



Student Name and Number : Sakshi Umesh Teli (20079303)

Course Title : M.Sc. Cyber Security

Lecturer Name: Swati Dongre

Module/Subject Title: Advanced Programming Techniques

Assignment Title: CA_ONE_(30%)

No of Words: 698

TABLE OF CONTENTS

1. Introduction.....	2
2. Part I: C# Programming.....	3-10
2.1 Contact Book.....	3-5
2.2 File Extension System.....	6-10
3. Part II: Python Programming.....	11-14
3.1 Client–Server Application.....	11-12
3.2 Web Scraping and CSV Handling.....	12-14
4. Conclusion.....	15
5. Reference.....	15
6. GitHub Repository Link.....	15

1. INTRODUCTION

In this assignment , I developed four applications in Python and C#. I developed a file extension information system and a contact book in C#. I wrote a hotel web-scraping application in Python which writes information to CSV file and a client-server application in Python to record student admission. These tasks helped me to practice programming, working with data, and solve real-life problems.

2. PART I: C# PROGRAMMING

2.1 Contact Book

In this assignment, I developed a C# contact book program . This project consisted of two files :

a) **Program.cs** for the main menu and all functions.

b) **ContactBook.cs** for the **ContactBook class**.

I created a **ContactBook** class with a name, company , phone , email and birthdate of each contact . The program starts with 20 contacts in an array . The menu lets the user to **add , view , update , or delete** a contact , also it checks that the phone number is 9 digits. I used basic **OOP concepts** and tested the program using **dotnet run** , which keeps going until the user exits .

The screenshot shows the Visual Studio Code interface with the terminal tab active. The terminal window displays the execution of a C# application named 'Que.1'. The application starts by listing 20 contacts with their last names and phone numbers. It then presents a menu with options 1 through 5 for adding, viewing, updating, or deleting contacts, and option 0 to exit. The user selects option 3 to view a contact, enters a mobile number, and views detailed contact information including first name, last name, company, mobile number, email, and birthdate. Finally, the user selects option 4 to update a contact, enters a new mobile number, and provides updated details for the contact.

```
PS C:\Users\Admin\Desktop\sakshi20079303\Que.1> dotnet run
Person13 lastname13 -- 12345678913
Person14 lastname14 -- 12345678914
Person15 lastname15 -- 12345678915
Person16 lastname16 -- 12345678916
Person17 lastname17 -- 12345678917
Person18 lastname18 -- 12345678918
Person19 lastname19 -- 12345678919
Person20 lastname20 -- 12345678920
Sakshi teli -- 123456789

--- Contact Book Menu ---
1. Add contact
2. Show All Contacts
3. Show one Contacts
4. Update Contact
5. Delete Contact
0. Exit
Enter Option
3
Enter Mobile Number :123456789

---Contact Details ---
First Name :Sakshi
Last Name :teli
Company :TMV
Mobile : 123456789
Email : sakshi@gmail.com
Birthdate : 01-03-2004

--- Contact Book Menu ---
1. Add contact
2. Show All Contacts
3. Show one Contacts
4. Update Contact
5. Delete Contact
0. Exit
Enter Option
4
Enter Mobile Number To Update : 123456789
New First Name :Tanvi
New Last Name :pashilkar
New Company :MTC
New Email:tnavi@gmail.com
New Birthdate :22-02-2004
Contact Updated !
```

The screenshot shows a Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, ..., back arrow, forward arrow.
- Explorer:** Shows a project named "QUE.1" containing files: bin, obj, ConsoleApp1.csproj, ContactBook.cs, Program.cs, and Que.1.sln. A file named "Program.cs" is currently selected.
- Terminal:** Displays the output of a dotnet run command. The application menu includes:
 1. Add contact
 2. Show All Contacts
 3. Show one Contacts
 4. Update Contact
 5. Delete Contact
 0. Exit

Enter Option

1
First Name :Sakshi
Last Name :teli
Company : TMV
Mobile Number (9 digits) : 123456789
Email : sakshi@gmail.com
Birthdate :01-03-2004
Contact Added !

