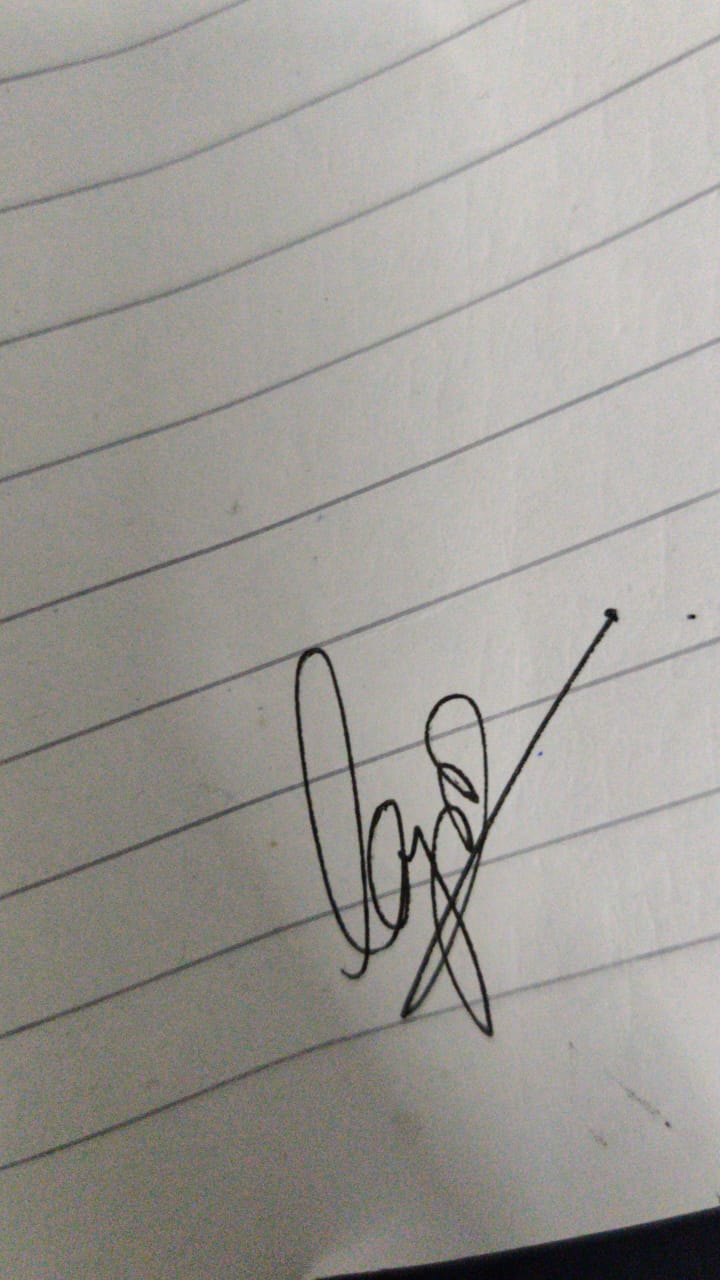
### The students of B. Tech.(Information Technology) Second Year, Session: 2019-20, MPSTME, Shirpur Campus, hereby declare that the work presented in this Project entitled “MENU DRIVEN PROGRAM OF A RESTAURANT IN JAVA” is the outcome of our work, is bonafide and correct to the best of our knowledge and this work has been carried out taking care of Engineering Ethics. The work presented does not infringe any patented work and has not been submitted to any other university or anywhere else for the award of any degree or any professional diploma.

**(RUDRA THORAT)**

#### SAP ID.: 70011118066

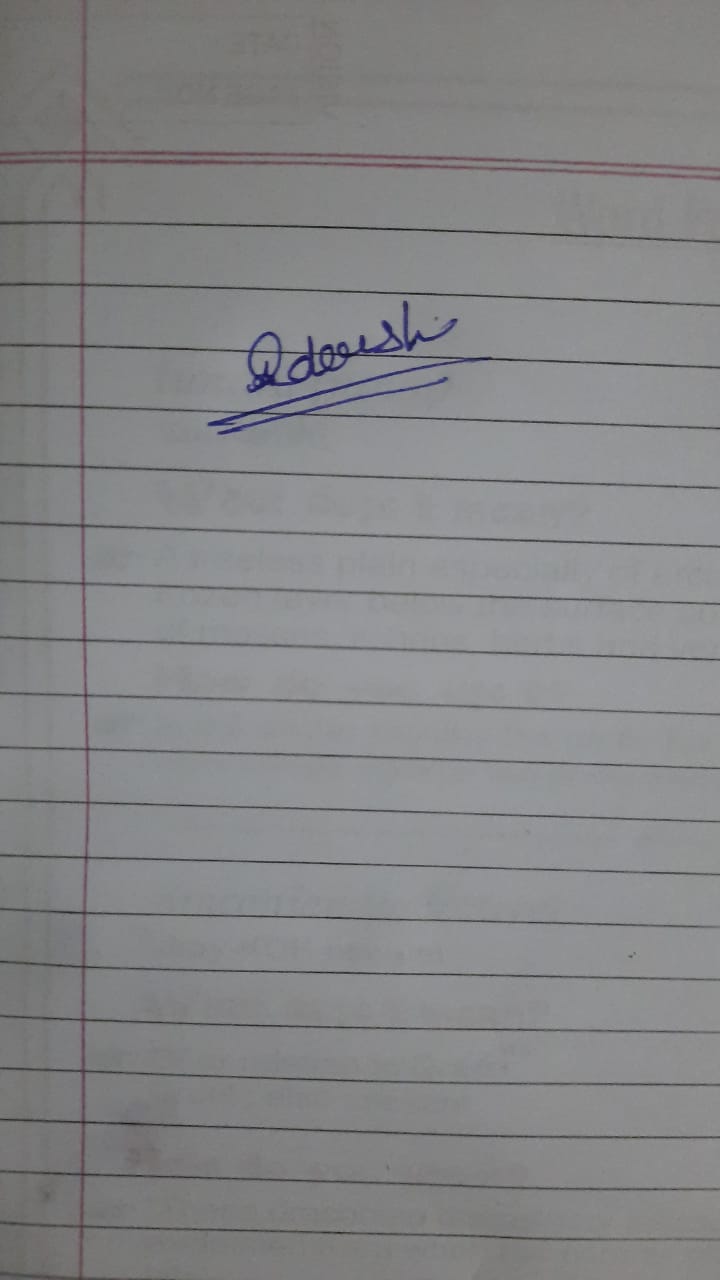
  **(AAYUSH RAWAT)**

#### SAP ID.:70011118049



**(ADARSH JHA)**

**SAP ID – 70011118026**

****

**Date: 10/4//2020**

**DEPARTMENT OF INFORMATION TECHNOLOGY**

**Mukesh Patel School of Technology Management & Engineering**

**ACKNOWLEDGEMENT**

*After the completion of this Mini-Project work, words are not enough to express my feelings about all those who helped me to reach my goal; feeling above this is my indebtedness to The Almighty for providing me this moment in life.*

*It’s a great pleasure and moment of immense satisfaction for me to express my profound gratitude to Prof. Ritesh Dhanare sir , Information Technology Department, MPSTME, Shirpur, whose constant encouragement enabled me to work enthusiastically. Their perpetual motivation, patience and excellent expertise in discussion during progress of the project work have benefited me to an extent, which is beyond expression. Their depth and breadth of knowledge in the field made me realize that theoretical knowledge always helps to develop efficient operational software, which is a blend of all core subjects of the field. I am highly indebted to them for their invaluable guidance and ever-ready support in the successful completion of this project in time. Working under their guidance has been a fruitful and unforgettable experience.*

