

## # Part-1

### # Task1

```
import os
```

```
filename = "diary.txt"
```

```
if not os.path.exists(filename):
```

```
    with open(filename, "w") as f:
```

```
        f.write("Today was productive.\n")
```

```
        f.write("Learned Python file handling.\n")
```

```
        f.write("Worked on internship tasks.\n")
```

```
        f.write("Improved coding skills.\n")
```

```
        f.write("Feeling motivated!\n")
```

```
with open(filename, "r") as f:
```

```
    content = f.read()
```

```
    print("\nFull Diary:\n", content)
```

```
print("\nFirst 50 characters:\n", content[:50])
```

```
print("\nDiary Line by Line:")
```

```
with open(filename, "r") as f:
```

```
    for line in f:
```

```
        print(line.strip())
```

```
os.remove(filename)
```

```
print("\nDiary file deleted.")
```

### # Task2

```
mobile = input("Enter Mobile Number: ")
```

```
budget = int(input("Enter Budget: "))
```

```
if budget == 199:
```

```
    print("Plan: 28 Days Validity")
```

```
elif budget == 299:
```

```
print("Plan: 56 Days Validity")
elif budget == 499:
    print("Plan: 84 Days Validity")
else:
    print("No recommended plan available.")
```

### # Task3

```
cloth_type = input("Enter type (Cotton/Wool/Synthetic): ")
num = int(input("Enter number of clothes: "))
weight = float(input("Enter weight per cloth (kg): "))
```

```
total = num * weight
```

```
if total <= 7:
    status = "Wash Started"
else:
    status = "Overload! Reduce clothes"
```

```
print("\nCloth Type:", cloth_type)
print("Total Load:", total, "kg")
print("Status:", status)
```

### # Task4

```
class BusTicket:
    def __init__(self, name, bus_no, seat):
        self.name = name
        self.bus_no = bus_no
        self.seat = seat
    def display(self):
        print(f"Passenger: {self.name}, Bus: {self.bus_no}, Seat: {self.seat}")
t1 = BusTicket("Asha", "KA01", 12)
t2 = BusTicket("Ravi", "KA02", 15)

t1.display()
t2.display()
```

## # Task5

```
from abc import ABC, abstractmethod
```

```
class Mobile(ABC):
```

```
    def __init__(self, pin):
```

```
        self.__pin = pin
```

```
    def verify_pin(self, entered):
```

```
        if entered == self.__pin:
```

```
            print("PIN Verified")
```

```
        else:
```

```
            print("Incorrect PIN")
```

```
    def change_pin(self, old, new):
```

```
        if old == self.__pin:
```

```
            self.__pin = new
```

```
            print("PIN Changed Successfully")
```

```
        else:
```

```
            print("Wrong Old PIN")
```

```
@abstractmethod
```

```
    def device_info(self):
```

```
        pass
```

```
class SmartPhone(Mobile):
```

```
    def device_info(self):
```

```
        print("SmartPhone Device")
```

```
phone = SmartPhone(1234)
```

```
phone.verify_pin(1234)
```

```
phone.change_pin(1234, 5678)
```

## # Task6

```
import datetime, math, os

name = input("Customer Name: ")
city = input("Destination City: ")
travelers = int(input("Number of Travelers: "))
cost = float(input("Cost per Ticket: "))

now = datetime.datetime.now()

total = travelers * cost
final = total + (0.05 * total)
final = math.ceil(final)

filename = "booking.txt"

if not os.path.exists(filename):
    with open(filename, "w") as f:
        f.write(f"Customer: {name}\nCity: {city}\nTotal Cost: {final}")

print("Booking Time:", now)
print("Final Cost:", final)
print("File Created Successfully")

# Task7
```

```
import datetime, sys, os

name = input("Patient Name: ")
age = int(input("Age: "))
temp = float(input("Body Temperature: "))
heart = int(input("Heart Rate: "))

status = "Fever Detected" if temp > 37.5 else "Normal"

filename = "health_report.txt"
```

```
with open(filename, "w") as f:  
    f.write(f"Patient: {name}\nStatus: {status}")  
  
print("Checkup Time:", datetime.datetime.now())  
print("Python Version:", sys.version)  
print("Report Generated Successfully")
```