--- Contact Book Menu ---
1. Add contact
2. Show All Contacts
3. Show one Contacts
4. Update Contact
5. Delete Contact
0. Exit
Enter Option

2
Person1 lastname1 -- 1234567891
Person2 lastname2 -- 1234567892
Person3 lastname3 -- 1234567893
Person4 lastname4 -- 1234567894
Person5 lastname5 -- 1234567895
Person6 lastname6 -- 1234567896
Person7 lastname7 -- 1234567897
Person8 lastname8 -- 1234567898
Person9 lastname9 -- 1234567899
Person10 lastname10 -- 12345678910
Person11 lastname11 -- 12345678911
Person12 lastname12 -- 12345678912
Person13 lastname13 -- 12345678913
Person14 lastname14 -- 12345678914
Person15 lastname15 -- 12345678915
Person16 lastname16 -- 12345678916
Person17 lastname17 -- 12345678917
Person18 lastname18 -- 12345678918
Person19 lastname19 -- 12345678919
Person20 lastname20 -- 12345678920
Sakshi teli -- 123456789
- Sidebar:** Includes icons for Explorer, Search, Open, Find, Test, Database, and Stack Overflow, along with links to Outline, Timeline, MySQL, and Solution Explorer.

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, ..., ⏪ ⏩
- Search Bar:** Que.1
- Explorer View (Left):** Shows a project named "QUE.1" containing "bin", "obj", "ConsoleApp1.csproj", "ContactBook.cs", and "Program.cs".
- Terminal View (Center):** Displays the output of the "dotnet run" command for a C# application. The application's code includes a MainClass with a static Main method. It prints a menu with options 1 through 5, then asks for a mobile number to delete, which is 123456789. It then lists 20 contacts from Person1 to Person20, each with a unique last name and phone number. Finally, it prints the same contact book menu again.
- Bottom Left:** Icons for Outline, Timeline, MySQL, and Solution Explorer.
- Bottom Right:** A small number "5" indicating the page number.

```
PS C:\Users\Admin\Desktop\sakshi20079303\Que.1> dotnet run
4. Update Contact
5. Delete Contact
0. Exit
Enter Option
5
Enter Mobile Number To Delete : 123456789
Contact Deleted !

--- Contact Book Menu ---
1. Add contact
2. Show All Contacts
3. Show one Contacts
4. Update Contact
5. Delete Contact
0. Exit
Enter Option
2
Person1 lastname1 -- 1234567891
Person2 lastname2 -- 1234567892
Person3 lastname3 -- 1234567893
Person4 lastname4 -- 1234567894
Person5 lastname5 -- 1234567895
Person6 lastname6 -- 1234567896
Person7 lastname7 -- 1234567897
Person8 lastname8 -- 1234567898
Person9 lastname9 -- 1234567899
Person10 lastname10 -- 12345678910
Person11 lastname11 -- 12345678911
Person12 lastname12 -- 12345678912
Person13 lastname13 -- 12345678913
Person14 lastname14 -- 12345678914
Person15 lastname15 -- 12345678915
Person16 lastname16 -- 12345678916
Person17 lastname17 -- 12345678917
Person18 lastname18 -- 12345678918
Person19 lastname19 -- 12345678919
Person20 lastname20 -- 12345678920

--- Contact Book Menu ---
1. Add contact
2. Show All Contacts
3. Show one Contacts
4. Update Contact
5. Delete Contact
0. Exit
Enter Option
```

2.2 File Extension System

In response to this question, I created a C# console application that displays the information about different file extensions. In this project .

I was working with two files :

- a) Program.cs** – The menu and all functions are contained here.
- b) FileExtension.cs** – The class contains the FileExtension .

FileExtension class contains two aspects:

- a) The file extension (like .mp4,.pdf)
- b) Its description (like video file, document , audio file)

I did a program where 20 default file extensions were created by default with the help of the **AddDefaultExtension()** method. The menu will allow the user to **view , add, delete ,update** or search an extension . In case the user enters an extension but forgets to add a dot , the program will add it automatically . In case there is no extension , a message stating not found is displayed . I was able to apply simple OOP principles such as classes , objects , and methods and debugged all of them by dotnet run .

