

INFORMATION & COMMUNICATION TECHNOLOGY DEPARTMENT
SESI 2: 2023/2024
FINAL ASSESSMENT: PRACTICAL TEST
SET 1

COURSE CODE: DFP40203	COURSE NAME: PYTHON PROGRAMMING	
NAME: NADIA BINTI SHAHRUL AZMEE		
REGISTRATION NO: 10DDT22F1051	CLASS: DDT5C	DURATION: 2 HOURS
CLO1: Construct Python application based on given scenario (P4,PLO3)		

ANSWER : CODES

Question 1

```
PS C:\Users\Administrator> & C:/Users/Administrator/AppData/Local/Programs/Python/Python39-64/python.exe main.py
py.py
Name: Nadia
Vehicle number: PNG6354
Number of summons to be paid: 3
Traceback (most recent call last):
  File "d:\10DDT22F1051\main.py.py", line 41, in <module>
    summonsPrice = user.summonCalculate()
                   ^^^^^
NameError: name 'user' is not defined. Did you mean: 'super'?
```

- i. NameError – in line 41 the run code error because 'user' is not defined in the code

```

1  import time
2  from datetime import date
3
4  class ABCSummonsUnit:
5      def __init__(self, totalSummons):
6          self.totalSummons = totalSummons
7
8      def countDay(self, summonsDate, today):
9          date_format = "%Y-%m-%d"
10         a = time.mktime(time.strptime(summonsDate, date_format))
11         b = time.mktime(time.strptime(today, date_format))
12         delta = b - a
13
14         return int(delta / 86400)
15
16     def calculatePrice(self, discount, duration):
17         charge = duration * 20
18         if discount.lower() == "yes":
19             charge = duration * 15
20         return charge
21
22     def summonCalculate(self):
23         print("\nSummons "+str(i+1)+" : ")
24         today = str(date.today())
25         print("Today's date: ", today)
26         summonsDate = input("Date of issue of summons (YYYY-MM-DD): ")
27         duration = self.countDay(summonsDate, today)
28         print("Duration of summonses in days:", duration)
29
30         discount = input("Does ABC Summons Unit offer a discount for this summons? (YES/NO): ")
31
32         payment = self.calculatePrice(discount, duration)
33
34
35         return payment

```

ii.

```

38  name = input("Name: ")
39  vehicleNumber = input("Vehicle number: ")
40  totalSummons = int(input("Number of summons to be paid: "))
41
42  abc = ABCSummonsUnit(totalSummons)
43  totalPayment = 0
44  for i in range(totalSummons):
45      summonsPrice = abc.summonCalculate()
46      totalPayment += summonsPrice
47
48  print("\nTotal Summons Amount (RM): ", totalPayment)

```

iii.

Price.py

```

1  def calculatePrice(discount, duration):
2      if discount.lower() == "yes":
3          charge = duration * 20
4      else:
5          return charge
6

```

Day.py

```
1  import time
2
3  def countDay(summonsDate, today):
4      date_format = "%Y-%m-%d"
5      a = time.mktime(time.strptime(summonsDate, date_format))
6      b = time.mktime(time.strptime(today, date_format))
7      delta = b - a
8
9      return int(delta / 86400)
10
```

iv.

Question 2

```
1  import mysql.connector
2  from mysql.connector import Error
3
4  class connection():
5      def __init__(self):
6          self.mydb = mysql.connector.connect(
7              host="localhost",
8              user="root",
9              password=""
10         )
11
12     def create(self):
13         mycursor = self.mydb.cursor()
14         try:
15             mycursor.execute("CREATE DATABASE IF NOT EXISTS ABCSummonsUnit")
16         except Error as e:
17             print(f"Error : {e}")
18
19         self.database = mysql.connector.connect(
20             host="localhost",
21             user="root",
22             password="",
23             database="ABCSummonsUnit"
24         )
25
26     def table(self):
27         cursorObj = self.database.cursor()
28         abc = """CREATE TABLE IF NOT EXISTS summonInfo (
29             name VARCHAR(50),
30             vehicleNumber VARCHAR(10) NOT NULL,
31             summonsDate VARCHAR(10),
32             duration int(10),
33             discount VARCHAR(10),
34             amount VARCHAR(10)
35             PRIMARY KEY (vehicleNumber))"""
36         cursorObj.execute(abc)
37         self.database.commit()
```