## # Task8

```
import datetime, math, os  
  
temp = []  
for i in range(5):  
    t = float(input("Enter temp for day {i+1}: "))  
    temp.append(t)  
  
avg = math.ceil(sum(temp)/len(temp))  
max_t = max(temp)  
min_t = min(temp)  
  
if avg >= 35:  
    status = "Hot Week"  
elif 20 <= avg < 35:  
    status = "Pleasant Week"  
else:  
    status = "Cold Week"
```

```
with open("weather_report.txt", "w") as f:  
    f.write(f"Average: {avg}\nStatus: {status}")  
  
print("Report Date:", datetime.date.today())  
print("Max:", max_t, "Min:", min_t, "Avg:", avg)
```

## # Task9

```
import tkinter as tk
```

```
def calculate(op):
```

```
    n1 = float(e1.get())
```

```
    n2 = float(e2.get())
```

```
    if op == "add":
```

```
        result = n1 + n2
```

```
    elif op == "sub":
```

```
        result = n1 - n2
```

```
    elif op == "mul":
```

```
        result = n1 * n2
```

```
    else:
```

```
        result = n1 / n2
```

```
lbl.config(text="Result: " + str(result))
```

```
root = tk.Tk()
```

```
e1 = tk.Entry(root)
```

```
e2 = tk.Entry(root)
```

```
e1.pack()
```

```
e2.pack()
```

```
tk.Button(root, text="Add", command=lambda: calculate("add")).pack()
```

```
tk.Button(root, text="Subtract", command=lambda: calculate("sub")).pack()
```

```
tk.Button(root, text="Multiply", command=lambda: calculate("mul")).pack()
```

```
tk.Button(root, text="Divide", command=lambda: calculate("div")).pack()
```

```
lbl = tk.Label(root)
```

```
lbl.pack()
```

```
root.mainloop()
```

## # Part-2

### # Task10

```
import datetime, random, math, os, sys

now = datetime.datetime.now()
print("Date:", now.date())
print("Time:", now.strftime("%I:%M %p"))
print("Day:", now.strftime("%A"))

roads = ["Road A", "Road B", "Road C"]
green = random.choice(roads)
duration = random.randint(30, 90)
print("\nTraffic Update:")
print("Green Signal →", green)
print("Duration →", duration, "seconds")

units = [120.5, 130.2, 125.3, 136.8]
total = sum(units)
avg = math.ceil(total/len(units))
print("\nEnergy Report:")
print("Total Units Used:", total)
print("Average Units:", avg)

file = "complaints.txt"
exists = os.path.exists(file)
print("\nComplaint File Exists:", "Yes" if exists else "No")

print("\nPython Version:", sys.version)
print("System Report Generated Successfully!")
```

### # Task11

```
import numpy as np
import pandas as pd

marks = np.array([85, 90, 78, 90, 85, 88])
```

```
arr2d = marks.reshape(3,2)
print("2D Array:\n", arr2d)

df = pd.DataFrame(arr2d, columns=["Maths", "Science"])

df.loc[1, "Science"] = np.nan

df["Science"] = df["Science"].fillna(df["Science"].mean())

df = df.drop_duplicates()

print("\nCleaned DataFrame:")
print(df)

print("\nSummary Statistics:")
print(df.describe())
```

