



Department of Software Technology

ETCOTH Programming Project Specifications

Deadline: April 11, 2023 (Tuesday) 7:30AM via AnimoSpace

Restaurants and fast-food chains have been using ordering systems for handling orders. These kinds of systems not only takes inputs of the orders but also computes the amount due for the order. This project will have you develop such a system.

The following tables will show the food items available and their prices.

Mains:

ID	Type	Price
1	Chicken	90.00
2	Pork	105.00
3	Fish	120.00
4	Beef	135.00

Table 1: List of main dishes and their corresponding prices

Sides:

ID	Type	Price
1	Steamed Rice	20.00
2	Shredded Corn	35.00
3	Mashed Potatoes	50.00
4	Steam Vegetables	65.00

Table 2: List of side dishes and their corresponding prices

Drinks:

ID	Type	Price
1	Mineral Water	25.00
2	Iced Tea	35.00
3	Soda	45.00
4	Fruit Juice	55.00

Table 3: List of drink and their corresponding prices

Order – An order is the process where the user will input 1 main, 1 side and 1 drink.

Features:

When the program starts up, the user will be given 2 choices:

- Order
- Exit

If the user selects **Exit**, the program will terminate/end.

If the user selects **Order**, the program will proceed to the ordering feature.

Group Ordering:

A group can make three(3) orders or less. They must first specify how many they are in the group before they order. They are allowed to only order specific items from either the main, sides or drinks(eg. Only order a main and nothing else).

Example:

```
How many people are in your group: 3
```

```
Order 1:
```

```
    Main:      3
           Fish
    Side:      0
           None
    Drink:      0
           None
```

```
Is this order correct(y/n)? y
```

```
Proceed with the next order(y/n)? y
```

```
Order 2:
```

```
    Main:      0
           None
    Side:      3
           Mashed Potatoes
    Drink:      4
           Fruit Juice
```

```
Is this order correct(y/n)? y
```

```
Proceed with the next order(y/n)? y
```

```
Order 3:
```

```
    Main:      1
           Chicken
    Side:      1
           Steamed Rice
    Drink:      0
           None
```

Is this order correct(y/n)? y

Cancel all orders(y/n)? n

Exclude an item from the total(y/n)? n

Each type is limited to 1 piece. (eg. 2 mains in 1 order is not allowed).

Cancelling:

The system will allow the user to cancel an order.

- The system will have the option to let the user cancel an order at the start of the next order after the first order has been completed. If cancelled is selected, the system will then proceed to checkout (the Amount Due part).

Example: The group initially decided to make 2 orders but then decides that they only want to make order 1 instead. After selecting the items for the first order(main, side and drink), the user may select the cancel option for the second order.

- The system will also have the option to let the user cancel an order after all orders have finished. The system will then go back to the Ordering part.

Example:

How many people are in your group: 2

Order 1:

Main: 1
Chicken
Side: 2
Shredded Corn
Drink: 3
Soda

Is this meal set order correct(y/n)? y

Proceed with the next order(y/n)? n

If 'n' is selected, the system will then proceed to the checkout and display the total amount due.

Modifying/Changing Order:

The system will allow a specific order to be modified. This will happen only after finishing inputting each item for that order. Specifically, after the input for the drink. The system will prompt the user to confirm if the order is correct. If the user selects no, then the system will repeat the process for that order.

Example:

How many people are in your group: 3

Order 1:

Main: 1
Chicken
Side: 2
Shredded Corn
Drink: 3
Soda

Is this meal set order correct(y/n)? n

Order 1:

Main: 3
Fish
Side: 4
Steamed Vegetables
Drink: 1
Mineral Water

Is this meal set order correct(y/n)? y

Proceed with the next order(y/n)? y

Order 2:

Main: ...

Total Amount Due:

The system will be able to display the subtotals and total amount that the group must pay for their order. It will also display how much each person in the group must pay.

Example:

Order for party of 3

Order 1:

Main:	Fish	P120.00	
Side:	Shredded Corn	P35.00	
Drink:	Soda	P45.00	
Subtotal:			P200.00

Order 2:

Main:	Pork	P105.00	
Side:	Mashed Potatoes	P50.00	
Drink:	None	P0.00	
Subtotal:			P155.00

Total Amount Due:	P355.00
-------------------	---------

Each person must pay:	P118.33
-----------------------	---------

Excluding items from the total:

Sometimes when we order as a group, we tend to add orders that we declare will not be included in the final amount that we will pay separately. This will happen after completing all orders before the amount due will be displayed.

Example:

Exclude an item from the total(y/n)? y
From which order? 1
Which item will be excluded? 3
Soda will be excluded from the total.

