



**UNIVERSITI TEKNOLOGI MARA  
KEDAH BRANCH  
SCHOOL OF INFORMATION SCIENCE  
COLLEGE OF COMPUTING, INFORMATICS AND MATHEMATICS**

**DIPLOMA IN LIBRARY INFORMATIC (CDIM144)  
IML208: PROGRAMMING FOR LIBRARIES**

**ASSESSMENT 1: INDIVIDUAL ASSIGNMENT**

***Prepared by:***

AWATIF BINTI AHMAD ZAIDI (2022618352)  
GROUP CDIM1443E

***Prepared for:***

SIR AIRUL SHAZWAN BIN NORSHAIMI

***Submission date:***

4 JANUARI 2024

**PROGRAMMING FOR CROCHET PURCHASE**

**PREPARED BY:**

AWATIF BINTI AHMAD ZAIDI

(202261835)

GROUP CDIM1443E

**IM144 – DIPLOMA IN INFORMATIC LIBRARY  
SCHOOL OF INFORMATION SCIENCE  
COLLEGE OF COMPUTING, INFORMATICS AND MATHEMATICS  
UNIVESITI TEKNOLOGI MARA (UITM)  
KEDAH BRANCH**

## **TABLE OF CONTENT**

<b>BIL</b>	<b>CONTENTS</b>	<b>PAGE</b>
1	Introduction	1
2	Flowchart	2
3	Coding	3-4
4	GUI	5
5	Database	6

# **INTRODUCTION**

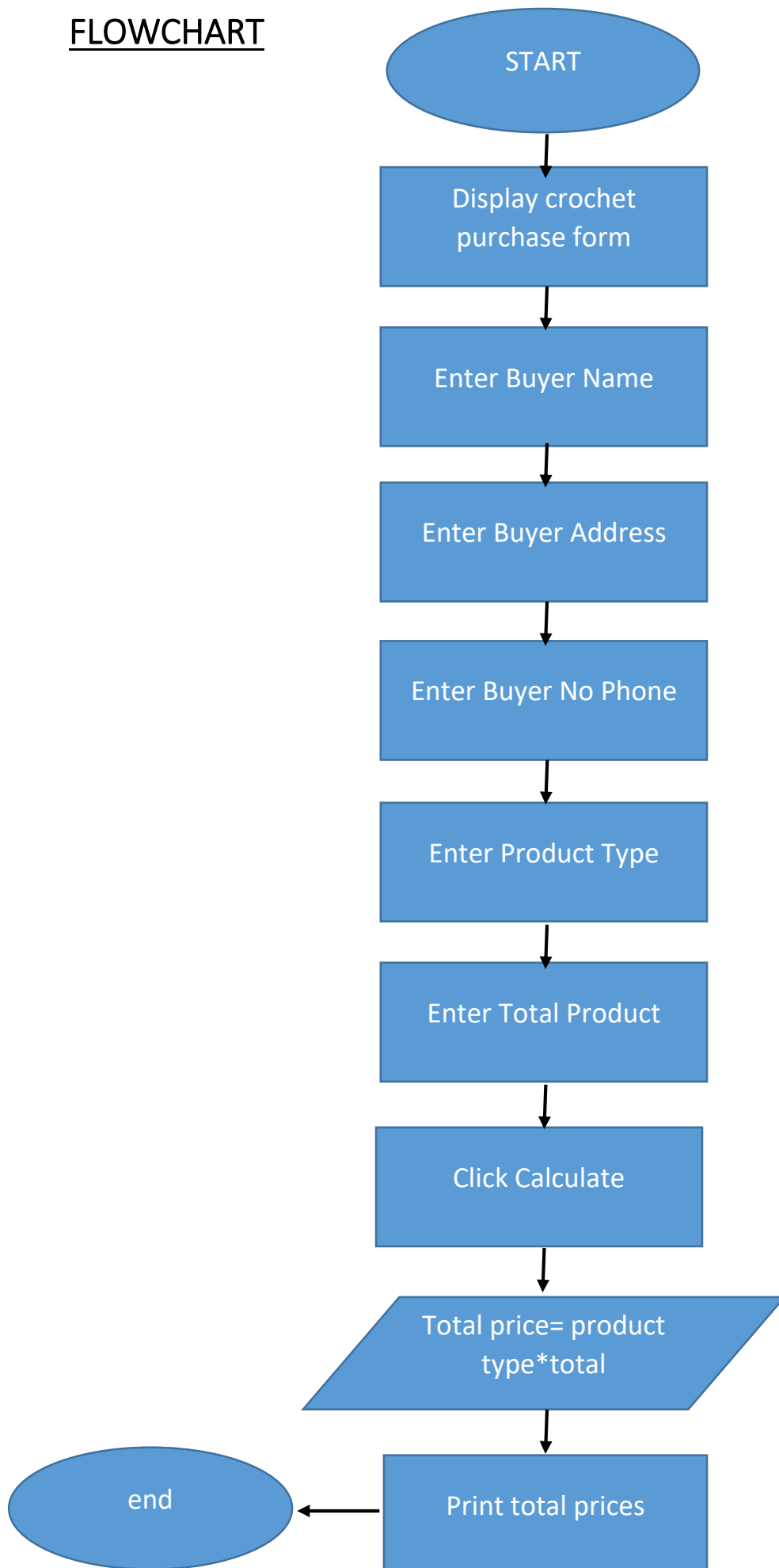
As we know that business is the one of important thing in all country. However in an increasingly developed world, even you do a small business you must make the proper ways to customer purchase the order.

