

**Karachi Institute of Economics and Technology College of Computing and Information Sciences**

OBJECT ORIENTED PROGRAMMING-111728

**Assignment - 1**

**IMPORTANT INSTRUCTIONS**:

**Read the following Instructions carefully:**

* Attempt all three-question using Windows Form.
* After attempting all question, add screenshot of each question’s solution with output.
* Must attempt all questions by yourself, in case of copied solutions your assignment will marked ZERO.
* You may provide answers HAND WRITTEN (if needed). The scanned solution must be submitted in PDF file format (Use any suitable Mobile Application for Scanning)
* Arrange questions and their subsequent parts in sequence.
* Submit on LMS before deadline.

# Save your word document with name: [Student Name Student ID] Submission Deadline: 15/04/2023

**[5 mark]**

## Students who’s roll no's ends with **even** number have to solve using decision structure(if-else) and for those who’s roll no's ends with **odd** number solve by using switch case.

A developer is being asked to create a meal planner tool. Users enter the appetizer, main course, and dessert into the program and it will print out a summary of the information. Rewrite the following menu-based program along with test class to show the desired output.

## **Usage Example**

**Welcome to the PAF-KIET Canteen Meal Selector**

Please select from the following

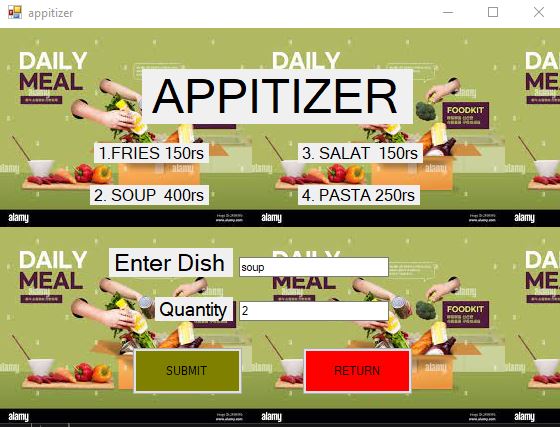
1 –Add appetizer

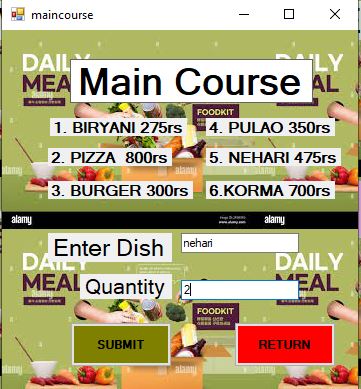
2 - Add main course

3 - Add dessert

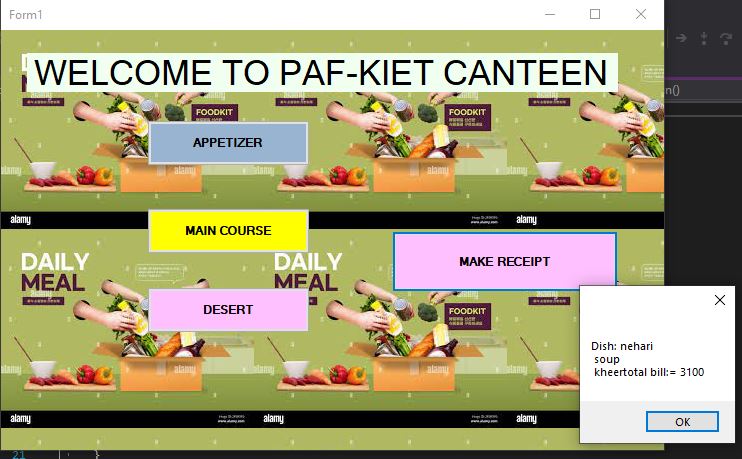
4 –Summary of items

5 – Exit









**APPITIZER**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace kiet\_canteen

{

public partial class appitizer : Form

{

public static string apptizerdish;

public static int apptizerquantity;

public static int total;

public appitizer()

{

InitializeComponent();

}

private void bgr\_CheckedChanged(object sender, EventArgs e)

{

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

}

private void submitbtn\_Click(object sender, EventArgs e)

{

apptizerdish = dishtxt.Text;

apptizerquantity = Convert.ToInt32(quantitytxt.Text);

switch (apptizerdish)

{

case "FRIES":

total = apptizerquantity \* 150;

break;

case "soup":

total = apptizerquantity \* 400;

break;

case "salat":

total = apptizerquantity \* 150;

break;

case "pasta":

total = apptizerquantity \* 250;

break;

case "fries":

total = apptizerquantity \* 150;

break;

case "SOUP":

total = apptizerquantity \* 400;

break;

case "SALAT":

total = apptizerquantity \* 150;

break;

case "PASTA":

total = apptizerquantity \* 250;

break;