*We express my sincere thanks and gratitude to Dr. N. S. Choubey, Head of Department, Information Technology Department, MPSTME, Shirpur, for providing necessary infrastructure and help to complete the project work successfully.*

# *We also extend my deepest gratitude to Dr. R.S. Gaud, Director, Shirpur Campus and Dr. Nikhlesh Kumar Sharma, Director (Engineering Programs), Shirpur Campus for providing all the necessary facilities and true encouraging environment to bring out the best of my endeavors.*

*We would like to acknowledge all my friends, who have contributed directly or indirectly in this Mini Project work.*

*The successful completion of a Mini Project is generally not an individual effort. It is an outcome of the cumulative effort of a number of persons, each having their own importance to the objective. This section is a vote of thanks and gratitude towards all those persons who have directly or indirectly contributed in their own special way towards the completion of this project.*

~RUDRA THORAT  
~AAYUSH RAWAT  
~ADARSH JHA

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Chapter No.** | **Page** |
| 1 | Introduction | 7 |
| 2 | Requirements | 9 |
| 3 | Problem Statement | 10 |
| 4 | Objectives | 13 |
| 5 | Project Description | 14 |
| 6 | Class Diagram | 15 |
| 7 | Test Cases | 16 |
| 8 | Source Code | 18 |
| 9 | Output Screen | 44 |
| 10 | User Manual | 46 |
| 11 | Conclusion | 51 |
| 12 | References | 52 |

**MENU DRIVEN PROGRAM OF A RESTAURANT MANAGEMENT**

****

**Introduction**

A program that obtains input from a user by displaying a list of options – the menu – from which the user indicates his/her choice. Systems running menu-driven programs are commonplace, ranging from icroprocessor controlled washing machines to bank cash dispensers. In the case of the cash dispenser, single keys are pressed to indicate the type of transaction (whether a receipt is wanted with the cash, or if a statement of the bank balance is required) and with many, a single key is pressed to indicate the amount of money required.

Menu-driven systems are advantageous in two ways: firstly, because input is via single key strokes, the system is less prone to user error; secondly, because only a limited range of characters are “allowed”, the way in which the input is to be entered is unambiguous. This contributes toward making the system more user-friendly. Compare command-line interface.

**Requirements**

Hardware Requirements

1.Intel Core i3 Laptop (Server) 2.Intel Core i3 Laptop (Client) 3.Android based Smart Phone (Client) 4.Android based Tablet (POS)

Software Requirements

1.Java JDK 2.Netbeans IDE 3.Apache Tomcat Server 4.Mysql Database 5.Android SDK 6.Andoird Eclipse IDE 7.Windows 7 OS

**Problem Statement:**

1.1 Background of the Study

The project is developing because; many restaurants have a lot difficult to manage the business such as customer ordering and reservation table. If the customer book an order and later wants to cancel the order, he is permitted to do this only within a specific time period. By using manual customer ordering it is difficult for the waiter to keep the correct customer information and may lose the customer information. The customer is also given the facility to view the status of the order to determine if it is ready..

Online Restaurant management system is the system for manage the restaurant business. After successfiil login the customer can access the menu page with the items listed according to the desired time. The main point of developing this system is to help restaurant administrator manage the restaurant business and help customer for online ordering and reserve table. In proposed system user can search for a menu according to his choice i.e. according to price range and category of food and later he can order a meal.

1.2 Statement of the Problem

Nowadays, many restaurants manage their business by manual especially take customer ordering. . In traditional booking system, a customer has to go torestaurant or make a phone call in order to get his meal reserved. Today, restaurant waiter takes the customer ordering by manual system with using paper. Customer does some formal conversation like hello, hi, etc. Than he demands for today's menu and do some discussion over menu items then he orders. It takes 5 to 10 minutes to book the order and waiter book the order on paper so there is probability of lost and duplication of customer information. Restaurant management system puts the order in a queue with specific priority according to time and quantity, and then a cook is assigned for the specific order to complete it.

Besides, the restaurant waiter information also by manual system kept use paper and this is difficult for restaurant administrator to find waiter information, probability missing the paper and difficult to arrange the schedule. Initial problem is that the customer has to get connected over the phone; it would be harder if the restaurant is very popular and busy. Sometimes, waiter information and customer information is important to restaurant administrator for reference in the future. The chances of committing mistakes at the restaurant side in providing a menu list for a specific time would be more.

Many people have experienced going to a restaurant where the service is poor and there is a lack of attention from the wait staff The paper menus can be flimsy, hard to navigate, and outdated. To leverage the growing mobile industry, the on —line restaurant proffers solution. This restaurant menu and management system will replace the paper waste, is more maintainable, and allows for greater customer engagement. The problem confronting the research is to determine the Documentation for online restaurant management system.

**Objectives :**

In our java program , our task is to:

1)create an application using java switch-case that would let the users place their order

2)Select a cuisine

3)Select a starter , select a main course, select a dessert.

4)after order is placed, menu would re-appear to let user place order again (if they want to)

5)Proceed to further menu

6)Process to the printing of the bill.

7)Also provide option to exit from the menu application to user.

