11. ZIP OPERATION ON FOLDER

```
import shutil

# 1. Backup photos folder
shutil.make_archive("photos_backup", 'zip', "photos")

# 2. Zip project folder
shutil.make_archive("project_zip", 'zip', "project")

# 3. Zip lecture notes
shutil.make_archive("lecture_notes", 'zip', "lecture_notes")
```

✅ Creates .zip files automatically.

12. INHERITANCE – AREAS

```
import math

class Shape:
    def area(self):
        pass

class Triangle(Shape):
    def __init__(self, base, height):
        self.base = base
        self.height = height
    def area(self):
        return 0.5 * self.base * self.height

class Circle(Shape):
    def __init__(self, radius):
        self.radius = radius
    def area(self):
        return math.pi * self.radius ** 2

class Rectangle(Shape):
    def __init__(self, length, width):
        self.length = length
        self.width = width
    def area(self):
        return self.length * self.width

# Example
```

```
shapes = [Triangle(10, 5), Circle(7), Rectangle(8, 6)]
for s in shapes:
    print(f"{s.__class__.__name__} Area:", s.area())
```

13. EMPLOYEE DETAILS

```
employees = [
    {"name": "Ravi", "dept": "Sales", "salary": 30000},
    {"name": "Anita", "dept": "IT", "salary": 40000},
    {"name": "Kiran", "dept": "HR", "salary": 35000}
]

# 1. Update Sales salaries by 10%
for emp in employees:
    if emp["dept"] == "Sales":
        emp["salary"] *= 1.10

# 2. Update IT salaries by 5%
for emp in employees:
    if emp["dept"] == "IT":
        emp["salary"] *= 1.05

# 3. Update HR salaries by 8% and print
for emp in employees:
    if emp["dept"] == "HR":
        emp["salary"] *= 1.08
    print(emp)
```

14. POLYMORPHISM & INHERITANCE – PALINDROME

```
class Palindrome:
    def check(self, value):
        str_val = str(value)
        return str_val == str_val[::-1]

class WordPalindrome(Palindrome):
    pass

class NumberPalindrome(Palindrome):
    pass

# Example
print("Word:", WordPalindrome().check("madam"))
print("Number:", NumberPalindrome().check(121))
print("Date:", Palindrome().check("20200202"))
```

15. SPREADSHEET OPERATIONS (USING OPENPYXL)

```
from openpyxl import Workbook, load_workbook

# 1. Read marks & write average
wb = Workbook()
ws = wb.active
ws.append(["Name", "Mark1", "Mark2", "Mark3"])
ws.append(["Ravi", 80, 90, 85])
ws.append(["Anita", 70, 75, 80])

for row in ws.iter_rows(min_row=2, max_col=4):
    marks = [cell.value for cell in row[1:]]
    avg = sum(marks) / len(marks)
    ws.cell(row=row[0].row, column=5, value=avg)

wb.save("marks.xlsx")

# 2. Store sales and update totals
wb = Workbook()
ws = wb.active
ws.append(["Product", "Qty", "Price", "Total"])
ws.append(["Laptop", 2, 50000, None])
ws.append(["Mouse", 5, 500, None])

for row in ws.iter_rows(min_row=2, max_col=4):
    qty, price = row[1].value, row[2].value
    ws.cell(row=row[0].row, column=4, value=qty * price)

wb.save("sales.xlsx")

# 3. Copy sheet
wb = Workbook()
ws1 = wb.active
ws1.title = "Original"
ws1.append(["A", "B", "C"])

ws2 = wb.create_sheet("Copy")
for row in ws1.iter_rows(values_only=True):
    ws2.append(row)

wb.save("copy.xlsx")
```

16. MERGE SELECTED PAGES FROM PDFS (USING PYPDF2)

```
from PyPDF2 import PdfReader, PdfWriter

def merge_selected_pages(files, pages, output):
    writer = PdfWriter()
    for file in files:
        reader = PdfReader(file)
        for p in pages:
            if p-1 < len(reader.pages):
                writer.add_page(reader.pages[p-1])
    with open(output, "wb") as f:
        writer.write(f)

# Example
merge_selected_pages(["file1.pdf", "file2.pdf"], [1, 3], "merged.pdf")
```

17. FETCH WEATHER DATA FROM JSON

```
import json

# Sample JSON data
weather_json = '''
{
  "city": "Chennai",
  "current": {"temp": 42, "humidity": 70},
  "forecast": [
    {"day": "Mon", "condition": "Rain", "temp": 34},
    {"day": "Tue", "condition": "Sunny", "temp": 36},
    {"day": "Wed", "condition": "Rain", "temp": 32}
  ]
}
'''

data = json.loads(weather_json)

# 1. Print current
print("Temp:", data["current"]["temp"], "°C")
print("Humidity:", data["current"]["humidity"], "%")

# 2. Rainy days
rainy = [d["day"] for d in data["forecast"] if d["condition"] == "Rain"]
print("Rainy Days:", rainy)

# 3. Alert if temp > 40
```

```
if data["current"]["temp"] > 40:
    print(" ALERT: High temperature!")
```