

## Table Of Figures

Figure 1Part A - Task 1) a. ....	2
Figure 2 Part A - Task 2) b. ....	2
Figure 3 Part A - Task 2) a. ....	2
Figure 4 Part A - Task 2) b. ....	2
Figure 5 Part A - Task 3) a., b. and c.....	3
Figure 6 Part A - Task 4 .....	3
Figure 7 Part B - Task 1 Error 1 .....	5
Figure 8 Part B - Task 1 Error 2.....	5
Figure 9 Part B - Task 1 .....	5
Figure 10 Part B - Task 3 .....	6
Figure 11 Part B - Task 3 .....	6

## PART A

### TASK

#### 1. Welcome Message and Setup

- a. Print a welcome message for the swimming pool membership system.

```
3 print("thnak you for using the swimming pool membership system")
```

*Figure 1 Part A - Task 1) a.*

Figure 1 shows the print out of the welcome for the user

- b. Display today's date using a variable

```
4 datetime = datetime.datetime.now()
5 print(f"Today's date and time: {datetime}")
```

*Figure 2 Part A - Task 2) b.*

Figure 2 shows will show the date and time that the user has used the system.

#### 2. Create user registration

- a. Ask the user for their name, age, and membership type (e.g.: adult, junior, senior, student and family)

```
7 member_name = input("Enter you name: ")
8 member_age = int(input("Enter your age: "))
9 membership_type = input("Enter membership type: (e.g: adult, junoir, senior, student and family)")
```

*Figure 3 Part A - Task 2) a.*

Figure 3 will want the user to key in their details which is the name, age, and their membership type

- b. Store them in variables and display back the entered details

```
11 print(f"Member Name: {member_name}\nMember Age: {member_age}\nMembership type: {membership_type}")
```

*Figure 4 Part A - Task 2) b.*

After getting all the information from figure 3, the information will be displayed with the format of figure 4

#### 3. Age-based membership eligibility

- a. If age < 12 \_ "Not eligible for membership"

- b. If age 12 – 60 \_ “Standard membership granted”
- c. If age > 60 \_ “Senior membership granted”

```
13  if member_age < 12:
14      print("Not eligible for membership")
15  elif 12 > member_age and member_age < 60:
16      print("Standard membership granted")
17  elif member_age > 60 and member_age < 100:
18      print("Senior membership granted")
19  else:
20      print("Invalid member age")
```

Figure 5 Part A - Task 3) a., b. and c.

Figure 5 shows the if else statement of the age, after the displaying the information of the user, this statement will decide on which membership are the member going to take according to the age that the user input.

- 4. Swimming session booking simulation
  - a. Ask the user how many swimming sessions they want to book
  - b. Using a loop, display “Booking session X “for each session number until all sessions are booked.

```
22  sessions = int(input("Enter how many swimming sessions you want to book?: "))
23
24  for x in range(sessions):
25      x = 1
26      print(f"Sessions Left: {sessions}")
27      choice = input("\nAre you going to continue the session?: (Yes/No) ")
28      if choice == "Yes":
29          print("Session continue...\n")
30          x = x + 1
31          sessions = sessions - 1
32      elif choice == "No":
33          break
34      elif x >= sessions:
35          break
36      else:
37          print("Invalid input\n")
```

Figure 6 Part A - Task 4

Figure 6 shows the swimming session booking simulation, through the simulation, we can acquire the session of the user that he wanted to book. When the user chooses to continue the session, it will show the session that is left to the user and when the session reaches 0 the system

will exit automatically. The user can exit the system also by entering not when at the choice selection.

## PART B

### TASK

1. Analysis the code and list THREE (3) syntax or logic errors in the provided code

```
23 v class Book:
24 v     def __init__(self,title,author):
25         self.title = title
26         self.author = author
27
28 v     def display_info():
29         print(f>Title: {self.title}, Author: {self.author}"]
```

Figure 7 Part B - Task 1 Error 1

Figure 7 shows the first error that is occurring in the library\_module.py, the define of display\_info can not be defined since display\_info() is defined into none; to solve this problem we just need to add self into the empty brackets of display\_info() as the figure below shows.

```
10 from library_module import book
```

Figure 8 Part B - Task 1 Error 2

Figure 8 shows the second error that is occurring in the main.py, importing book from library\_module is not possible since the class name that is in the liobrary\_module is named as Book. Change from book to Book will solve the problem as the figure below shows

```
26 print("\nBook List from File: ")
27 v for line in lines:
28     t, a = line.strip().split(":")
29     b = books(f"{t}, {a}")
30     b.display_info()
```

Figure 9 Part B - Task 1

Figure 9 shows the last error that occurred, line 29 is the error, which is a type error since dictionary object is not callable.

2. Explain why each error will cause the program to fail or behave incorrectly

The error will cause the program to not be able to identify the declaration that is declared by the user.

3. Correct the errors in library\_module.py and main.py so the program successfully

```
23 class Book:
24     def __init__(self, title, author):
25         self.title = title
26         self.author = author
27
28     def display_info(self):
29         print(f"Title: {self.title}, Author: {self.author}")
```

Figure 10 Part B - Task 3

```
10 from library_module import Book
```

Figure 11 Part B - Task 3

Figure 10 and 11 shows the error correction that will be able to make the program to run

4. Identify ONE (1) advantage of separating the class into a different module file

Separating classes into different modules allows the user to have a clear view of what the class does, and it is easier to keep track of the module

5. Modify the program so that after reading the books from the file, it also displays the total number of books stored.