**TOPIC-**

PARADISE

****

****

**NAME: RIDA MUSKAAN**

**CLASS: X**

**SECTION: H**

**SCHOOL: NASR GIRLS SCHOOL**

**YEAR: 2016-2017**

**SUBJECT: COMPUTER APPLICATIONS**

**ACKNOWLEDGEMENT**

**I express my heartfelt gratitude to Mrs.Sunitha, my Computer Teacher who has directed me to work in the right manner and to accomplish complete knowledge of the JAVA language.**

**She has used her manoeuvre which has helped me in finishing the project to perfection.**

**I thank my parents for helping me in getting the copy of the project and encouraging me during the course of finishing the project.**

**Thus this project has become a great learning experience for me and has enriched my skills in JAVA programming.**

**INDEX**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **TOPIC** | **PAGE NO.** |
| **1.** | ACKNOWLEDGEMENT | **1** |
| **2.** | INTRODUCTION TO BLUEJ | **3-4** |
| **3.** | INTRODUCTION TO TOPIC | **5-6** |
| **4.** | VARIABLE DESCRIPTION | **7-11** |
| **5.** | METHOD DESCRIPTION | **12-13** |
| **6.** | PROGRAM | **14-33** |
| **7.** | OUTPUT SCREENS | **34-40** |
| **8.** | SCOPE OF EXPANSION | **41** |
| **9.** | CONCLUSION | **42-43** |
| **10.** | BIBLIOGRAPHY | **44** |

**INTRODUCTION TO BLUEJ**

**BlueJ  is an integrated development environment (IDE) for the Java programming language, developed mainly for educational purposes, but also suitable for small-scale software development. It runs with the help of JDK (Java Development Kit).**

**BlueJ was developed to support the learning and teaching of**[**object-oriented programming**](https://en.wikipedia.org/wiki/Object-oriented_programming)**. The main screen graphically shows the class structure of an application under development in a**[**UML**](https://en.wikipedia.org/wiki/Unified_Modeling_Language) **(Unified Modelling Language) like diagram, and objects can be interactively created and tested. This interaction facility, combined with a clean, simple user interface, allows easy experimentation with objects under development. Object-oriented concepts (objects, classes, communication through method calls etc.) are represented visually in its interaction design in the interface.**

**BlueJ was started in 1999 as a successor to the Blue system.**

**BlueJ is currently being maintained at the University of Kent, Canterbury in England.**

**JAVA was originally developed by James Gosling at Sun Microsystems which has since been acquired by Oracle Corporation.**

**BlueJ has several features not seen before in other IDE’s. It has a deliberately smaller and simpler interface and has been designed with good pedagogy in mind. INTRODUCTION TO TOPIC**

**Paradise Food Court is one among the finest and best restaurants in Hyderabad, Telangana which is known far and wide for its aromatic and delectable Biryani.**

**Beginning as a modest café in 1953, Paradise has grown into one of the most-admired restaurant chains in the country today and has maintained its status of being an iconic place that serves the best Biryani. Its award-winning cuisine is loved for its purity and taste.**

**Started by Mr.G.Hussain, the Paradise café served Irani chai, coffee and snacks which became instantly popular. Many developments took place in the Paradise Franchise from then onwards.**

**The Paradise Group embraced the use of technology in all its operations.**

**Order management, fulfilment and billing have been computerized in 2006.**

**Since then Paradise is still working in its benevolent manner and spreading the joy of food everywhere.**

**I have included Non-Vegetarian, Vegetarian, Desserts and Beverage items and have made four different menus for each category in my program.**

**I have allowed the user to view each menu, place an order in each category individually and when he desires to exit, I have displayed the final bill including all purchased items.**