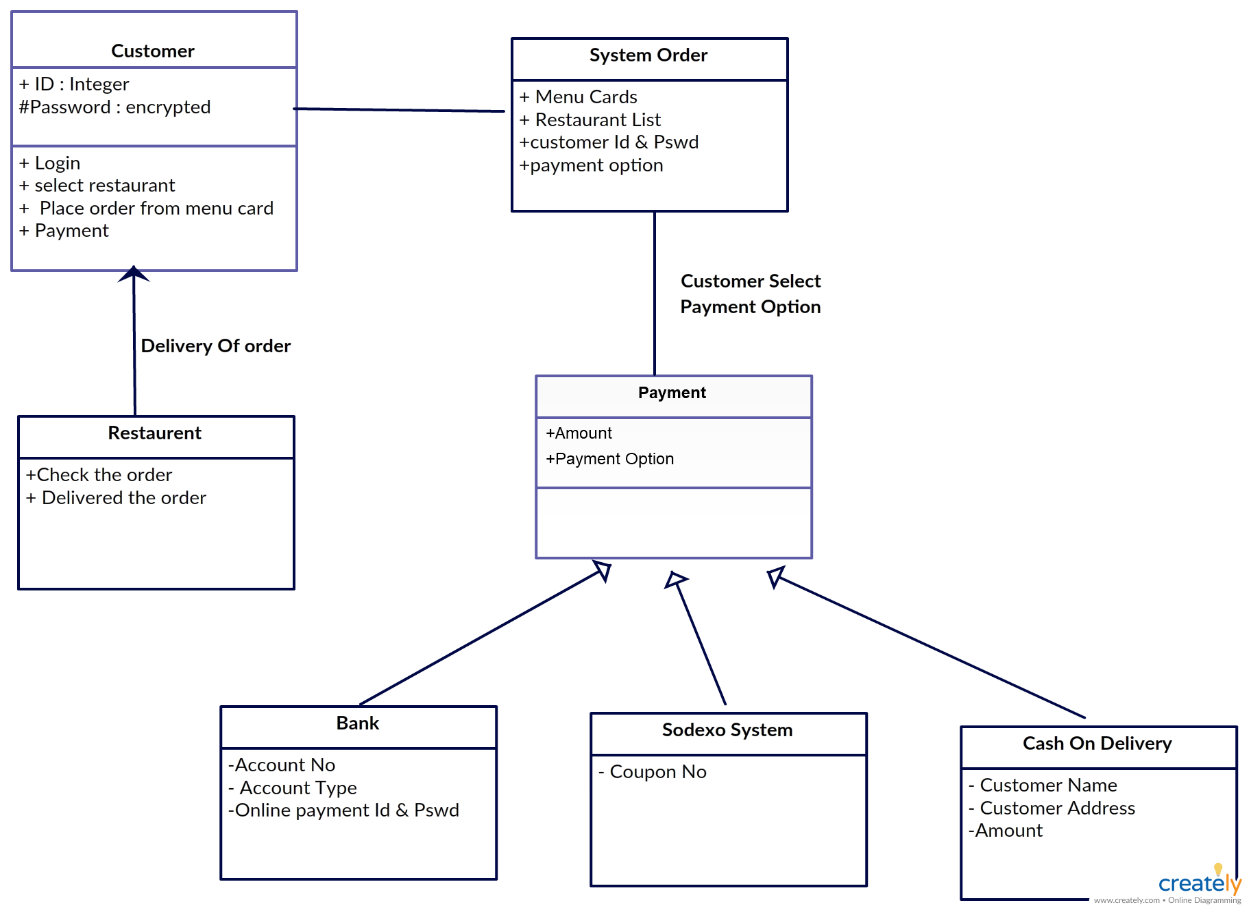
**Project Description :**

We propose to build a software project that can efficiently handle and manage various activities of a restaurant and all these activities will be happening under the supervision of the administrator. The businesses in restaurants are now growing constantly. At the same time, the need for managing its operations and tasks arises. The best way to optimize these activities is growing the business online as well. Today’s generation encourages high-tech services especially over the Internet. Hence the project is developed proficiently to help restaurant owners automate their business operations. This project serves the best way of maintaining customer’s information and caters their needs.

Advantages

* Sometimes it happens that the tables get booked soon during festive seasons therefore user can make advance booking using this system.
* It saves user time when dining in restaurants.
* The system generates online bill for table.
* It saves business’s resources and expenses.

**Class Diagram**

****

**Test Cases:**

Testers will need to validate such scenarios depending on which method is used for food ordering –

Dining – In a case of Dining, guests want to sit in the restaurant and have food, so servers should ensure they are seated on a proper table and at the same time enter the table number with the number of guests in the POS. Testers should test the functionality by inputting a valid value for table number and the number of guests field. Based on the system configuration the validation can be done. Say, table 99 doesn’t exist, so testers while testing provides input as 99 should see an error message like ‘Table # doesn’t exist, please enter a valid table number’. Also, a table shouldn’t accept other orders if a check is already open against that table. Say, guests are already seated on table 10 and if a server enters table # 10, system should throw an error.

Online Ordering – In the case of online ordering the food is ordered via website/mobile and POS are integrated with the online system, so when the order is placed the POS DB gets updated, rest of the push to KDS and other systems can be manual/automated.

For Online Order table entry is not required, since guests order food online and do not visit the restaurant.

Printing a Check Printing a check is the functionality used by servers to print the check/bill which is given to the guests.

Printing functionality can be validated by ensuring the correct menu items are displayed, with correct quantities and amount, tax amount, service charge, total payable amount, discounts if any, coupons used if any, on the check. Checks that are handed to the guests should display complete information.

**Source Code :**

import java.util.\*;

// a menu tye program

public class Lodha\_Zaika//enter class

{

public static void main(String as[])

{

Scanner in=new Scanner(System.in);

int i,m=0,vstr,tvstr=0,nvstr,tnvstr=0,vfd,tvfd=0,nvfd,tnvfd=0,fd,tfd=0,amt=0,tamt=0,totalamt=0,d,totald=0,damt=0,ch;

double vat=0;

int conti=0,conti1=0;//initiallising variables

String str="",ans,choice="Y";

System.out.println("## ##\t##############\t##############\t###############\t## ");

System.out.println("## ##\t##############\t##############\t###############\t## ");

System.out.println("## ##\t## ##\t ## \t## \t## ");

System.out.println("## ##\t## ##\t ## \t## \t## ");

System.out.println("## ##\t## ##\t ## \t## \t## ");

System.out.println("##############\t## ##\t ## \t############## \t## ");

System.out.println("##############\t## ##\t ## \t############## \t## ");

System.out.println("## ##\t## ##\t ## \t## \t## ");

System.out.println("## ##\t## ##\t ## \t## \t## ");

System.out.println("## ##\t## ##\t ## \t## \t## ");

System.out.println("## ##\t##############\t ## \t###############\t##############");

System.out.println("## ##\t##############\t ## \t###############\t##############");

System.out.println(" Welcome to our Multi cuisine restaurant : LODHA ZAIKA!");

do

{

conti1=0;

System.out.println("Starters : 1");

System.out.println("Main course : 2");

System.out.println("Desserts : 3");

System.out.println();

System.out.println("Enter a choice from the above list to enjoy!");

ch=in.nextInt();

System.out.print("Loading");

for(i=0;i<1000000000;i++)

{

if(m==100000000)

{

System.out.print(".");

m=0;

}

m++;

}

System.out.println();

switch(ch)

{

case 1:

do

{

conti=0;

System.out.println("STARTERS");

System.out.println("Enter 'VS' to have a taste of our vegetarian delicacies ");

System.out.println("Enter 'NVS' to have a taste of our non-vegetarian delicacies ");

str=in.next();

System.out.print("Loading");

for(i=0;i<1000000000;i++)

{

if(m==100000000)

{

System.out.print(".");

m=0;

}

m++;

}

System.out.println();

if(str.equalsIgnoreCase("VS"))

{//veg starters menu

System.out.println("|-------------------------------|");

System.out.println("| VEGETARIAN STARTERS |");

System.out.println("|-------------------------------|");

System.out.println(" Delicacy \t\t\t Price(in Rs)");

System.out.println("1. MOROCCAN CHICKPEA SOUP \t\t\t 1100");

System.out.println("2. RAVIOLI WITH SQUASH AND CRUNCHY CRUMBS \t\t\t 1100");

System.out.println("3. BAKED STUFFED ROMANO PEPPERS \t\t\t 1100");

System.out.println("4. WILD MUSHROOM TARTLETS \t\t\t 1300");

System.out.println("5. HONEYED WINTER SALAD \t\t\t 1300");

System.out.println("6. MEXICAN BEAN SALAD \t\t\t 1300");

System.out.println("7. STUFFED COURGETTE ROLLS \t\t\t 1400");

System.out.println("8. SPICY ROASTED PARNSNIP SOUP \t\t\t 1400");

System.out.println("9. SEEDED FLATBREADS \t\t\t 1400");

System.out.println("10. CRISPY AMERICAN CORN \t\t\t 1400");

System.out.println();

choice="Y";

while(choice.equalsIgnoreCase("Y"))

{

{

System.out.println("To have a great start choose your starter from the above delicacy by entering the number");

vstr=in.nextInt();