For this assignment I have choose crochet purchase as my tittle. I have do the business that were related to crochet. However, I always have to face the problem when the customer want to make an order. Sometimes my team and I will miss some customer order because we just do the manual which is we just write the customer order in book. Other than that, the information about the customer order also not arranged and that's why the information will be mistaken.

After I realise about the problem I have make decision to try to make this tittle into my individual assignment. I believe that maybe one day the crochet business can use this python for our business to make the proper crochet purchase than can facilitate the customer and also we as a seller.

With this purchase way, seller can see the order that they get easily. The information would be more proper which is in database. Customer also easily to make the order which is they just need to key in their order and it automatically will send it to the buyer system.

## FLOWCHART



# CODE

```
crochet_purchase.py X  crochet_purchase.sql
crochet_purchase.py > collect_data
1 import tkinter as tk
2 from tkinter import messagebox
3 import mysql.connector
4
5 #connect to MySQL database
6 mydb= mysql.connector.connect(
7     host= "localhost",
8     user= "root",
9     password= "",
10    database= "crochet_purchase"
11 )
12
13 #cursor object
14 mycursor = mydb.cursor()
15
16 # calculation and database saving
17 def collect_data():
18     buyername = name_entry.get()
19     buyeraddress = address_entry.get()
20     buyernumber = phone_entry.get()
21     producttype = product_type.get() # Fixed variable name
22     total = total_entry.get()
23
24     print("Name:", buyername, "Address:", buyeraddress, "Phone Number", buyernumber)
25     print("Select Your Product", producttype, "Total:", total)
26     print("-----")
27
28     # Inserting data into a table
29     sql="INSERT INTO crochet_purchase (NAME,ADDRESS,NO_PHONE,PRODUCT_TYPE,TOTAL) VALUES (%s,%s,%s,%s,%s)"
30     val = (buyername,buyeraddress,buyernumber,producttype,total)
31
32     try:
```

```
crochet_purchase.py X  crochet_purchase.sql
crochet_purchase.py > collect_data
32     try:
33         mycursor.execute(sql, val)
34         mydb.commit()
35         print("Data inserted successfully!")
36     except mysql.connector.Error as err:
37         print(f"Error: {err}")
38         mydb.rollback()
39
40     jumlah = int(total_entry.get())
41
42     # prices of product
43     prices = {
44         "Product A": 6,
45         "Product B": 35,
46         "Product C": 10,
47         "Product D": 30,
48     }
49
50     # calculation
51     total_prices = prices[producttype] * jumlah # Fixed variable name
52
53     # output
54     output_label.config(text=f"Total Price: RM{total_prices}")
55     output_label.pack(pady=5)
56
57
58 # Main window
59 root = tk.Tk()
60 root.title("CROCHET BY WANIE")