**VARIABLE DESCRIPTION**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.NO.** | **VARIABLE** | **DATATYPE** | **DESCRIPTION** |
| 1. | nitem[] | String | To store the non-veg items |
| 2. | vitem[] | String | To store the veg items |
| 3. | ditem[] | String | To store the dessert items |
| 4. | bitem[] | String | To store the beverage items |
| 5. | nprice[] | int | To store prices of non-veg items |
| 6. | vprice[] | int | To store prices of veg items |
| 7. | dprice[] | int | To store prices of dessert items |
| 8. | bprice[] | int | To store prices of beverages |
| 9. | n[] | String | To store the name of the chosen item |
| 10. | no[] | int | To store the item number of the chosen item |
| 11. | p[] | double | To store price of the chosen item |
| 12. | q[] | double | To store quantity of the chosen item |
| 13. | t[] | double | To store total price of the chosen item |
| 14. | ch | int | To accept the number of the category for menu display |
| 15. | i | int | Loop variable to print items of each category with price in serial order |
| 16. | ch1 | String | To check whether user wants to continue menu display or not |
| 17. | c1 | String | To accept user’s choice to place an order in any category |
| 18. | ino | int | Stores the number of the item chosen by the user |
| 19. | qty | int | Stores the quantity of the chosen item |
| 20. | total | double | Calculates and stores the total price of the chosen item |
| 21. | tax | double | To calculate and store 15 percent tax on the total amount |
| 22. | netamt | double | To store the net amount after adding the tax to the total amount |
| 23. | c2 | String | To accept user’s choice and check whether he wants to continue ordering in the category |
| 24.. | pos | int | To increment the index position in the various arrays and to store the number of items brought by the user |
| 25. | ch2 | int | To accept choice of user to view details, place an order or to exit |
| 26. | k | int | Loop variable to print final bill of each item in serial order |

**METHOD DESCRIPTION**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.NO** | **FUNCTION NAME** | **RETURN TYPE** | **DESCRIPTION** |
| 1. | view  **NO PARAMETERS USED** | void | Accepts user’s choice for menu display and displays  non-vegetarian, vegetarian, desserts or beverage items in the menu according to the user’s choice. |
| 2. | order  **NO PARAMETERS USED** | void | Accepts user’s choice for the category of dishes from which he would like to order and asks him to enter the item number and quantity of the dish. It allows him to continue ordering as long as he desires. |
| 3. | menu  **NO PARAMETERS USED** | void | Accepts user’s choice and checks whether he wants to view the menu details, place an order or view his bill and exit. |

**PROGRAM**

import java.io.\*;

class Paradise

{

InputStreamReader ir=new InputStreamReader(System.in);

BufferedReader br= new BufferedReader(ir);

String nitem[]={"Chicken Soup", "Chicken 65", "Pepper Chicken", "Chicken Sticks", "Chilli Prawns", "Chicken Biryani", "Mutton Biryani", "Butter Chicken", "Mutton Masala", "Butter Naan"};

String vitem[]={"Veg Corn Soup", "Gobi Manchurian", "Spring Rolls", "Paneer 65", "Corn Manchurian", "Veg Biryani", "Veg Noodles", "Palak Paneer", "Malai Kofta", "Paneer Kulcha"};

String ditem[]={"Caramel Custard", "Gulab Jamun", "Choco Brownie", "Ice Cream", "Gajar Ka Halwa"};

String bitem[]={"Cream Lassi", "Cold Coffee", "Fruit Juice", "Milk Shake", "Coca Cola", "Kinley Water"};

int nprice [] ={134, 309, 309, 309, 378, 244, 254, 288, 309, 59};

int vprice [] ={125, 221, 175, 229, 229, 200, 184, 210, 210, 75};

int dprice []={84,84,100,150,84};

int bprice []={84,84,84,84,42,40};

String n[]=new String[10];

int no[]=new int[10];

double p[]=new double[10];

double q[]=new double[10];

double t[]=new double[10];

String ch1,c1,c2;

int ch, ino, qty, pos=0,ch2;

double total=0,tax,netamt;

void view()throws IOException

{

do /\*allows the user to continue menu display\*/

{

System.out.println("1: Non Veg");