System.out.println("Enter the total number of starters you wish to have");

tvstr=in.nextInt();

if(vstr==1 || vstr<=3)

amt=tvstr\*1100;

if(vstr==4 ||vstr==5|| vstr==6)

amt=tvstr\*1300;

if(vstr>=7)

amt=tvstr\*1400;

}

tamt=tamt+amt;

System.out.println("Do you want to place order of a different type of starter press 'y'");

System.out.println("If you want to have the same choice of yours again press 'n'");

System.out.println("If you want to Exit the current cuisine type'Main'");

choice=in.next();

System.out.print("Loading ");

for(i=0;i<1000000000;i++)

{

if(m==100000000)

{

System.out.print(".");

m=0;

}

m++;

}

System.out.println();

}

if(choice.equalsIgnoreCase("n"))

{

conti=1;

}

}

if(str.equalsIgnoreCase("NVS"))

{// non veg starters menu

System.out.println("|-------------------------------|");

System.out.println("| NON-VEGETARIAN STARTERS |");

System.out.println("|-------------------------------|");

System.out.println(" DELICACY \t\t\t Price(in Rs)");

System.out.println("1. CORNFLAKES CRUSTED CHICKEN NUGGETS \t\t\t 1500");

System.out.println("2. CHICKEN CHEESE POPS \t\t\t 1500");

System.out.println("3. BARLEY AND CHICKEN NUGGETS \t\t\t 1500");

System.out.println("4. DUMDAR PRAWNS \t\t\t 1700");

System.out.println("5. CREAMY CHICKEN TOAST \t\t\t 1700");

System.out.println("6. CHICKEN VAL-AU-VENT \t\t\t 1700");

System.out.println("7. CHICKEN LOLLYPOP \t\t\t 1800");

System.out.println("8. POLENTA PESTO CHICKEN \t\t\t 1800");

System.out.println("9. CRISPY BARBEQUE CHICKEN \t\t\t 1800");

System.out.println("10.CHICKEN LAZONE \t\t\t 1800");

System.out.println();

choice="Y";

while(choice.equalsIgnoreCase("Y"))

{

System.out.println("Choose your starter from the above delicacy by entering the number");

nvstr=in.nextInt();

System.out.println("Enter the total number of starters you want");

tnvstr=in.nextInt();

if(nvstr==1||nvstr<=3)

amt=tnvstr\*1500;

if(nvstr==4||nvstr==5|| nvstr==6)

amt=tnvstr\*1700;

if(nvstr>=7 )

amt=tnvstr\*1800;

tamt=tamt+amt;

System.out.println("Do you want to place order of a different type of food press 'y");

System.out.println("If you want to have the same choice of yours again press 'n'");

System.out.println("If you want to Exit the current cuisine type'Main'");

choice=in.next();

}

if(choice.equalsIgnoreCase("n"))

{

conti=1;

}

}

}while(conti==1);

break;

case 2:

do

{

conti=0;

System.out.println("Main Course:Indian and Chinese Dishes!");

System.out.println("Enter 'V' to have a taste of our Indian Veg Dishes");

System.out.println("Enter 'NV' to have a taste of our non-vegetarian Indian main course dishes ");

System.out.println("Enter 'C' to have a taste of our chinese dishes ");

str=in.next();

System.out.print("Loading");

for(i=0;i<1000000000;i++)

{

if(m==100000000)

{

System.out.print(".");

m=0;

}

m++;

}

System.out.println();

if(str.equalsIgnoreCase("V"))

{// vegtarian maincourse menu

System.out.println("|-------------------------------|");

System.out.println("| INDIAN VEGETARIAN MAIN COURSE |");

System.out.println("|-------------------------------|");

System.out.println(" DELICACY \t\t\t Price(in Rs)");

System.out.println("1. DAHI KI SABZI \t\t\t1800");

System.out.println("2. BHINDI KI SABZI \t\t\t1800");

System.out.println("3. ALOO KI SABZI \t\t\t1500");

System.out.println("4. GREEN PEA MASH \t\t\t1400");

System.out.println("5. GARLIC GRILLED TOMATOES \t\t\t1400");

System.out.println("6. FRENCH BEANS WITH MOONG DAL \t\t\t1400");

System.out.println("7. CREAMY MUSHROOM \t\t\t1500");

System.out.println("8. CHOLE CHANNA DAL \t\t\t1500");

System.out.println("9. CHILLI PANEER \t\t\t1100");

System.out.println("10. TITORI \t\t\t1400");

System.out.println("11. ANANAS MENSKAI \t\t\t400");

System.out.println("12. SHAHI PANNER \t\t\t600");

choice="Y";

while(choice.equalsIgnoreCase("Y"))

{

System.out.println("Choose your Main Veg Course from the delicacy by entering number:");

vfd=in.nextInt();

System.out.println("How many plates do you want to place from the above delicacy?");

tvfd=in.nextInt();

if(vfd==1||vfd==2)

amt=tvfd\*1800;

if(vfd==3)

amt=tvfd\*1500;

if(vfd==4||vfd==5||vfd==6)

amt=tvfd\*1400;

if(vfd==7||vfd==8)

amt=tvfd\*1500;

if(vfd==9)

amt=tvfd\*1100;

if(vfd==10)

amt=tvfd\*140;

if(vfd==11)

amt=tvfd\*400;

if(vfd==12)

amt=tvfd\*600;

tamt=tamt+amt;

System.out.println("Do you want to place order of a different type of food press 'y'");

System.out.println("if you want to have the same choice of yours again press 'n'");

System.out.println("IF you want to Exit the current cuisine type'Main'");

choice=in.next();

System.out.print("Loading");

for(i=0;i<1000000000;i++)

{

if(m==100000000)

{

System.out.print(".");

m=0;

}

m++;

}

System.out.println();

}

if(choice.equalsIgnoreCase("n"))

{

conti=1;

}

}

if(str.equalsIgnoreCase("NV"))

{//non veg main course menu

System.out.println("|-----------------------------------|");

System.out.println("| INDIAN NON-VEGEARIAN MAIN COURSE |");

System.out.println("|-----------------------------------|");

System.out.println(" DELICACY \t\t\t: Price(in Rs.)");

System.out.println("1. EGG FRIED QUINOA \t\t\t1800");

System.out.println("2. BAKED EGGS \t\t\t1800");

System.out.println("3. HOT GARLIC EGGS \t\t\t1500");

System.out.println("4. GRILLED TAWA FISH \t\t\t600");

System.out.println("5. CABBAGE PRAWNS \t\t\t1600");

System.out.println("6. BRAISED CHICKEN WITH WALNUTS \t\t\t1600");

System.out.println("7. CHICKEN CURRY \t\t\t1700");

System.out.println("8. CHARGRILLED CHICKEN WITH THYME \t\t\t1700");

System.out.println("9. GRILLED CHICKEN WITH PINEAPPLE SALSA \t\t\t1900");

System.out.println("10. CHICKEN IN PRUNE SAUCE \t\t\t1900");

System.out.println("11. CHICKEN AND PEPPER STIR FRY \t\t\t1400");

System.out.println("12. BANJARA GOSHT \t\t\t1600");

choice="Y";

while(choice.equalsIgnoreCase("Y"))

{

System.out.println("Choose your Main Non-Veg Course from the delicacy by entering number:");

vfd=in.nextInt();