}

this.Close();

}

private void Returnbtn\_Click(object sender, EventArgs e)

{

this.Close();

}

}

}

**MAIN COURSE**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace kiet\_canteen

{

public partial class maincourse : Form

{

public static string maincoursedish;

public static int total;

public static int mainquan;

public maincourse()

{

InitializeComponent();

}

private void submitbtn\_Click(object sender, EventArgs e)

{

maincoursedish = maindish.Text;

mainquan = Convert.ToInt32(mainquantity.Text);

switch (maincoursedish)

{

case "biryani":

total = mainquan \* 275;

break;

case "pulao":

total = mainquan \* 350;

break;

case "pizza":

total = mainquan \* 800;

break;

case "nehari":

total = mainquan \* 475;

break;

case "burger":

total = mainquan \* 300;

break;

case "korma":

total = mainquan \* 700;

break;

case "BIRYANI":

total = mainquan \* 275;

break;

case "PULAO":

total = mainquan \* 350;

break;

case "PIZZA":

total = mainquan \* 800;

break;

case "NEHARI":

total = mainquan \* 475;

break;

case "BURGER":

total = mainquan \* 300;

break;

case "KORMA":

total = mainquan \* 700;

break;

}

this.Close();

}

private void returnbtn\_Click(object sender, EventArgs e)

{

this.Close();

}

}

}

**DESSERT**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace kiet\_canteen

{

public partial class desert : Form

{

public static string desertd;

public static int desertquan, deserttotal;

public desert()

{

InitializeComponent();

}

private void returnbtn\_Click(object sender, EventArgs e)

{

this.Close();

}

private void submitbtn\_Click(object sender, EventArgs e)

{

desertd = desertdish.Text;

desertquan = Convert.ToInt32(desertquantity.Text);

switch (desertd)

{

case "ice cream":

deserttotal = desertquan \* 250;

break;

case "falooda":

deserttotal = desertquan \* 275;

break;

case "kheer":

deserttotal = desertquan \* 450;

break;

case "ICE CREAM":

deserttotal = desertquan \* 250;

break;

case "FALOODA":

deserttotal = desertquan \* 275;

break;

case "KHEER":

deserttotal = desertquan \* 450;

break;

}

this.Close();

}

}

}

**MAKE RECIPT**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace kiet\_canteen

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void apptizerbtn\_Click(object sender, EventArgs e)

{

appitizer apitizerobj = new appitizer();

apitizerobj.Show();

}

private void makereceiptbtn\_Click(object sender, EventArgs e)

{

int aptquan = appitizer.apptizerquantity;

string apptdish = appitizer.apptizerdish;

int appttotal = appitizer.total;

int mainquan = maincourse.mainquan;

string maindish = maincourse.maincoursedish;

int totalmain = maincourse.total;

string desertdish = desert.desertd;

int deserttotal = desert.deserttotal;

int overalltotal = deserttotal + totalmain + appttotal;

MessageBox.Show("Dish: " + maindish + "\n " + apptdish + "\n " + desertdish + "total bill:= " + overalltotal.ToString());

}

private void maincoursebtn\_Click(object sender, EventArgs e)

{

maincourse maincour = new maincourse();

maincour.Show();

}

private void desertbtn\_Click(object sender, EventArgs e)

{

desert desert1 = new desert();

desert1.Show();

}

}

}

## Note: Every individual thinks differently, so make sure your work is not copied or shared with anyone. Else you lose marks.

**Consider the following situation.** **[3 mark]**

“The Head of Department of XYZ university allocates duties to members of teaching staff and others at the end of each academic year. One person (Teacher) is assigned to lecture each of the modules which are supposed to be available in the following year. Each teacher updates his/her course outline for the module assigned. The Course coordinator updates other parts of course outline and check the module entries produced by the teachers. The academic officer keeps the master list of all students and updates the mailing list of students taking different modules and send it to the course coordinator. Student provisionally registers for modules. The academic officer checks that every student registered for a reasonable set of modules.”

1. Predict the potential classes under the scope of this project? Also explain the reason of

assumption.

1. Show the purpose and scope of the project as per your own understanding?
2. Predict attributes and methods that are essential in the classes explain the reason?
3. Illustrate an appropriate class hierarchy.

a) Based on the given description, potential classes that can be identified are:

HeadOfDepartment: This class represents the head of the department who assigns duties to the teaching staff.

Teacher: This class represents the teaching staff who are assigned to lecture on specific modules.

Module: This class represents each module that is offered in the following year.

CourseCoordinator: This class represents the coordinator who updates the course outline and checks the module entries produced by the teachers.