System.out.println("2: Veg");

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

System.out.println("4: Beverages");

System.out.println("Enter choice for menu display:");

ch=Integer.parseInt(br.readLine());

switch(ch) {

case 1: /\* prints the non-vegetarian menu \*/

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"); System.out.println("No.\tItem\t\t\tPrice"); System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

for(int i=0;i<nitem.length;i++) System.out.println((i+1)+"\t"+nitem[i]+"\t\t"+ nprice[i]); System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

break;

case 2:/\* prints vegetarian menu \*/ System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"); System.out.println("No.\tItem\t\t\tPrice"); System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

for(int i=0;i<vitem.length;i++) System.out.println((i+1)+"\t"+vitem[i]+"\t\t"+ vprice[i]); System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

break;

case 3: /\* prints the desserts menu \*/ System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"); System.out.println("No.\tItem\t\t\tPrice"); System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

for(int i=0;i<ditem.length;i++) System.out.println((i+1)+"\t"+ditem[i]+"\t\t"+ dprice[i]); System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

break;

case 4: /\* prints the beverages menu \*/ System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"); System.out.println("No.\tItem\t\t\tPrice"); System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

for(int i=0;i<bitem.length;i++) System.out.println((i+1)+"\t"+bitem[i]+"\t\t"+ bprice[i]); System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

break;

default:

System.out.println("Wrong Choice Entered");

}

System.out.println("Enter y For Continuing Menu Display:");

ch1=br.readLine();

}while(ch1.equalsIgnoreCase("y"));

}

void order()throws IOException

{

do /\* checks if user has entered a wrong choice\*/

{

System.out.println("Enter 'N' for Non Veg, 'V' for Veg, 'D' for Desserts or 'B' for Beverages :");

c1=br.readLine(); if((c1.equals("N")||c1.equals("V")||

c1.equals("D")||c1.equals("B"))==false)

System.out.println("Error!! You Have To Enter 'N', 'V', 'D' or 'B'"); }while((c1.equals("N")||c1.equals("V")|| c1.equals("D")||c1.equals("B"))==false);

switch(c1)

{

case "N":/\* displays menu and allows user to order non-veg\*/ System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"); System.out.println("No.\tItem\t\t\tPrice"); System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

for(int i=0;i<nitem.length;i++) System.out.println((i+1)+"\t"+nitem[i]+"\t\t"+ nprice[i]); System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

do/\* allows user to continue ordering\*/

{

do{ /\*checks if item number is correct\*/

System.out.println("Enter Item Number:");

ino=Integer.parseInt(br.readLine()); if(ino<1||ino>nitem.length)

System.out.println("Error In Item Number");

}while(ino<1||ino>nitem.length);

System.out.println("Price is:"+nprice[ino-1]);

System.out.println("Enter Quantity:"); qty=Integer.parseInt(br.readLine());

total=total+qty\*nprice[ino-1];

n[pos]=nitem[ino-1];

no[pos]=ino;

p[pos]=nprice[ino-1];

q[pos]=qty;

t[pos]=qty\*p[pos];

System.out.println("Enter y To Buy More:");

c2=br.readLine();

pos++;

}while(c2.equalsIgnoreCase("y"));

menu();/\*calls the function to check what the user’s wish is\*/

break;

case "V":/\* displays menu and allows user to order veg\*/ System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"); System.out.println("No.\tItem\t\t\tPrice"); System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

for(int i=0;i<vitem.length;i++) System.out.println((i+1)+"\t"+vitem[i]+"\t\t"+ vprice[i]); System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

do

{

do

{

System.out.println("Enter Item Number:"); ino=Integer.parseInt(br.readLine());

if(ino<1||ino>vitem.length)

System.out.println("Error In Item Number");

}while(ino<1||ino>vitem.length);