System.out.println("how many plates do you want to place from the above delicacy?");

tvfd=in.nextInt();

if(vfd==1||vfd==2)

amt=tvfd\*1800;

if(vfd==3)

amt=tvfd\*1500;

if(vfd==4||vfd==5||vfd==6)

amt=tvfd\*1600;

if(vfd==7||vfd==8)

amt=tvfd\*1700;

if(vfd==9||vfd==10)

amt=tvfd\*1900;

if(vfd==11)

amt=tvfd\*1400;

if(vfd==12)

amt=tvfd\*1600;

tamt=tamt+amt;

System.out.println("Do you want to place order of a different type of food press 'y'");

System.out.println("if you want to have the same choice of yours again press 'n'");

System.out.println("IF you want to exit the current cuisine type'Main'");

choice=in.next();

System.out.print("Loading");

for(i=0;i<1000000000;i++)

{

if(m==100000000)

{

System.out.print(".");

m=0;

}

m++;

}

System.out.println();

}

if(choice.equalsIgnoreCase("n"))

{

conti=1;

}

}

if(str.equalsIgnoreCase("C"))

{//chinese menu

System.out.println("|-------------------------------|");

System.out.println("| CHINESE MAIN COURSE | ");

System.out.println("|-------------------------------|");

System.out.println(" DELICACY \t\t\t Price(in Rs.)");

System.out.println("1. GENERAL TSO'S CHICKEN \t\t\t 2400");

System.out.println("2. TOFU \t\t\t 2800");

System.out.println("3. LAMB SKEWERS WITH CUMIN \t\t\t 2800");

System.out.println("4. BLACK PEPPER TOFU \t\t\t 2100");

System.out.println("5. BLACK PEPPER AND SHRIMP FRIED RICE \t\t\t 2100");

System.out.println("6. GARLIC BLACK BEAN PORK FRIED RICE \t\t\t 1900");

System.out.println("7. GRILLED CHICKEN AND BIBB LETTUCE \t\t\t 1900");

System.out.println("8. PORK STIR FRY WITH BABY CHOY AND CASHEWS \t\t\t 1900");

System.out.println("9. CLASSIC SESAME NOODLES \t\t\t 2400");

System.out.println("10. SAUSAGE AND BROCCOLI WITH OYSTER SAUCE \t\t\t 2100");

System.out.println("11. CLASSIC DRY FRIED PEPPER AND SALT SHRIMP \t\t\t 2100");

System.out.println("12. RED COOKED PORK SHOULDER AND CHESTNUTS \t\t\t 2100");

choice="Y";

while(choice.equalsIgnoreCase("Y"))

{

System.out.println("Choose your main course from the above delicacy by entering number");

fd=in.nextInt();

System.out.println("How many plates do you want to place from the above delicacy?");

tfd=in.nextInt();

if(fd==1)

amt=tfd\*2400;

if(fd==2||fd==3)

amt=tfd\*2800;

if(fd==4||fd==5)

amt=tfd\*2100;

if(fd==6||fd==7||fd==8)

amt=tfd\*1900;

if(fd==9)

amt=tfd\*2400;

if(fd==10||fd==11||fd==12)

amt=tfd\*2400;

tamt=tamt+amt;

System.out.println("Do you want to place order of a different type of food press 'y'");

System.out.println("if you want to have the same choice of yours again press 'n'");

System.out.println("IF you want to Exit the current cuisine type'Main'");

choice=in.next();

System.out.print("Loading");

for(i=0;i<1000000000;i++)

{

if(m==100000000)

{

System.out.print(".");

m=0;

}

m++;

}

System.out.println();

}

if(choice.equalsIgnoreCase("n"))

{

conti=1;

}

}

}while(conti==1);

System.out.println();

break;

case 3:

do// Desert menu

{

conti=0;

System.out.println("|-------------------------------|");

System.out.println("| DESSERTS | ");

System.out.println("|-------------------------------|");

System.out.println(" DELICACY \t\t\t Price(in Rs.)");

System.out.println("1. WALNUT BAKLAVA \t\t\t 1100");

System.out.println("2. CHOCO BITE ICE CREAM \t\t\t 1100");

System.out.println("3. BASIC SPONGE CAKE \t\t\t 1900");

System.out.println("4. CHOCOLATE STUFFED WITH PEDHA \t\t\t 1900");

System.out.println("5. SANTRA BASUNDI \t\t\t 1700");

System.out.println("6. COCONUT BARFI \t\t\t 2900");

System.out.println("7. ATTA BESAN LADDOO \t\t\t 2900");

System.out.println("8. PEANUT AND DRY FRUIT LADDOO \t\t\t 2800");

System.out.println("9. TIL AND DRY FRUITS LADDOO \t\t\t 2800");

System.out.println("10. DRY FRUIT SHRIKHAND \t\t\t 2800");

choice="Y";

while(choice.equalsIgnoreCase("Y"))

{

System.out.println("To melt your burning mouth,choose your desert by entering the number!");

d=in.nextInt();

System.out.println("Enter the total number of items you want to buy!");

totald=in.nextInt();

if(d==1||d==2)

damt=totald\*1100;

if(d==3||d==4)

damt=totald\*1900;

if(d==5)

damt=totald\*1700;

if(d==6||d==7)

damt=totald\*2900;

if(d==8||d==9||d==10)

damt=totald\*2800;

tamt=tamt+damt;

System.out.println("Do you want to place order of a different type of food press 'y'");

System.out.println("IF you want to Exit the current cuisine type'Main'");

choice=in.next();

System.out.print("Loading");

for(i=0;i<1000000000;i++)

{

if(m==100000000)

{

System.out.print(".");

m=0;

}

m++;

}

System.out.println();

}

if(choice.equalsIgnoreCase("n"))

{

conti=1;

}

System.out.println();

}while(conti==1);

break;

default:

System.out.println("You have entered wrong choice!");

System.out.println("You can enter in this Multi Cuisine Restaurant~ LODHA ZAIKA by executing the program again with the correct choice!");

System.out.println("Now, 'Quit' for the Bill");

}

System.out.println("Enter 'Bill' to get your Bill printed");

System.out.println("Enter 'Menu' if you want to have something other than what you ordered ");

ans=in.next();

System.out.print("Loading");

for(i=0;i<1000000000;i++)

{

if(m==100000000)

{

System.out.print(".");

m=0;

}

m++;

}

System.out.println();

if(ans.equalsIgnoreCase("Bill"))//PRinting of the bill

{

System.out.println("################\t################\t## \t## ");

System.out.println("## ##\t################\t## \t## ");

System.out.println("## ## \t ## \t## \t## ");

System.out.println("## ## \t ## \t## \t## ");

System.out.println("## ## \t ## \t## \t## ");

System.out.println("########### \t ## \t## \t## ");

System.out.println("########### \t ## \t## \t## ");

System.out.println("## ## \t ## \t## \t## ");

System.out.println("## ## \t ## \t## \t## ");

System.out.println("## ## \t ## \t## \t## ");

System.out.println("## ##\t################\t################\t################");

System.out.println("################\t################\t################\t################");

System.out.println();

System.out.println();

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.println("\* \t\*");

System.out.println("\*\tTotal Cost \t= Rs."+tamt+(".0")+"\t\*");

System.out.println("\* \t\*");

vat=Math.round(14.5/100.0\*tamt);

System.out.println("\*\tValue Added Tax(14.5%)\t= Rs."+vat+ " \t\*");

System.out.println("\*\tAmount to be paid \t= Rs."+(tamt+vat)+" \t\*");