### Outputs:

```
● PS D:\Internship\Day18> python task1.py

Full Diary:
Today was productive.
Learned Python file handling.
Worked on internship tasks.
Improved coding skills.
Feeling motivated!

First 50 characters:
Today was productive.
Learned Python file handling

Diary Line by Line:
Today was productive.
Learned Python file handling.
Worked on internship tasks.
Improved coding skills.
Feeling motivated!

Diary file deleted.
```

```
● PS D:\Internship\Day18> python task2.py
Enter Mobile Number: 7795131735
Enter Budget: 499
Plan: 84 Days Validity
```

```
● PS D:\Internship\Day18> python task3.py
Enter type (Cotton/Wool/Synthetic): Cotton
Enter number of clothes: 5
Enter weight per cloth (kg): 1
```

```
Cloth Type: Cotton
Total Load: 5.0 kg
Status: Wash Started
```

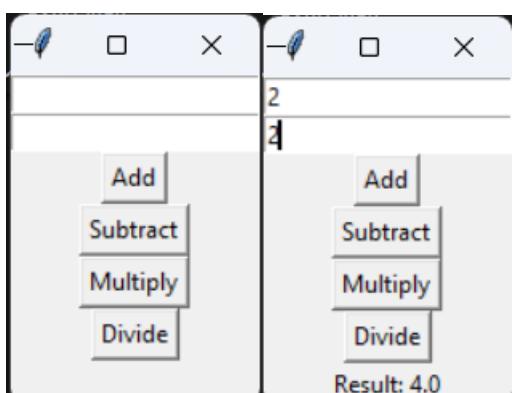
```
● PS D:\Internship\Day18> python task4.py
Passenger: Asha, Bus: KA01, Seat: 12
Passenger: Ravi, Bus: KA02, Seat: 15
```

```
● PS D:\Internship\Day18> python task5.py
PIN Verified
PIN Changed Successfully
```

```
● PS D:\Internship\Day18> python task6.py
Customer Name: NS
Destination City: Bangalore
Number of Travelers: 2
Cost per Ticket: 1000
Booking Time: 2026-02-21 16:31:03.453356
Final Cost: 2100
File Created Successfully
```

```
● PS D:\Internship\Day18> python task7.py
Patient Name: ABC
Age: 20
Body Temperature: 35
Heart Rate: 65
Checkup Time: 2026-02-21 18:23:55.395080
Python Version: 3.10.0 (tags/v3.10.0:b494f59, Oct 4 2021, 19:00:18) [MSC v.1929 64 bit (AMD64)]
Report Generated Successfully
```

```
● PS D:\Internship\Day18> python task8.py
Enter temp for day 1: 35
Enter temp for day 2: 34
Enter temp for day 3: 30
Enter temp for day 4: 31
Enter temp for day 5: 25
Report Date: 2026-02-21
Max: 35.0 Min: 25.0 Avg: 31
```



```
● PS D:\Internship\Day18> python task10.py
```

Date: 2026-02-21

Time: 06:27 PM

Day: Saturday

Traffic Update:

Green Signal → Road C

Duration → 36 seconds

Energy Report:

Total Units Used: 512.8

Average Units: 129

Complaint File Exists: No

Python Version: 3.10.0 (tags/v3.10.0:b494f59, Oct 4 2021, 19:00:18) [MSC v.1929 64 bit (AMD64)]

System Report Generated Successfully!

```
● PS D:\Internship\Day18> python task11.py
```

2D Array:

[[85 90]

[78 90]

[85 88]]

Cleaned DataFrame:

	Maths	Science
--	-------	---------

0	85	90.0
---	----	------

1	78	89.0
---	----	------

2	85	88.0
---	----	------

Summary Statistics:

	Maths	Science
--	-------	---------

count	3.000000	3.0
-------	----------	-----

mean	82.666667	89.0
------	-----------	------

std	4.041452	1.0
-----	----------	-----

min	78.000000	88.0
-----	-----------	------

25%	81.500000	88.5
-----	-----------	------

50%	85.000000	89.0
-----	-----------	------

75%	85.000000	89.5
-----	-----------	------

max	85.000000	90.0
-----	-----------	------