UNIT-5

1 MARK Q&A

a) List any four file operations in Python.

1. Open a file – open() function is used to open a file.
2. Read from a file – read() or readline() methods are used to read data from a file.
3. Write to a file – write() or writelines() methods are used to write data to a file.
4. Close a file – close() method is used to close the file after operations.

b) Write syntax for opening a file in Python in read-only mode.

```
Syntax:
 file = open("filename.txt", "r")
```

c) Define Tkinter in Python.

Tkinter is the standard GUI (Graphical User Interface) library in Python, used to create desktop applications. It provides various widgets and controls such as buttons, labels, text boxes, etc.

d) List the different geometry managers available in Python.

1. **pack()** – Organizes widgets in blocks before placing them in the parent widget.
2. **grid()** – Organizes widgets in a tabular (row-column) structure.
3. **place()** – Places widgets at an absolute position using x and y coordinates.

e) State how to import Tkinter in a Python program.

```
import tkinter as tk
```

3 MARKS Q&A

a) Describe the process of creating a Label widget in Python with an example.

In **Tkinter**, a **Label** widget is used to display text or images in a window. The Label() function is used to create a label with specified options such as text, font, and background color.

Syntax:

```
Label(parent, text="text", options...)
```

Ex:

```
import tkinter as tk

root = tk.Tk()
root.title("Label Example")

# Creating a label
label1 = tk.Label(root, text="Hello, Tkinter!", font=("Arial", 16), bg="yellow")
label1.pack(pady=10)

root.mainloop()
```

b) Write short notes on text files and binary files.

1. **Text Files:**
   - Stores data in human-readable format.
   - Each line ends with a newline character (\n).
   - Commonly used for storing textual information such as .txt, .csv, etc.

2. **Binary Files:**

- Stores data in a machine-readable format (0s and 1s).
- Cannot be read directly in text format.
- Used for storing images, audio, video, and other multimedia.
- Example: .jpg, .mp4, .exe, etc.

c) List the various ways to read a file in Python.

1. **read()** – Reads the entire content of the file as a string.

Ex:

```
file.read()
```

2. **readline()** – Reads one line from the file.

Ex:

```
file.readline()
```

3. **readlines()** – Reads all lines and returns them as a list.

Ex:

```
file.readlines()
```

4. **for loop** – Iterates through each line in the file.

Ex:

```
for line in file:
   print(line)
```

d) Explain how to open a text file in Python and list a few access modes available for files.

To **open a text file** in Python, use the open() function.

Syntax:

file_object = open("filename.txt", mode)

Ex:

file = open("example.txt", "r")

content = file.read()

print(content)

file.close()

Common Access Modes:

1. "r" – Read mode (default), opens the file for reading.
2. "w" – Write mode, creates a new file or truncates an existing file.
3. "a" – Append mode, adds content to the end of the file.
4. "rb" / "wb" – Read/Write in binary mode.

e) Differentiate between readline() and readlines() in Python.

Feature	**readline()**	**readlines()**
Functionality	Reads a single line from the file.	Reads all lines and returns a list.
Return Type	Returns a string.	Returns a list of strings.
Usage	Ideal for reading one line at a time.	Ideal for reading entire file into memory.
Example	file.readline()	file.readlines()
Efficiency	More memory-efficient for large files.	Less memory-efficient for large files.

5 MARKS Q&A

a) Write a Python program to copy the contents of one file to another.

```
# Open the source file in read mode
with open("source.txt", "r") as source_file:
    # Read the content of the source file
    content = source_file.read()

# Open the destination file in write mode
with open("destination.txt", "w") as destination_file:
    # Write the content to the destination file
    destination_file.write(content)

print("File copied successfully!")
```

b) Explain in detail about File built-in methods.

Python provides several built-in methods to manipulate files. Below are some commonly used methods:

1. open()

- Opens a file and returns a file object.
- Syntax:

```
file = open("filename.txt", "r")
```

---

2. read()

- Reads the entire content of a file.
- Syntax:

```
content = file.read()
```

---

3. readline()

- Reads a single line from the file.
- Syntax:

```
line = file.readline()
```

---

4. readlines()

- Reads all lines from a file and returns them as a list.
- Syntax:

```
lines = file.readlines()
```

---

5. write()

- Writes data to the file.
- Syntax:

```
file.write("Hello, World!")
```

c) Explain about Radiobutton widget in Tkinter. Demonstrate how to create two radiobutton sets (one for gender and another for Indian or not) on the same canvas.

Radiobutton Widget in Tkinter:

- A **Radiobutton** allows the user to select one option from a set of options.
- Radiobuttons are associated with a variable that holds the selected value.

Ex:

```
import tkinter as tk


def show_choice():
    print("Gender:", gender.get(), "Indian:", indian.get())

root = tk.Tk()

gender = tk.StringVar(value="Unknown")
tk.Label(root, text="Gender:").pack()
tk.Radiobutton(root, text="Male", variable=gender, value="Male").pack()
tk.Radiobutton(root, text="Female", variable=gender, value="Female").pack()

indian = tk.StringVar(value="No")
tk.Label(root, text="Indian:").pack()
tk.Radiobutton(root, text="Yes", variable=indian, value="Yes").pack()
tk.Radiobutton(root, text="No", variable=indian, value="No").pack()

tk.Button(root, text="Show", command=show_choice).pack()

root.mainloop()
```

d) Write a Python program to count the number of lines in a file.

```
# Open the file in read mode
with open("example.txt", "r") as file:
    # Read all lines and count them
    line_count = len(file.readlines())

print(f"Number of lines in the file: {line_count}")
```

e) Write short notes on any four file operations in Python with an example.