The screenshot shows a terminal window within a dark-themed IDE interface. The terminal output is as follows:

```
Program.cs • FileExtension.cs
ConsoleApp1 > Program.cs > ...
class MainClass
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POLYGLOT NOTEBOOK
PS C:\Users\Admin\Desktop\sakshi20079303\Que.2> cd "C:\Users\Admin\Desktop\sakshi20079303\Que.2\ConsoleApp1"
>>
PS C:\Users\Admin\Desktop\sakshi20079303\Que.2\ConsoleApp1> dotnet run

---File Extension ---
1.Show All Extensions
2.Show One Extensions
3.Add Extensions
4.Update Extensions
5.Delete Extensions
0.Exit
Enter option: 1

Supported Extensions :
.mp4 - Video File
.avi - Video File
.mov - Video File
.mkv - Video File
.webm - Video File
.mp3 - Audio File
.wav - Audio File
.flac - Audio File
.jpg - Image File
.jpeg - Image File
.png - Image File
.gif - Image File
.pdf - Document
.docx - Word Document
.xlsx - Excel Sheet
.pptx - PowerPoint
.txt - Text File
.zip - Compressed File
.rar - Compressed File
.7z - Compressed File

---File Extension ---
1.Show All Extensions
2.Show One Extensions
3.Add Extensions
4.Update Extensions
5.Delete Extensions
0.Exit
Enter option: 2
Enter Extension : .7z
Extension : .7z
Description: Compressed File

---File Extension ---
1.Show All Extensions
```

The screenshot shows a dark-themed IDE interface, likely Visual Studio Code, with a terminal window open in the center-right area. The terminal window displays a command-line application for managing file extensions. The application starts with a help menu and then asks for a file extension to add. It lists supported extensions and their descriptions. The terminal output is as follows:

```
PS C:\Users\Admin\Desktop\sakshi20079303\Que.2\ConsoleApp1> dotnet run
---File Extension ---
1.Show All Extensions
2.Show One Extensions
3.Add Extensions
4.Update Extensions
5.Delete Extensions
0.Exit
Enter option: 3
Enter Extension : .png
Enter Description : Image file
Entension added !

---File Extension ---
1.Show All Extensions
2.Show One Extensions
3.Add Extensions
4.Update Extensions
5.Delete Extensions
0.Exit
Enter option: 1

Supported Extensions :
.mp4 - Video File
.avi - Video File
.mov - Video File
.mkv - Video File
.webm - Video File
.mp3 - Audio File
.wav - Audio File
.flac - Audio File
.jpg - Image File
.jpeg - Image File
.png - Image File
.gif - Image File
.pdf - Document
.docx - Word Document
.xlsx - Excel Sheet
.pptx - PowerPoint
.txt - Text File
.zip - Compressed File
.rar - Compressed File
.7z - Compressed File
.png - Image file
```

The left side of the interface features a sidebar with various icons and sections like EXPLORER, TERMINAL, and SOLUTION EXPLORER. The EXPLORER section shows a project named "ConsoleApp1" with files like "Program.cs", "FileExtension.cs", and "ConsoleApp1.csproj". The TERMINAL section shows the command "dotnet run" being executed. The bottom status bar indicates "Projects: ConsoleApp1" and "Debug Any CPU".

The screenshot shows a terminal window within a dark-themed IDE interface. The terminal output is as follows:

```
Program.cs ● FileExtension.cs
ConsoleApp1 > Program.cs > ...
  ↓ class MainClass
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POLYGLOT NOTEBOOK
PS C:\Users\Admin\Desktop\sakshi20079303\Que.2\ConsoleApp1> dotnet run
2.Show One Extensions
3.Add Extensions
4.Update Extensions
5.Delete Extensions
0.Exit
Enter option: 4
Extension : .png
New Description : Picture
Extension Updated

---File Extension ---
1.Show All Extensions
2.Show One Extensions
3.Add Extensions
4.Update Extensions
5.Delete Extensions
0.Exit
Enter option: 1

Supported Extensions :
.mp4 - Video File
.avi - Video File
.mov - Video File
.mkv - Video File
.webm - Video File
.mp3 - Audio File
.wav - Audio File
.flac - Audio File
.jpg - Image File
.jpeg - Image File
.png - Picture
.gif - Image File
.pdf - Document
.docx - Word Document
.xlsx - Excel Sheet
.pptx - PowerPoint
.txt - Text File
.zip - Compressed File
.rar - Compressed File
.7z - Compressed File
.png - Image file

---File Extension ---
1.Show All Extensions
2.Show One Extensions
3.Add Extensions
4.Update Extensions
5.Delete Extensions
```

The terminal also displays the current working directory and command: PS C:\Users\Admin\Desktop\sakshi20079303\Que.2\ConsoleApp1> dotnet run.