61 root.geometry('500x600')
62 root.configure(bg= 'pink')
63
Ln 21, Col 60  Spaces: 4  UTF-8  CRLF  Python  3.11.7 64-bit (Microsoft Store)
```

```
crochet_purchase.py X  crochet_purchase.sql
crochet_purchase.py > collect_data
64 # Page title
65 label = tk.Label(root, text='CROCHET PURCHASE', font=("Input Mono", 30, "bold"), fg='FF1493', bg='pink')
66 label.pack(padx=20, pady=10)
67
68 frame = tk.Frame(root, bg='pink')
69 frame.pack()
70
71 info = tk.LabelFrame(frame, text="", bg='pink')
72 info.grid(row=0, column=1, sticky="News", padx=1, pady=1)
73
74 # customer information
75 name_label = tk.Label(info, text="Name:", bg='FF83FA', font=("Times New Roman", 15, "bold"))
76 name_label.pack()
77 name_entry = tk.Entry(info)
78 name_entry.pack(pady=15)
79
80 address_label = tk.Label(info, text="Address:", bg='FF83FA', font=("Times New Roman", 15, "bold"))
81 address_label.pack()
82 address_entry = tk.Entry(info)
83 address_entry.pack(pady=15)
84
85 phone_label = tk.Label(info, text="Phone Number:", bg='FF83FA', font=("Times New Roman", 15, "bold"))
86 phone_label.pack()
87 phone_entry = tk.Entry(info)
88 phone_entry.pack(pady=15)
89
90 price_frame = tk.LabelFrame(frame, text="", bg='pink')
91 price_frame.grid(row=0, column=0, sticky="News", padx=1, pady=1)
92
93 # Price List
94 price_text = tk.Text(price_frame, height=1, width=17, font=("bold", 20), fg='purple', bg='FF83FA')
95 price_text.pack(pady=5, padx=10)
96
Ln 21, Col 60  Spaces: 4  UTF-8  CRLF  Python  3.11.7 64-bit (Microsoft Store)
```

```

File Edit Selection View Go Run ... ← → INDIVIDUAL ASS(AWATIF)
crochet_purchase.py X crochet_purchase.sql
crochet_purchase.py > collect_data
...
93 # Price List
94 price_text = tk.Text(price_frame, height=1, width=17, font=("bold", 20), fg='purple', bg='#FF83FA')
95 price_text.pack(pady=5, padx=10)
96 # Pricebox
97 price_text.insert(tk.END, "PRODUCT & PRICE")
98 price_text.config(state="disabled")
99
100 # Product List
101 productA_text = tk.Text(price_frame, height=2, width=25, font=(10), fg='purple', bg='#EE3A8C')
102 productA_text.pack(pady=5)
103 # ProductA box
104 productA_text.insert(tk.END, "Product A: Keychain \nPrice: RM6")
105 productA_text.config(state="disabled")
106
107 # Product List
108 productB_text = tk.Text(price_frame, height=2, width=25, font=(10), fg='purple', bg='#FF82AB')
109 productB_text.pack(pady=5)
110 # ProductB Box
111 productB_text.insert(tk.END, "Product B: Plushie \nPrice: RM35")
112 productB_text.config(state="disabled")
113
114 # Product List
115 productC_text = tk.Text(price_frame, height=2, width=25, font=(10), fg='purple', bg='#EE3A8C')
116 productC_text.pack(pady=5)
117 # ProductC Box
118 productC_text.insert(tk.END, "Product C: Flower \nPrice: RM10")
119 productC_text.config(state="disabled")
120
121 # Product List
122 productD_text = tk.Text(price_frame, height=2, width=25, font=(10), fg='purple', bg='#FF82AB')
123 productD_text.pack(pady=5)
124 # ProductD Box