1. Open a File (open())

- Opens a file in a specified mode.
- Modes include:
    - "r" – Read
    - "w" – Write
    - "a" – Append
- Example:

```
file = open("example.txt", "r")
```

---

2. Read from a File (read())

- Reads the content of a file.
- Example:

```
file = open("example.txt", "r")
content = file.read()
print(content)
file.close()
```

---

3. Write to a File (write())

- Writes data to a file. If the file doesn’t exist, it creates a new file.
- Example:

```
file = open("example.txt", "w")
file.write("Hello, World!")
file.close()
```

---

4. Append to a File (a)

- Adds new content to the end of the file without deleting the existing data.
- Example:

```
file = open("example.txt", "a")
file.write("\nNew content added.")
file.close()
```

10 MARKS Q&A

a) Describe in detail about Tkinter with an example of three layout managers. ✅ What is Tkinter?

- Tkinter is the standard Python library used to create Graphical User Interface (GUI) applications.
- It provides various widgets such as buttons, labels, text boxes, etc., and controls user interaction.

✅ Three Layout Managers in Tkinter:

1. **pack()**
   - Organizes widgets in a block before placing them in the parent widget.
   - Automatically adjusts the size based on the content.
   - Example:

```
import tkinter as tk

root = tk.Tk()
root.title("pack() Example")
```

```
tk.Label(root, text="Top Label").pack(side="top")
tk.Label(root, text="Bottom Label").pack(side="bottom")
tk.Label(root, text="Left Label").pack(side="left")
tk.Label(root, text="Right Label").pack(side="right")

root.mainloop()
```

---

2. **grid()**
   - Organizes widgets in a **row-column** structure.
   - Ideal for creating forms and structured layouts.
   - Example:

```
import tkinter as tk

root = tk.Tk()
root.title("grid() Example")

tk.Label(root, text="Name:").grid(row=0, column=0)
tk.Entry(root).grid(row=0, column=1)

tk.Label(root, text="Age:").grid(row=1, column=0)
tk.Entry(root).grid(row=1, column=1)

tk.Button(root, text="Submit").grid(row=2, column=1)
root.mainloop()
```

---

3. **place()**
   - Positions widgets at an **exact coordinate (x, y)**.

    - Gives absolute control over widget placement.
    - Example:

```
import tkinter as tk


root = tk.Tk()
root.title("place() Example")


tk.Label(root, text="Label at (50, 50)").place(x=50, y=50)
tk.Button(root, text="Click Me!").place(x=100, y=100)


root.mainloop()
```

---

b) Explain the following file built-in functions and methods with clear syntax, description, and illustration.

---

1. open()

- **Description:** Opens a file and returns a file object.
- **Syntax:**

```
file_object = open("filename.txt", mode)
```

- **Modes:**
    - "r" – Read mode.
    - "w" – Write mode.
    - "a" – Append mode.
- **Example:**

```
file = open("example.txt", "r")
content = file.read()
```

```
print(content)

file.close()
```

---

2. file()

✅ In Python 3.x, file() is **not available**. Use open() instead.
✅ In Python 2.x:

```
file_object = file("filename.txt", "r")
```

---

3. seek()

- **Description:** Moves the file pointer to a specified position.
- **Syntax:**

```
file.seek(offset, from_what)
```

- **Example:**

```
file = open("example.txt", "r")

file.seek(5)

content = file.read()

print(content)

file.close()
```

---

4. tell()

- **Description:** Returns the current file pointer position.
- **Syntax:**

```
position = file.tell()
```

- **Example:**

```
file = open("example.txt", "r")

file.read(10)

print("Position:", file.tell())
```

```
file.close()
```

---

5. read()

- **Description:** Reads the entire content of the file or specified number of bytes.
- **Syntax:**

```
content = file.read(size)
```

- **Example:**

```
file = open("example.txt", "r")
content = file.read(10)
print(content)
file.close()
```

c) Write a Python program to design a GUI-based student registration form.

```
import tkinter as tk
from tkinter import messagebox


def submit_form():
    name = entry_name.get()
    age = entry_age.get()


    if name and age:
        messagebox.showinfo("Success", f"Student Registered
Successfully!\n\nName: {name}\nAge: {age}")
    else:
        messagebox.showwarning("Error", "Please enter both Name and Age!")
```

```
# Create main window

root = tk.Tk()

root.title("Student Registration")

root.geometry("250x200")


# Name Label and Entry

tk.Label(root, text="Name:").pack(pady=5)

entry_name = tk.Entry(root)

entry_name.pack(pady=5)


# Age Label and Entry

tk.Label(root, text="Age:").pack(pady=5)

entry_age = tk.Entry(root)

entry_age.pack(pady=5)


# Submit Button

tk.Button(root, text="Submit", command=submit_form).pack(pady=10)


# Run the application

root.mainloop()
```