System.out.println("\* \t\*");

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.println();

System.out.println("Thanks for visiting Lodha Zaika!");

System.out.println("Your pleasure Our Comfort!");

System.out.println("Visit Again");

System.out.println();

conti1=0;

}

else

{

conti1=1;

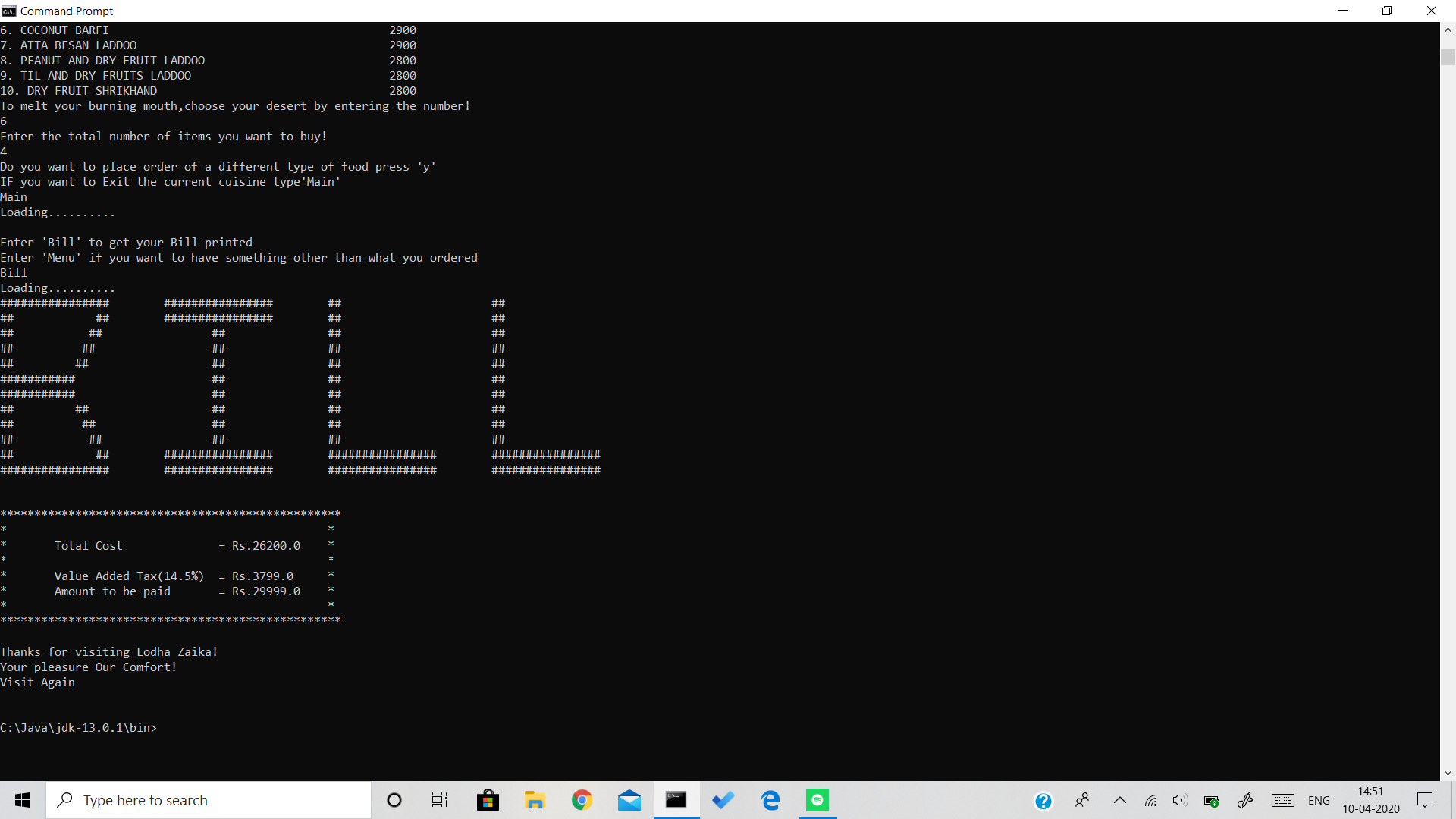
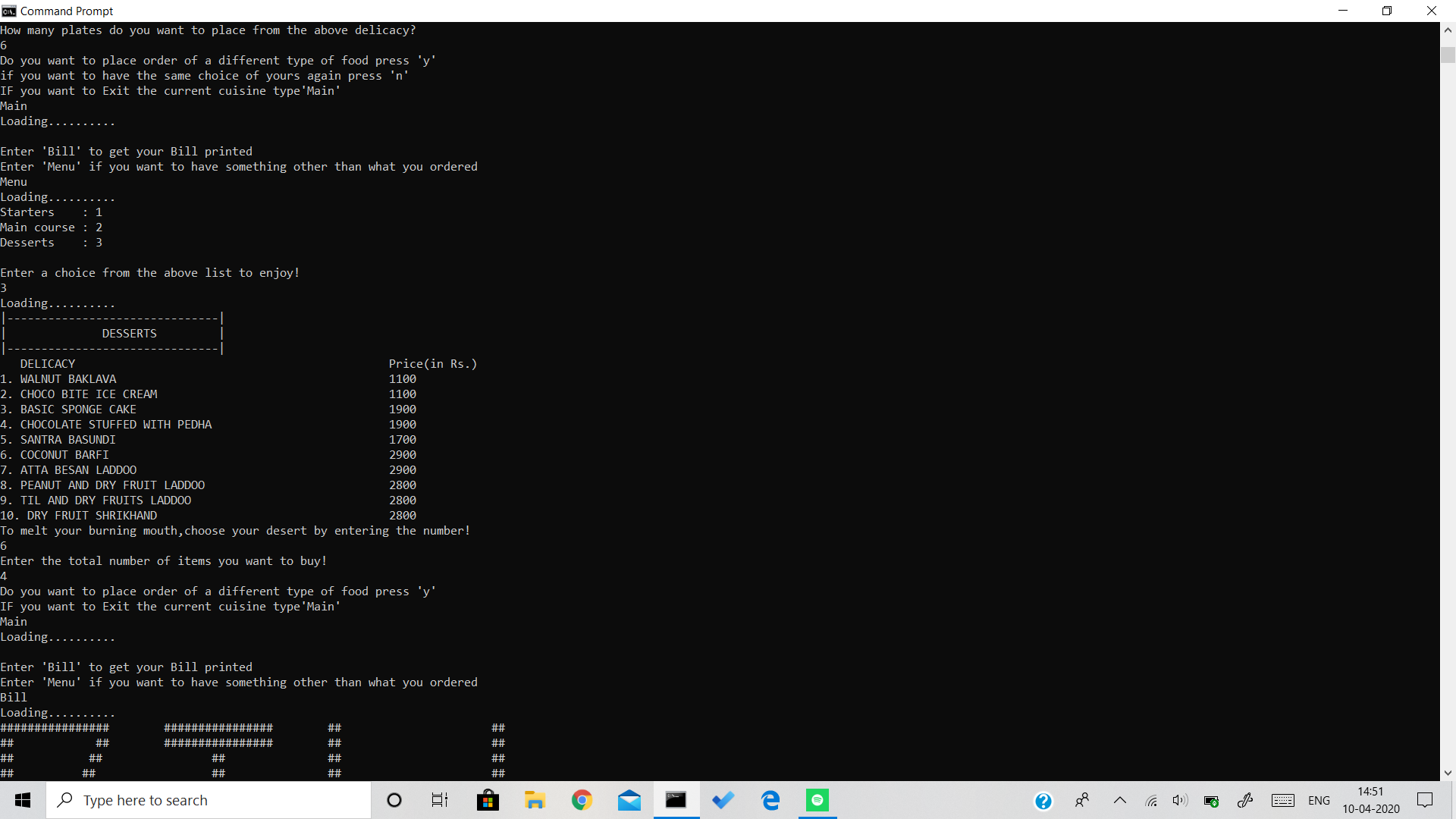
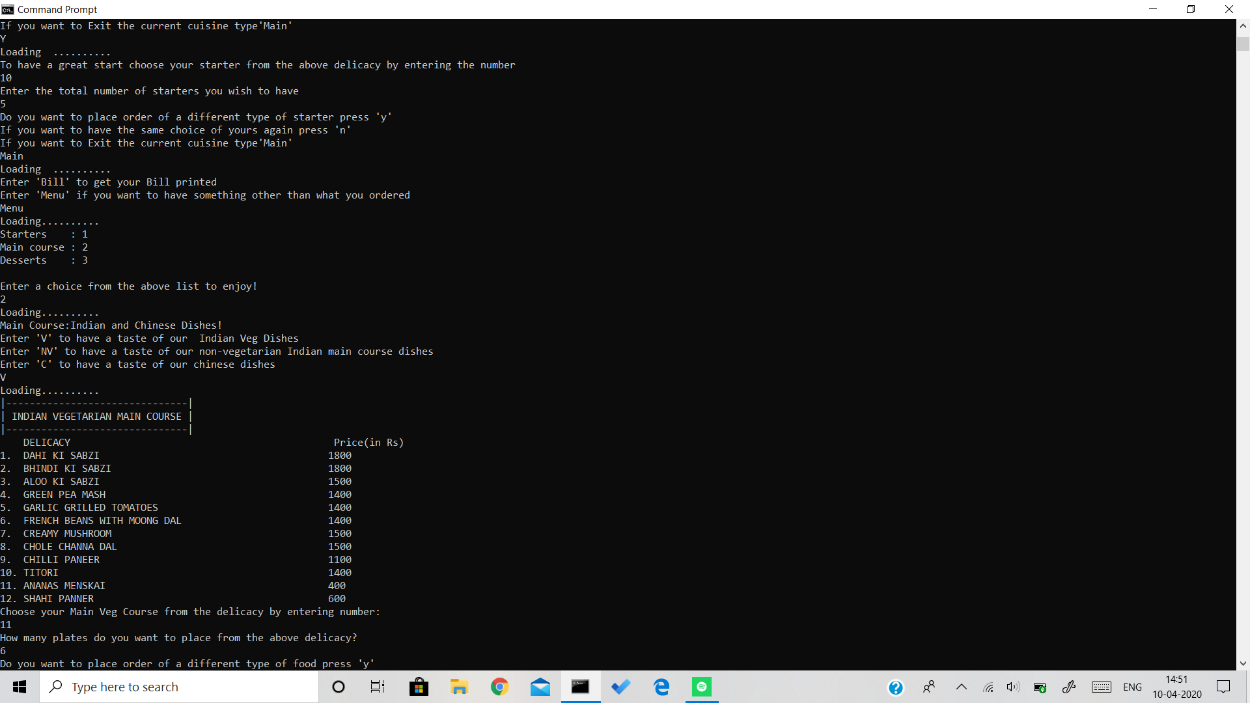
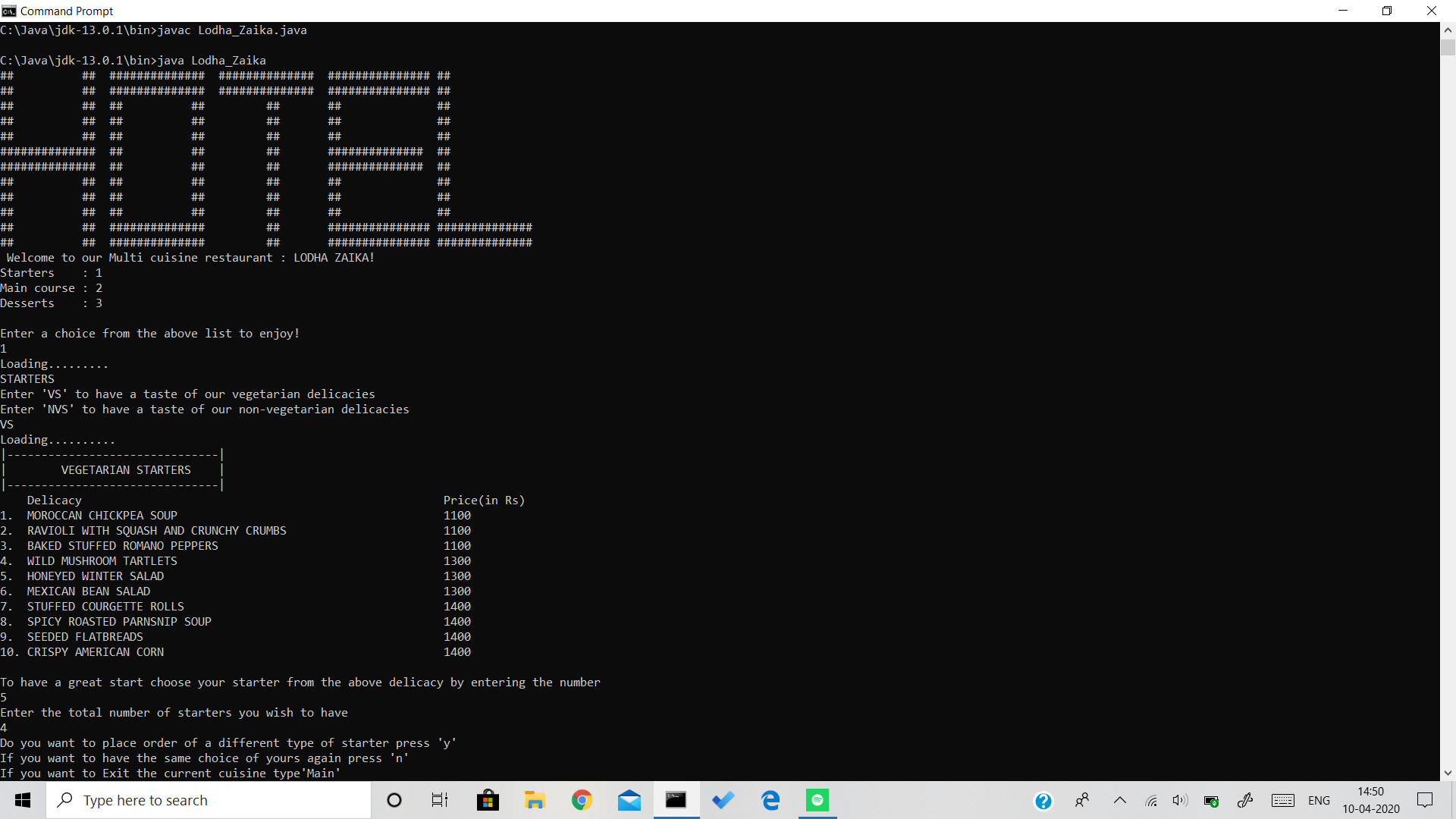
}

}while(conti1==1);

}

}//End of the program

**Output/Screen :**

****

**USER MANUAL**

This is a JAVA based Application, requirements to run this application are:

1. Desktop/Laptop

2. OS (LINUX/Windows/MACOS)

3. JAVA Compiler

Steps to install Java compiler:

In Windows-

Step 0: Un-Install Older Version(s) of JDK/JRE

I recommend that you install only the latest JDK. Although you can install multiple versions of JDK/JRE concurrently, it is messy.