AcademicOfficer: This class represents the officer who keeps the master list of all students, updates the mailing list of students taking different modules, and checks that every student registered for a reasonable set of modules.

Student: This class represents the students who provisionally register for modules.

b) The purpose of the project is to facilitate the allocation of duties to teaching staff and ensure that the course outline for each module is updated and checked. The scope of the project includes assigning teachers to specific modules, updating the course outline, maintaining a master list of students and their registered modules, and ensuring that each student is registered for a reasonable set of modules.

c) The following attributes and methods can be essential in the identified classes:

HeadOfDepartment:

Attributes: name, department, duties\_assigned

Methods: assign\_duties(), get\_assigned\_duties()

Teacher:

Attributes: name, module\_assigned, course\_outline

Methods: update\_course\_outline(), get\_module\_assigned()

Module:

Attributes: code, name, description, teacher\_assigned

Methods: get\_code(), get\_name(), get\_description(), get\_teacher\_assigned()

CourseCoordinator:

Attributes: name, course\_outline, module\_entries

Methods: update\_course\_outline(), check\_module\_entries(), get\_course\_outline()

AcademicOfficer:

Attributes: name, master\_list, mailing\_list

Methods: update\_master\_list(), update\_mailing\_list(), check\_registered\_modules()

Student:

Attributes: name, student\_id, registered\_modules

Methods: register\_modules(), get\_registered\_modules()

d) An appropriate class hierarchy can be as follows:

Person (parent class)

HeadOfDepartment (child class)

Teacher (child class)

CourseCoordinator (child class)

AcademicOfficer (child class)

Student (child class)

Module (separate class)

**[2 mark]**

**Consider the scenario**

of a "Bookstore" which is described by its bookId, author, title, price, Publishing Year and quantity Available.

Include a method called purchase() in class "Book" taking your roll number as the purchase quantity as a parameter and update the available quantity appropriately.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace question\_3

{

class book

{

int bookid = 11152, price = 449, publishing\_year = 2023, quantityAvailable = 100000;

string author="sohaib", title="student life";

public void Purchase(int quantity)

{

if (quantity <= quantityAvailable)

{

int remaining\_quantity=quantityAvailable- quantity;

Console.WriteLine("BOOK ID:"+bookid+" \n "+"price: "+price+" \n"+" author: "+author+" \n "+"TITTLE: "+title+" publishing year: "+publishing\_year+" quantity availabe: "+quantityAvailable+" QUANTITY PURCHASED: "+quantity+" REMAINING QUANTITY: "+remaining\_quantity);

}

else

{

Console.WriteLine("Not enough copies available for purchase.");

Console.WriteLine("{0} Quantity of book remaining", quantityAvailable);

}

}

}

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Enter your id number as your book quantity");

int pur = Convert.ToInt32(Console.ReadLine());

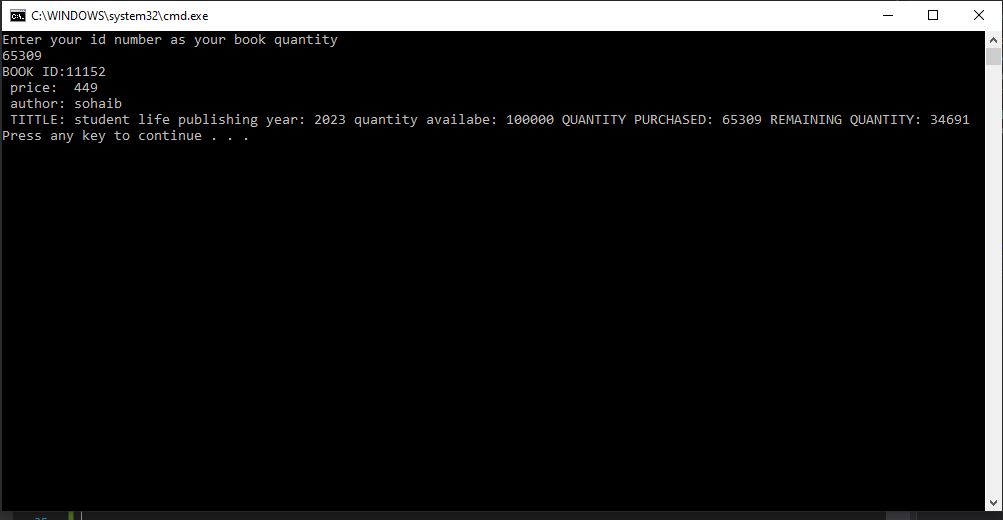
book bok = new book();

bok.Purchase(pur);

}

}

}

****

**Example to Question # 1**

To make a meal planner tool, we have to create a Win Form first. The Win Form we create will be named Canteen.

\*Below is the image of the Main Display Form.