Question 3

```
1  from tkinter import *
2  from tkinter import ttk
3  import Price as p
4  import Day as d
5  from datetime import date
6  import connection as data
7
8  db = data.mydb.cursor()
9
10 def message(msg):
11     root = Tk()
12     root.geometry("250x100")
13     w = Label(root, text = 'ABC Summons Unit', font = "20")
14     w.pack()
15
16     message = Message(root, text = msg)
17     message.pack()
18     root.mainloop()
19
20 def summonCalculate():
21     today = str(date.today())
22     summonDate = summonDateR.get()
23     duration = d.coundDay(summonDate, today)
24     durationL.config(text=duration)
25     discount = discountR.get()
26     payment = p.calculatePrice(discount, duration)
27     ammountL.config(text=payment)
28
29 def insert():
30     name = nameE.get()
31     vehicleNumber = vehicleNumberE.get()
32     summonDate = summonDateR.get()
33     duration = durationL.cget("text")
34     discount = discountR.get()
35     ammount = ammountL.cget("text")
36
37     try:
38         db.execute('INSERT INTO summonInfo (name, vehicleNumber , summonDate, duration, discount, ammount) VALUES (%s,%s,%s,%s,%s,%s)',
39             (name, vehicleNumber, summonDate, duration, discount, ammount))
40         data.mydb.commit()
41         message("***Success** Data has been successfully inserted.")
42
43     except Exception as e:
44         data.mydb.rollback()
45         message("Error occurred: " + str(e))
```

```

47 def delete():
48     vehicleNumber = vehicleNumberE.get()
49     try:
50         db.execute('DELETE FROM summonInfo WHERE vehicleNumber = %s', (vehicleNumber,))
51         data.mydb.commit()
52         message("**Success** Data has been successfully deleted.")
53     except Exception as e:
54         data.mydb.rollback()
55         message("Error occurred: " + str(e))
56
57 window = Tk()
58 window.title("ABC Summons Unit")
59
60 up=Label(window, text="----- Summons Form -----", font=("Arial Bold", 10))
61 up.grid(columnspan=3, row=3, sticky='S')
62
63 nameLbl=Label(window, text="Name : ")
64 nameLbl.grid(column=0, row=4, sticky='W')
65 nameE=Entry(window, width=30)
66 nameE.grid(column=1, columnspan=2, row=4, sticky='W')
67
68 vehicleNumberLbl=Label(window, text="Vehicle Number : ")
69 vehicleNumberLbl.grid(column=0, row=5, sticky='W')
70 vehicleNumberE=Entry(window, width=30)
71 vehicleNumberE.grid(column=1, columnspan=2, row=5, sticky='W')
72
73 middle=Label(window, text="--"*20)
74 middle.grid(columnspan=3, row=8, sticky='S')
75
76 calculateLbl=Label(window, text="Summons Calculate:", font=("Arial Bold", 10))
77 calculateLbl.grid(column=0, row=9, sticky='W')
78
79 todayR = date.today()
80 todayLbl=Label(window, text="Date of today: ")
81 todayLbl.grid(column=0, row=10, sticky='W')
82 todayE=Label(window, text=f"{todayR:%Y-%m-%d}", textvariable=todayR, width=13, borderwidth=2, relief="groove")
83 todayE.grid(column=1, row=10, sticky='W')
84
85 summonDateR = StringVar()
86 summonDateLbl=Label(window, text="Summons Date: (YYYY-MM-DD)")
87 summonDateLbl.grid(column=0, row=11, sticky='W')
88 summonDateE=Entry(window, width=15, textvariable=summonDateR)
89 summonDateE.grid(column=1, row=11, sticky='W')

```

```

91 discountR = StringVar()
92 discountlbl=Label(window, text="Discount for this summons?:")
93 discountlbl.grid(column=0, row=12, sticky='W')
94 discountC = ttk.Combobox(window, textvariable=discountR, values=["YES", "NO"], width=6)
95 discountC.grid(column=1, row=12, sticky='W')
96
97 durationlbl=Label(window, text="Duration of summonses in days:")
98 durationlbl.grid(column=0, row=13, sticky='W')
99 durationL=Label(window, width=8, borderwidth=2, relief="groove")
100 durationL.grid(column=1, row=13, sticky='W')
101
102 down=Label(window, text="--"*20)
103 down.grid(columnspan=3, row=14, sticky='S')
104
105 ammountlbl=Label(window, text="Total Summons Amount (RM):")
106 ammountlbl.grid(column=0, row=15, sticky='W')
107 ammountL=Label(window, text='', width=8, borderwidth=2, relief="groove")
108 ammountL.grid(column=1, row=15, sticky='W')
109
110 last=Label(window, text="--"*20)
111 last.grid(columnspan=3, row=16, sticky='S')
112
113 menubar = Menu(window)
114 filemenu = Menu(menubar, tearoff=0)
115 filemenu.add_command(label="Calculate", command=summonCalculate)
116 filemenu.add_command(label="Submit", command=insert)
117 filemenu.add_command(label="Delete", command=delete)
118 filemenu.add_separator()
119 menubar.add_cascade(label="File", menu=filemenu)
120 window.config(menu=menubar)
121
122 window.geometry("370x300")
123 window.mainloop()

```

ANSWER : OUTPUT

Questions 1

```
PS C:\Users\Administrator> & C:/Users/Administrator/AppData/Local/Programs/Python/Python312/python.exe d:/10DDT22F1051/main.py
Name: Nadia
Vehicle number: PNG6354
Number of summons to be paid: 3
Traceback (most recent call last):
  File "d:\10DDT22F1051\main.py", line 41, in <module>
    summonsPrice = user.summonCalculate()
                   ^^^^^
NameError: name 'user' is not defined. Did you mean: 'super'?
```

- i. NameError – in line 41 the run code error because 'user' is not defined in the code

ii.

```
PS C:\Users\Administrator> & C:/Users/Administrator/AppData/Local/Programs/Python/Python312/python.exe d:/10DDT22F1051/main.py
Name: Nadia
Vehicle number: BMW2004
Number of summons to be paid: 2

Summons 1:
Today's date: 2024-06-04
Date of issue of summons (YYYY-MM-DD): 2024-06-01
Duration of summonses in days: 3
Does ABC Summons Unit offer a discount for this summons? (YES/NO): YES

Summons 2:
Today's date: 2024-06-04
Date of issue of summons (YYYY-MM-DD): 2024-05-24
Duration of summonses in days: 11
Does ABC Summons Unit offer a discount for this summons? (YES/NO): YES

Total Summons Amount (RM): 210
```

iii.

```
PS C:\Users\Administrator> & C:/Users/Administrator/AppData/Local/Programs/Python/Python312/python.exe d:/10DDT22F1051/Price.py
PS C:\Users\Administrator> & C:/Users/Administrator/AppData/Local/Programs/Python/Python312/python.exe d:/10DDT22F1051/Day.py
```

iv.

```
PS C:\Users\Administrator> & C:/Users/Administrator/AppData/Local/Programs/Python/Python312/python.exe d:/10DDT22F1051/main.py
```

ANSWER: DATABASE SCREENSHOTS