```

```

crochet_purchase.py X crochet_purchase.sql
crochet_purchase.py > collect_data
...
121 # Product List
122 productD_text = tk.Text(price_frame, height=2, width=25, font=(10), fg='purple', bg='#FF82AB')
123 productD_text.pack(pady=5)
124 # ProductD Box
125 productD_text.insert(tk.END, "Product D: Beanie \nPrice: RM30")
126 productD_text.config(state="disabled")
127
128 product = tk.LabelFrame(frame, bg='pink')
129 product.grid(row=1, column=0, sticky="News", padx=1, pady=1)
130
131 # product type dropdown
132 ttl_label = tk.Label(product, text="Choose Your Product", bg='#FF83FA', font=("Times New Roman", 15, "bold"))
133 ttl_label.pack(padx=5)
134
135 product_type = tk.StringVar(root)
136 product_type.set("Select Your Product")
137 product_dropdown = tk.OptionMenu(product, product_type, "Product A", "Product B", "Product C", "Product D")
138 product_dropdown.pack(pady=10)
139
140 # total entry
141 total_label = tk.Label(product, text="Total:", bg='#FF83FA', font=("Times New Roman", 15, "bold"))
142 total_label.pack()
143 total_entry = tk.Entry(product)
144 total_entry.pack(pady=5)
145
146 total = tk.LabelFrame(frame, text="", bg='pink')
147 total.grid(row=1, column=1, sticky="News", padx=1, pady=1)
148
149 # Save Button
150 save_button = tk.Button(total, text="Calculate", font=("Times New Roman", 15, "bold"), bg='#FF6A6A', command=collect_data)
151 save_button.pack(pady=10)
152

```

# GUI

 CROCHET BY WANIE

## CROCHET PURCHASE

### PRODUCT & PRICE

Product A: Keychain  
Price: RM6

Product B: Plushie  
Price: RM35

Product C: Flower  
Price: RM10

Product D: Beanie  
Price: RM30

### Choose Your Product

Select Your Product

Total:

Name:

Address:

Phone Number:

Calculate

Prices Product



# DATABASE

The screenshot shows the phpMyAdmin web interface. The left sidebar displays a database structure with the following items: New, crochet\_purchase, New, crochet\_purchase, information\_schema, mysql, performance\_schema, phpmyadmin, python\_example, and test. The main panel shows the 'Table: crochet\_purchase' view. A warning message states: 'Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.' Below this, a green status bar indicates 'Showing rows 0 - 3 (4 total. Query took 0.0003 seconds.)'. The SQL query entered is 'SELECT \* FROM `crochet\_purchase`'. Below the query, there are options for 'Show all', 'Number of rows: 25', and a 'Filter rows' search box. The 'Extra options' section shows a table with 5 columns: NAME, ADDRESS, NO\_PHONE, PRODUCT\_TYPE, and TOTAL. The table contains 4 rows of data. Below the table, there are options for 'Show all', 'Number of rows: 25', and a 'Filter rows' search box. The 'Query results operations' section includes buttons for 'Print', 'Copy to clipboard', 'Export', 'Display chart', and 'Create view'. At the bottom, there is a 'Bookmark this SQL query' button and a 'Console' tab.

Server: 127.0.0.1 » Database: crochet\_purchase » Table: crochet\_purchase

Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

Showing rows 0 - 3 (4 total. Query took 0.0003 seconds.)

SELECT \* FROM `crochet\_purchase`

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

Show all | Number of rows: 25 | Filter rows: Search this table

Extra options

NAME	ADDRESS	NO_PHONE	PRODUCT_TYPE	TOTAL
dxfgghjk	sdfgfgghij	123456789	Product A	3
izzat	langgar	123456789	Product A	5
alin	kedah	1234567890	Product A	4
alif	kedah	1124156564	Product C	3

Show all | Number of rows: 25 | Filter rows: Search this table

Query results operations

Print Copy to clipboard Export Display chart Create view

Bookmark this SQL query

Console