**UNIVERSITI MALAYSIA SARAWAK**

**Faculty of Computer Science and Information Technology**

**Assignment**

|  |  |  |
| --- | --- | --- |
| Assignment  Number of Title | Assignment 2 | |
| Subject Code | TMF 1014 | Subject Name: System Analysis and Design |

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Matric No** | **Phone No** | **E-mail** |
| Christopher Sii How Chiong | 69385 | 0196386769 | [69385@siswa.unimas.my](mailto:69385@siswa.unimas.my) |
|  |  |  | [@siswa.unimas.my](mailto:69847@siswa.unimas.my) |
|  |  |  | [@siswa.unimas.my](mailto:70086@siswa.unimas.my) |
|  |  |  | [@siswa.unimas.my](mailto:72713@siswa.unimas.my) |
| Joash Paul | 69990 | 0163339721 | [69990@siswa.unimas.my](mailto:69990@siswa.unimas.my) |
|  |  |  |  |

|  |  |
| --- | --- |
| Name of Lecturer : Dr. Fatihah binti Ramli | |
| Due Date : 6 December 2019 | Date received and approved (for office use only) |

|  |
| --- |
| ***Plagiarism and Collusion are methods of cheating that falls under Peraturan Akademik Universiti Malaysia Sarawak para 11: Etika Akademik***    **Plagiarism**  Plagiarism is the presentation of work which has been copied in whole or in part from another person’s work, or from any other source such as the internet, published books or periodicals without due acknowledgement given in the text.    **Collusion**  Collusion is the presentation of work that is the result in whole or in part of unauthorized collaboration with another person or persons.    Where there are reasonable grounds for believing that cheating has occurred, the only action that may be taken when plagiarism or collusion is detected is for the staff member not to mark the item of work and to report or refer the matter to the Dean. This may result in work being disallowed and given a fail grade or if the circumstances warrant, the matter may be referred to a Committee of inquiry for investigation. Such investigation may result in the matter being referred to the University Discipline Committee, which has the power to exclude a student. |
| **Note:** Assignments are returned after being assessed by lecturer and to be collected from the lecturer’s office. Students should endeavour to collect their assignments prior to commencement of next semester. All unclaimed assignments become the property of the faculty and will become subject to destruction. |

|  |
| --- |
| **Student’s Statement :**  I certify that I have not plagiarized the work of others or participated in unauthorized collusion when preparing this assignment. I also certify that I have taken proper case in safeguarding my work and have made all reasonable efforts to ensure that may work not be able to be copied.  Signature **: ………………………………………….**  Signature **: ………………………………………….**  Signature **: ………………………………………….**  Signature **: ………………………………………….**  Signature **: ………………………………………….**  Signature **: ………………………………………….** |

|  |  |
| --- | --- |
| **MARK :** | Comments: |

This cover sheet must be completed, signed and firmly attached to the front of the submission.

All work must be submitted by the due date. If an extension of work is granted, an assignment extension acknowledge slip must be signed by lecturer/ tutor and attached to the assignment.

Please note that it is the student’s responsibility to retain a copy of his/her own assignment.

**PSEUDOCODE**

**Start**

Declarations

Int option;

Struct Customer Cust;

InputFile fw;

InputFile fa;

InputFile ft;

Open fa “Receipt.txt”

Close fa

While (option != 0) do

Display function display\_title

Display function get\_nasiAyam(cust)

Display function get\_type(cust)

Display function get\_addon(cust)

Display function get\_extra\_addon(cust)

Display function receipt (cust, fw)

Display function get\_extra\_order2(cust)

Display function get\_extra\_order3(cust)

Display function get\_extra\_order4(cust)

Display function total\_price (cust, ft)

Display "Enter 0 to (EXIT), Else for the next customer "

Open ft “Receipt.txt”

Input cust[i].sum\_price;

Close ft

**END**

Struct Customer

int choice;

int no\_addon1

int no\_addon2

int no\_addon3

int receiptno

float base\_price

float initial\_price

float addon\_price

float sum\_price

float extra price

char nasiAyam

char type

char addon

char extra\_addon

char new\_customer

end struct

display\_title()

display menu

nasiAyam()

for (i = 0; i < 1; i++)

display “[1] Steam Chicken Rice [2] Roasted Chicken Rice [3] Curry Chicken Rice [4] Fried Chicken Rice”

get cust[i].choice

switch (cust[i].choice)

case 1: cust[i].nasiAyam= "Steam Chicken Rice "

calculate cust[i].base\_price = 8.50

case 2: cust[i].nasiAyam= "Roasted Chicken Rice "

calculate cust[i].base\_price = 8.50

case 3: cust[i].nasiAyam= "Curry Chicken Rice "

calculate cust[i].base\_price = 9.50

case 4: cust[i].nasiAyam= "Curry Chicken Rice "

calculate cust[i].base\_price = 8.50

default: display “Invalid input. Please key-in again.”

nasiAyam()

type()

for (i = 0; i < 1; i++)

display “[1] Regular [2] Special \n Enter your order type:”

get cust[i].choice

switch (cust[i].choice)

case 1: calculate cust[i].initial\_price=cust[i].base\_price + 0

case 2: calculate cust[i].initial\_price=cust[i].base\_price + 1.50

addon()

for (i = 0; i < 1; i++)

display “[1] Liver [2] Gizzard [3] Egg [0] No add-on”

get cust[i].choice

switch (cust[i].choice)

case 1: cust[i].addon= " with Liver added"

calculate cust[i].addon\_price= cust[i].initial\_price+ 1.00;

case 2: cust[i].addon, " with Gizzard added"

calculate cust[i].addon\_price= cust[i].initial\_price+ 1.20

case 3: cust[i].addon, " with Egg added"

calculate cust[i].addon\_price= cust[i].initial\_price+ 1.50

case 0: cust[i].addon, " with Nothing add-on added"

default: display “Invalid input, please try again”

addon()

extra\_addon()

for (i = 0; i < 1; i++)

display “Do you prefer to add another add-on? [1] Yes [0] No”

get cust[i].choice

switch (cust[i].choice)

case 1: addon()

case 0: display “Price after add-on: RM%.2f\n”, cust[i].addon\_price

default: display “Invalid input. Please try again”

extra\_addon()

receipt ()

for (i = 0; i < 1; i++)

cust[i].receiptno=cust[i].receiptno+1

Open fw “Receipt.txt”

intput cust[i]. receiptno, cust[i].nasiAyam, cust[i].type, cust[i].no\_addon1

,cust[i].no\_addon2, cust[i].no\_addon3, cust[i].initial\_price from fw

Close fw

Calculate cust[i].sum\_price = cust[i].sum\_price + cust[i].initial\_price

total\_price()

calculate cust[i].sum\_price = cust[i].sum\_price + cust[i].initial\_price