If you have previously installed older version(s) of JDK/JRE, un-install ALL of them. Go to "Control Panel" ⇒ (optional) "Programs" ⇒ "Programs and Features" ⇒ Un-install ALL programs begin with "Java", such as "Java SE Development Kit ...", "Java SE Runtime ...", "Java X Update ...", and etc.

Step 1: Download JDK

Goto http://www.oracle.com/technetwork/java/javase/downloads/index.html.

Under "Java Platform, Standard Edition" ⇒ "Java SE 13.0.{x}", where {x} denotes a fast running security-update number ⇒ Click the "Oracle JDK Download" button.

Under "Java SE Development Kit 13.0.{x}" ⇒ Check "Accept License Agreement".

Choose the JDK for your operating system, i.e., "Windows". Download the "exe" installer (e.g., "jdk-13.0.{x}\_windows-x64\_bin.exe" - about 159MB).

Step 2: Install JDK

Run the downloaded installer (e.g., "jdk-13.0.{x}\_windows-x64\_bin.exe"), which installs both the JDK and JRE.

By default, JDK is installed in directory "C:\Program Files\Java\jdk-13.0.{x}", where {x} denotes the update number. Accept the defaults and follow the screen instructions to install JDK.

Use your "File Explorer", navigate to "C:\Program Files\Java" to inspect the sub-directories. Take note of your JDK installed directory jdk-13.0.{x}, in particular, the update number {x}, which you will need in the next step.

I shall refer to the JDK installed directory as <JAVA\_HOME>, hereafter, in this article.

Step 3: Include JDK's "bin" Directory in the PATH

Windows' Command Prompt (CMD) searches the current directory and the directories listed in the PATH environment variable (or system variable) for executable programs. JDK's programs (such as Java compiler "javac.exe" and Java runtime "java.exe") reside in the sub-directory "bin" of the JDK installed directory. You need to include JDK's "bin" in the PATH to run the JDK programs.

To edit the PATH environment variable in Windows 10:

Launch "Control Panel" ⇒ (Optional) "System and Security" ⇒ "System" ⇒ Click "Advanced system settings" on the left pane.

Switch to "Advanced" tab ⇒ Click "Environment Variables" button.

Under "System Variables" (the bottom pane), scroll down to select variable "Path" ⇒ Click "Edit...".

For Newer Windows 10:

You shall see a TABLE listing all the existing PATH entries (if not, goto next step). Click "New" ⇒ Click "Browse" and navigate to your JDK's "bin" directory, i.e., "c:\Program Files\Java\jdk-13.0.{x}\bin", where {x} is your installation update number ⇒ Select "Move Up" to move this entry all the way to the TOP.

For Older Windows 10 (Time to change your computer!):

(CAUTION: Read this paragraph 3 times before doing this step! Don't push "Apply" or "OK" until you are 101% sure. There is no UNDO!!!)

(To be SAFE, copy the content of the "Variable value" to Notepad before changing it!!!)

In "Variable value" field, APPEND "c:\Program Files\Java\jdk-13.0.{x}\bin" (where {x} is your installation update number) IN FRONT of all the existing directories, followed by a semi-colon (;) to separate the JDK's bin directory from the rest of the existing

Directories. DO NOT DELETE any existing entries; otherwise, some existing applications may not run.

Variable name: PATH

Variable value: c:\Program Files\Java\jdk-13.0.{x}\bin;[do not delete exiting entries...]

Note: If you have started CMD, you need to re-start for the new environment settings to take effect.

Step 4: Verify the JDK Installation

Launch a CMD via one of the following means:

Click "Search" button ⇒ Type "cmd" ⇒ Choose "Command Prompt", or

Right-click "Start" button ⇒ run... ⇒ enter "cmd", or

Click "Start" button ⇒ Windows System ⇒ Command Prompt

Issue the following commands to verify your JDK installation:

Issue "path" command to list the contents of the PATH environment variable. Check to make sure that your JDK's "bin" is listed in the PATH.

PATH=c:\Program Files\Java\jdk-13.0.{x}\bin;other entries...

Issue the following commands to verify that JDK/JRE are properly installed and display their version:

// Display the JDK version

javac -version

javac 13.0.1

// Display the JRE version

java -version

java version "13.0.1" 2019-10-15

Java(TM) SE Runtime Environment (build 13.0.1+9)

Java HotSpot(TM) 64-Bit Server VM (build 13.0.1+9, mixed mode, sharing)

FOR MACOS-

Download the JDK .dmg file, jdk-10.interim.update.patch-macosx-x64.dmg.

Before the file can be downloaded, you must accept the license agreement.

From either the browser Downloads window or from the file browser, double-click the .dmg file to start it.

A Finder window appears that contains an icon of an open box and the name of the .pkg file.

Double-click the JDK 10.pkg icon to start the installation application.

The installation application displays the Introduction window.

Click Continue.

The Installation Type window appears.

Click Install.

A window appears that displays the message: Installer is trying to install new software. Enter your password to allow this.

Enter the Administrator user name and password and click Install Software.

The software is installed and a confirmation window is displayed.

FOR LINUX-

At the time of writing, the latest LTS version of Java is version 11.

Follow the steps below to install Java OpenJDK 11 on your Ubuntu system:

First, update the apt package index with:

sudo apt update

Once the package index is updated install the default Java OpenJDK package with:

sudo apt install default-jdk

Verify the installation, by running the following command which will print the Java version:

java -version

The output will look something like this:

openjdk version "11.0.2" 2019-01-15OpenJDK Runtime Environment (build 11.0.2+9-Ubuntu-3ubuntu118.04.3)OpenJDK 64-Bit Server VM (build 11.0.2+9-Ubuntu-3ubuntu118.04.3, mixed

Now Compile the test.java file and run it.

You have successfully entered the Application.

**Conclusion**

We conclude by the following project that, We have successfully prepared a menu driven program of a hotel, by adding cuisines, type of food to be ordered that is Starter, main course or a dessert.

We have added the ability for the user to pick any dish from the menu provided and ask for the net quantity of the dish to be ordered.

The user can proceed back to the main menu and order it accordingly .The user can also wish to stop at any moment if needed.

The user after his completion of the meal, will be asked upon the print of the total bill summary. This bill summary will list out all the items that were ordered and print the total that the user has to pay to the restaurant. The program will have a walkthrough to all these scenarios. **We had successfully completed the project in timely manner by considering all the user requirements.**

**References:**

Eclipse IDE for Java Developers

~2019-20

**The essential tools for any Java developer, including a Java IDE, a CVS client, Git client, XML Editor, Mylyn, Maven integration and WindowBuilder**

[**https://youtu.be/cv6GvRCIuTs**](https://youtu.be/cv6GvRCIuTs)

[**https://youtu.be/g0PrXoWKM2Y**](https://youtu.be/g0PrXoWKM2Y)