System.out.println("Price is:"+vprice[ino-1]);

System.out.println("Enter Quantity:"); qty=Integer.parseInt(br.readLine());

total=total+qty\*vprice[ino-1];

n[pos]=vitem[ino-1];

no[pos]=ino;

p[pos]=vprice[ino-1];

q[pos]=qty;

t[pos]=qty\*p[pos];

System.out.println("Enter y To Buy More:");

c2=br.readLine();

pos++;

}while(c2.equalsIgnoreCase("y"));

menu();

break;

case "D":/\*displays menu and allows user to order desserts\*/ System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"); System.out.println("No.\tItem\t\t\tPrice"); System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

for(int i=0;i<ditem.length;i++) System.out.println((i+1)+"\t"+ditem[i]+"\t\t"+ dprice[i]); System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

do

{

do

{

System.out.println("Enter Item Number:"); ino=Integer.parseInt(br.readLine());

if(ino<1||ino>ditem.length)

System.out.println("Error In Item Number");

}while(ino<1||ino>ditem.length);

System.out.println("Price is:"+dprice[ino-1]);

System.out.println("Enter Quantity:"); qty=Integer.parseInt(br.readLine());

total=total+qty\*dprice[ino-1]; n[pos]=ditem[ino-1];

no[pos]=ino;

p[pos]=dprice[ino-1];

q[pos]=qty;

t[pos]=qty\*p[pos];

System.out.println("Enter y To Buy More:");

c2=br.readLine();

pos++;

}while(c2.equalsIgnoreCase("y"));

menu();

break;

case "B":/\* displays menu and allows user to order beverages\*/ System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"); System.out.println("No.\tItem\t\t\tPrice"); System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

for(int i=0;i<bitem.length;i++) System.out.println((i+1)+"\t"+bitem[i]+"\t\t"+ bprice[i]); System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

do

{

do

{

System.out.println("Enter Item Number:"); ino=Integer.parseInt(br.readLine());

if(ino<1||ino>bitem.length)

System.out.println("Error In Item Number");

}while(ino<1||ino>bitem.length);

System.out.println("Price is:"+bprice[ino-1]);

System.out.println("Enter Quantity:"); qty=Integer.parseInt(br.readLine());

total=total+qty\*bprice[ino-1];

n[pos]=bitem[ino-1];

no[pos]=ino;

p[pos]=bprice[ino-1];

q[pos]=qty;

t[pos]=qty\*p[pos];

System.out.println("Enter y To Buy More:");

c2=br.readLine();

pos++;

}while(c2.equalsIgnoreCase("y"));

menu();

break;

}

}

void menu()throws IOException

{

do/\* checks if choice entered is correct\*/

{

System.out.println("1: View Details");

System.out.println("2: Place An Order");

System.out.println("3: View Bill And Exit");

System.out.println("Enter choice from 1-3:");

ch2=Integer.parseInt(br.readLine());

if(ch2<1||ch2>3)

System.out.println("Error In Choice. It Should Be From 1-3 Only");

}while(ch2<1||ch2>3);

switch(ch2)

{

case 1:/\*displays details\*/

view();

menu();

break;

case 2:/\*allows user to place an order\*/

order();

break;

case 3:/\*prints final bill and exits\*/

System.out.println("Total Items:"+pos); System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"); System.out.println("S.no\tCode\tName\t\t\t Price\t\tQuantity\tTotal"); System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

for(int k=0;k<pos; k++) System.out.println((k+1)+"\t"+no[k]+"\t"+n[k]+ "\t\t"+p[k]+"\t\t"+q[k]+"\t\t"+t[k]); System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.println("Bill Amount:"+total);

tax=0.15\*total;

netamt=total+tax;

System.out.println("Tax:"+tax);

System.out.println("Net Amount:"+netamt);

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"); System.out.println("Paradise Has Been Enriched By Your Visit"); System.out.println("Do Give Your Feedback At The Reception And Help Us To Enhance Our Work");