Order for party of 3

Order 1:

Main:	Fish	P120.00
Side:	Shredded Corn	P35.00
Drink:	Soda	P0.00

Subtotal: P155.00

Order 2:

Main:	Pork	P105.00
Side:	Mashed Potatoes	P50.00
Drink:	None	P0.00

Subtotal: P155.00

Total Amount Due: P310.00

Each person must pay: P103.33

You will notice in the example the the price for soda is now P0.00 since it was excluded.

How to Approach the Machine Project

Step 1: Problem analysis and algorithm formulation

Read the MP Specifications again! Identify clearly what are the required information from the user, what kind of processes are needed, and what will be the output (s) of your program. Clarify with your professor any issues that you might have regarding the machine project.

When you have all the necessary information, identify the necessary functions that you will need to modularize the project. Identify the required data of these functions and what kind of data they will return to the caller. Write your algorithm for each of these modules/functions as well as the algorithm for your main program.

Step 2: Implementation

In this step, you are to translate your algorithm into proper Python statements. While implementing, you are to perform the other phases of program planning and design (discussed in the other steps below) together with this step.

It is best that you perform your coding “incrementally.” This means:

- Dividing the program specification into subproblems, and solving each problem separately according to your algorithm;
- Code the solutions to the subproblems one at a time. Once you’re done coding the solution for one subproblem, apply testing and debugging.

Documentation

While coding, you have to include internal documentation in your programs. You are expected to have the following:

- File comments or Introductory comments
- Function comments
- In-line comments

Introductory comments are found at the very beginning of your program before the preprocessor directives. Follow the format shown below. Note that items in between < > should be replaced with the proper information.

```
#Description:      <Describe what this program does briefly>
#Programmed by:   <your name here>   <section>
#Last modified:   <date when last revision was made>
#Version:        <version number>
#Acknowledgements: <list of sites or borrowed libraries and sources>
```

STEP 3: TESTING AND DEBUGGING

For each feature of your program, you have to fully test it before moving to the next feature. Sample questions that you should ask yourself are:

1. What should be displayed on the screen if the user inputs an order?
2. What would happen if I input incorrect inputs? (e.g., values not within the range)
3. Is my program displaying the correct output?
4. Is my program following the correct sequence of events (correct program flow)?
5. Is my program terminating (ending/exiting) correctly? Does it exit when I press the command to quit? Does it exit when the program's goal has been met? Is there an infinite loop?
6. and others...

IMPORTANT POINTS TO REMEMBER:

1. You are required to implement the project using the Python language.
2. The implementation will require you to:
 - Appropriately use conditional statements, loops and other constructs discussed in class.
 - Consistently employ coding conventions
 - Include internal documentation (i.e., comments)
3. Deadline for the project is the **7:30AM of April 11, 2023 (Tuesday)** via submission through **AnimoSpace**. After this time, submissions will be considered late and will have corresponding deductions depending on how late it was submitted.
4. The following are the deliverables:

Checklist:

- ☐ Upload in AnimoSpace by clicking **Submit Assignment** on Programming Project and adding the following files:
 - ☐ source code* (python file)
- ☐ email the softcopies of everything as attachments to **YOUR own email address** on or before the deadline

Legend:

*Source Code also includes the internal documentation. The **first few lines of the source code** should have the following declaration (in comment) **BEFORE** the introductory comment:

```
# This is to certify that this project is my own work, based on my personal
# efforts in studying and applying the concepts learned. I have constructed
# the functions and their respective algorithms and corresponding code by
# myself. The program was run, tested, and debugged by my own efforts. I
# further certify that I have not copied in part or whole or otherwise
# plagiarized the work of other students and/or persons.
# <your full name>, DLSU ID# <number>
# <your full name>, DLSU ID# <number>
```

5. Any requirement not fully implemented and instruction not followed will merit deductions.
6. This is an **individual/pair project**. Working in collaboration with other groups, asking other group's help, and/or copying other people's work are considered as cheating. Cheating is

punishable by a failing grade for course, aside from which, a cheating case may be filed with the Discipline Office.

Note that any additional feature not stated here may be added but **should not conflict with whatever instruction was given in the project specifications**. Bonus points are given upon the discretion of the teacher, based on the difficulty and applicability of the feature to the program. Note that **bonus points can only be credited if all the basic requirements are fully met** (i.e., complete and no bugs).

HONESTY POLICY AND INTELLECTUAL PROPERTY RIGHTS

Honesty policy applies. Please take note that you are NOT allowed to borrow and/or copy-and-paste – in full or in part any existing related program code from the internet or other sources (such as printed materials like books, or source codes by other people that are not online). **You should develop your own codes from scratch by yourself.**