The screenshot shows a terminal window within a code editor interface. The terminal is running a C# application named 'ConsoleApp1'. The application displays a menu for managing file extensions:

```
Program.cs • FileExtension.cs
ConsoleApp1 > Program.cs > ...
5. class MainClass
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POLYGLOT NOTEBOOK
PS C:\Users\Admin\Desktop\sakshi20079303\Que.2\ConsoleApp1> dotnet run
2.Show One Extensions
3.Add Extensions
4.Update Extensions
5.Delete Extensions
0.Exit
Enter option: 5
Extension : .png
Extension Deleted :

---File Extension ---
1.Show All Extensions
2.Show One Extensions
3.Add Extensions
4.Update Extensions
5.Delete Extensions
0.Exit
Enter option: 1

Supported Extensions :
.mp4 - Video File
.avi - Video File
.mov - Video File
.mkv - Video File
.webm - Video File
.mp3 - Audio File
.wav - Audio File
.flac - Audio File
.jpg - Image File
.jpeg - Image File
.gif - Image File
.pdf - Document
.docx - Word Document
.xlsx - Excel Sheet
.pptx - PowerPoint
.txt - Text File
.zip - Compressed File
.rar - Compressed File
.7z - Compressed File
.png - Image file

---File Extension ---
1.Show All Extensions
2.Show One Extensions
3.Add Extensions
4.Update Extensions
5.Delete Extensions
0.Exit
Enter option: 
```

The application has deleted the file extension '.png' and then listed all supported file extensions. The terminal also shows the supported extensions for various file types.

3. PART II: PYTHON PROGRAMMING

3.1 Client–Server Application

I created a server program based on **Python Sockets**, which waits to have a client . I created a client program that sends information about student admission to the server . I used **SQLite** to create a database file that contains all of the student data .

Once the data is received by the server , a unique application number is created and stored . The server sends the application number to the client . The client is sent back the application number by the server. The client shows the application number to the user . This work makes use of **databases**, **TCP communication** and **basic string manipulation**.

```
File Edit Selection ... ← → sakshi20079303 powershell - Que.3 - ...  
EXPLORER PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POLYGLOT NOTEBOOK  
SAKSHI20079303  
Que.1  
Que.2  
ConsoleApp1  
ConsoleApp1.sln  
Que.3  
Admission.db  
Que3_client.py  
Que3_server.py  
Que.4  
~$(CA1) Report( Saks...  
~WRL0003.tmp  
AP(CA1) Report( Saks...  
sakshi20079303.sln  
OUTLINE  
TIMELINE  
MYSQL  
SOLUTION EXPLORER  
PS C:\Users\Admin\Desktop\sakshi20079303> cd C:\Users\Admin\Desktop\sakshi20079303\Que.3  
PS C:\Users\Admin\Desktop\sakshi20079303\Que.3> python Que3_server.py  
Server started.... waiting for client....  
Client Connected  
Client Connected  
PS C:\Users\Admin\Desktop\sakshi20079303> cd C:\Users\Admin\Desktop\sakshi20079303\Que.3  
PS C:\Users\Admin\Desktop\sakshi20079303\Que.3> python Que3_client.py  
Enter Your Details :  
Name: shinde  
Address :Ireland  
Qualification :Diploma  
Course: Cyber security  
Start Year: 2021  
Start Month: aug  
Your Application Number Is : DBS5725  
PS C:\Users\Admin\Desktop\sakshi20079303\Que.3> python Que3_client.py  
Enter Your Details :  
Name: hardik  
Address :mumbai  
Qualification :bca  
Course: msc  
Start Year: 2022  
Start Month: oct  
Your Application Number Is : DBS1643  
PS C:\Users\Admin\Desktop\sakshi20079303\Que.3>
```

The screenshot shows a dark-themed SQLite viewer application window. At the top, there's a menu bar with File, Edit, Selection, View, and other options. A search bar contains the text "sakshi20079303". The main area displays a database named "Admission.db". Under "Tables", there is one entry: "students". The "students" table has the following schema:

	Application ID	Name	Address	Qualification	Course	Year	Month
1	DBS6026	sakshi	pune	BCA	cyber security	2025	sep
2	DBS8901	tanvi	mumbai	bca	AI	2025	02
3	DBS8739	shinde	ireland	diploma	cyber	2021	03
4	DBS5725	shinde	Ireland	Diploma	Cyber securtiy	2021	aug
5	DBS1643	hardik	mumbai	bca	msc	2022	oct
6							

At the bottom, the footer shows "SQLITE VIE... v0.10.6" and "Try Hydejack, a Jekyll theme for hackers, nerds, and academics".

3.2 Web Scraping and CSV Handling

I created two sample **Hotel Webpages** (**hotelA.html** and **hotelB.html**) containing the names of rooms and prices since scraping of actual hotel websites is not possible. Then I wrote Python program which scraped these two pages with **BeautifulSoup** and collected the room data which was then stored in a CSV file and finally read the CSV file to display the results. This also enabled me to show safe and legal web scraping.

The screenshot shows a Microsoft Visual Studio Code (VS Code) interface. The left sidebar contains icons for Explorer, Search, Problems, and other extensions like MySQL and Solution Explorer. The main area has a search bar at the top with the text "sakshi20079303". Below the search bar is a tab bar with "EXPLORER", "SAKSHI20079303", "webScraping.py X", and "Que.4 > webScraping.py > ...". The "TERMINAL" tab is selected, showing the following terminal output:

```
PS C:\Users\Admin\Desktop\sakshi20079303> cd C:\Users\Admin\Desktop\sakshi20079303\Que.4
PS C:\Users\Admin\Desktop\sakshi20079303\Que.4> python webScraping.py
Scraping Hotel A...
Scraping Hotel B...
Saving to csv...
CSV saved as HotelRooms.csv

Reading csv data:

{'Hotel': 'Hotel A', 'Room': 'Deluxe Sea View', 'Price': '$120'}
{'Hotel': 'Hotel A', 'Room': 'Luxury Suite', 'Price': '$199'}
{'Hotel': 'Hotel A', 'Room': 'Family Room', 'Price': '$150'}
{'Hotel': 'Hotel B', 'Room': 'Standard Room', 'Price': '$89'}
{'Hotel': 'Hotel B', 'Room': 'Premium King', 'Price': '$140'}
{'Hotel': 'Hotel B', 'Room': 'Executive Suite', 'Price': '$210'}
```

At the bottom of the terminal, it says "Done!" followed by the command "PS C:\Users\Admin\Desktop\sakshi20079303\Que.4>".

The screenshot shows the Microsoft Visual Studio Code interface. The Explorer sidebar on the left lists files and folders, including a folder named 'SAKS...' containing 'Que.1', 'Que.2', 'Que.3', and 'Que.4' (which further contains 'hotelA.html' and 'hotelB.html'). A file named 'HotelRooms.csv' is selected in the Explorer. The main editor area displays the contents of 'HotelRooms.csv':

```
1 Hotel,Room,Price
2 Hotel A,Deluxe Sea View,$120
3 Hotel A,Luxury Suite,$199
4 Hotel A,Family Room,$150
5 Hotel B,Standard Room,$89
6 Hotel B,Premium King,$140
7 Hotel B,Executive Suite,$210
8
```

The status bar at the bottom shows 'Projects: 1' and the solution file 'sakshi20079303.sln' is open.

4. CONCLUSION

I was able to practice C# and Python by creating basic applications for this assignment. I learned how to use OOP , data management , database administration , and client-server communication.

5. REFERENCE

- a) **Microsoft. (2024)** *C# Documentation: Classes and Objects*. Available at:
<https://learn.microsoft.com/en-us/dotnet/csharp/fundamentals/object-oriented/classes>
- b) **Python Software Foundation. (2024)** *socket — Low-level networking interface*. Available at: <https://docs.python.org/3/library/socket.html>
- c) **SQLite. (2024)** *SQLite Language Documentation*. Available at:
<https://www.sqlite.org/lang.html>

6. GITHUB REPOSITORY LINK

<https://github.com/sakshi220122/Sakshi20079303-AdvProgrammingCA1-.git>