System.out.println("Thanks For Visiting Us And Come Back Soon!!");

break;

}

}

}

**OUTPUT SCREEN**

**1: View Details**

**2: Place An Order**

**3: View Bill And Exit**

**Enter choice from 1-3:**

**2**

**Enter 'N' for Non Veg, 'V' for Veg, 'D' for Desserts or 'B' for Beverages :**

**V**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**No. Item Price**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**1 Veg Corn Soup 125**

**2 Gobi Manchurian 221**

**3 Spring Rolls 175**

**4 Paneer 65 229**

**5 Corn Manchurian 229**

**6 Veg Biryani 200**

**7 Veg Noodles 184**

**8 Palak Paneer 210**

**9 Malai Kofta 210**

**10 Paneer Kulcha 75**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Enter Item Number:**

**3**

**Price is:175**

**Enter Quantity:**

**6**

**Enter y To Buy More:**

**y**

**Enter Item Number:**

**6**

**Price is:200**

**Enter Quantity:**

**2**

**Enter y To Buy More:**

**n**

**1: View Details**

**2: Place An Order**

**3: View Bill And Exit**

**Enter choice from 1-3:2**

**Enter 'N' for Non Veg, 'V' for Veg, 'D' for Desserts or 'B' for Beverages :**

**D**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**No. Item Price**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**1 Caramel Custard 84**

**2 Gulab Jamun 84**

**3 Choco Brownie 100**

**4 Ice Cream 150**

**5 Gajar Ka Halwa 84**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Enter Item Number:2**

**Price is:84**

**Enter Quantity:**

**4**

**Enter y To Buy More:**

**y**

**Enter Item Number:**

**4**

**Price is:150**

**Enter Quantity:**

**2**

**Enter y To Buy More:**

**n**

**1: View Details**

**2: Place An Order**

**3: View Bill And Exit**

**Enter choice from 1-3:3**

**Total Items: 4**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*s.no code name price quantity total**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*1 3 Spring Rolls 175.0 6.0 1050.0**

**2 6 Veg Biryani 200.0 2.0 400.0**

**3 2 Gulab Jamun 84.0 4.0 336.0**

**4 4 Ice Cream 150.0 2.0 300.0**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Bill Amount:2086.0**

**Tax:312.0**

**Net Amount:2398.0**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Paradise Has Been Enriched By Your Visit**

**Do Give Your Feedback At The Reception And Help Us To Enhance Our Work**

**Thanks For Visiting Us And Come Back Soon!!**

**SCOPE OF EXPANSION**

**Though this program has turned out well and has also made me proud, it still has some drawbacks which I would like to list down.**

* **I could not include many menu options when compared to what exactly is available in Paradise.**
* **The process of ordering is tedious and time consuming for the user.**
* **I have not been able to include the feature through which the user can cancel the order which he has already made.**

**CONCLUSION**

**This project not only helped me in learning how to do good programming, but it also made me develop some skills which will certainly help me in the future.**

**It made me take spontaneous and right decisions in choosing the topic which would suit me best.**

**It also inculcated the capacity of logical reasoning in me and improved my thought process.**

**It also brought a sense of enjoyment which increased my passion and love for JAVA and for Computer Applications.**

**This project has definitely strengthened my skills in Arrays and decision-making programs and has made me more comfortable and knowledgeable in the use of switch case.**

**Hence, this project has helped me gain proficiency in many aspects which will be a long-lasting impression on my life.**

**BIBLIOGRAPHY**

**I have referred to the following websites to collect and organise the information in my project:**

* [**www.hyatt.com**](http://www.hyatt.com)
* [**www.wikipedia.com**](http://www.wikipedia.com)
* [**www.bluej.org**](http://www.bluej.org)
* [**www.wideskills.com**](http://www.wideskills.com)

**I have also been to the Paradise Restaurant in order to carry out my research.**