TECHNIK NEST

INNOVATIVE MINDS, NESTING SUCCESS

Name: Hassan Ali

Intern ID: TN/IN02/PY/012

Email ID: **hassanali2127294@gmail.com**

Task Week: 2

Internship Domain: Python Development

Instructor Name: Mr. Hassan Ali

Question: 1. Store 5 student names & print each.

Code:

```
students = ["Ali", "Hassan", "Sara", "Ayesha", "Bilal"]
for name in students:
    print(name)
```

Output:

```
Ali
Hassan
Sara
Ayesha
Bilal
```

Question: 2. Reverse list without reverse().

Code:

```
students = ["Ali", "Hassan", "Sara", "Ayesha", "Bilal"]
reversed_students = students[::-1]
print(reversed_students)
```

Output:

```
['Bilal', 'Ayesha', 'Sara', 'Hassan', 'Ali']
```

Question: 3. Store 3 coordinates & unpack.

Code:

```
coords = (10, 20, 30)
x, y, z = coords
```

```
print(x, y, z)
```

Output:

```
10 20 30
```

Question: 3. Swap vars using tuple assignment.

Code:

```
a, b = 5, 10
a, b = b, a
print(a, b)
```

Output:

```
10 5
```

Question: 4. Remove duplicates from list.

Code:

```
nums = [1, 2, 2, 3, 4, 4, 5]
unique_nums = list(set(nums))
print(unique_nums)
```

Output:

```
[1, 2, 3, 4, 5]
```

Question: 5. Find intersection of two sets.

Code:

```
set1 = {1, 2, 3, 4}
```

```
set2 = {3, 4, 5, 6}
intersection = set1 & set2
print(intersection)
```

Output:

```
{3, 4}
```

Question: 6. Student record CRUD in dict.

Code:

```
student = {'name': 'Hassan', 'age': 22, 'grade': 'A'}
student['subject'] = 'Math'  # Create
print(student)               # Read
student['grade'] = 'A+'      # Update
del student['subject']       # Delete
print(student)
```

Output:

```
{'name': 'Hassan', 'age': 22, 'grade': 'A', 'subject': 'Math'}
{'name': 'Hassan', 'age': 22, 'grade': 'A+'}
```

Question: 7. Count word frequency in sentence.

Code:

```
sentence = "python is fun and python is powerful"
word_freq = {}
for word in sentence.split():
    word_freq[word] = word_freq.get(word, 0) + 1
print(word_freq)
```

Output:

```
{'python': 2, 'is': 2, 'fun': 1, 'and': 1, 'powerful': 1}
```

Question: 8. Write calc(a,b,op).

Code:

```
def calc(a, b, op):
    if op == '+': return a + b
    elif op == '-': return a - b
    elif op == '*': return a * b
    elif op == '/': return a / b if b != 0 else None
    else: return None

print(calc(5, 3, '+'))
print(calc(5, 3, '-'))
print(calc(5, 3, '*'))
print(calc(5, 0, '/'))
```

Output:

```
8
2
15
None
```

Question: 9. Write factorial(n) recursive.

Code:

```
def factorial(n):
    if n < 0:
        raise ValueError("n must be non-negative")
    if n in (0, 1):
        return 1
    return n * factorial(n-1)

print(factorial(5))
```

Output:

```
120
```

Question: 10. Use random & datetime in script.

Code:

```
import random
from datetime import datetime

print(random.randint(1, 100))
print(datetime.now())
```

Output:

```
57
2025-08-14 04:55:23.612345   # Values will vary
```

Question: 11. Create math_utils module & import.

Code:

```
# math_utils.py
def add(a, b): return a + b
def sub(a, b): return a - b
def mul(a, b): return a * b
def div(a, b): return a / b if b != 0 else None

# main.py
import math_utils
print(math_utils.add(2, 3))
print(math_utils.sub(7, 4))
print(math_utils.mul(3, 5))
print(math_utils.div(10, 2))
print(math_utils.div(10, 0))
```

Output:

```
5
3
15
5.0
None
```

Question: 12. Safe int input loop.

Code:

```
while True:
    try:
        num = int(input("Enter an integer: "))
        print(f"You entered: {num}")
        break
    except ValueError:
        print("Invalid input, please try again.")
```

Output:

```
Enter an integer: hello
Invalid input, please try again.
Enter an integer: 42
You entered: 42
```

Question: 13. File open with error message.

Code:

```
try:
    with open("file.txt", "r") as f:
        data = f.read()
    print(data)
except FileNotFoundError:
    print("File not found.")
```

Output:

```
File not found.
```

Question 14: Phonebook App: CRUD contacts dict <-> JSON file storage

Code:

```
import json

def load_contacts(filename="contacts.json"):
    try:
        with open(filename, "r") as f:
            return json.load(f)
    except (FileNotFoundError, json.JSONDecodeError):
        return {}

def save_contacts(contacts, filename="contacts.json"):
    with open(filename, "w") as f:
        json.dump(contacts, f)

def add_contact(contacts, name, phone):
    contacts[name] = phone
    save_contacts(contacts)
    print(f"Contact '{name}' saved.")

def view_contacts(contacts):
    if not contacts:
        print("No contacts found.")
    else:
        for name, phone in contacts.items():
            print(f"{name}: {phone}")

def delete_contact(contacts, name):
    if name in contacts:
        del contacts[name]
        save_contacts(contacts)
        print(f"Contact '{name}' deleted.")
    else:
        print(f"Contact '{name}' not found.")

contacts = load_contacts()
add_contact(contacts, "Hassan", "12345")
add_contact(contacts, "Ali", "98765")
view_contacts(contacts)
delete_contact(contacts, "Ali")
view_contacts(contacts)
```

Output:

```
Contact 'Hassan' saved.
Contact 'Ali' saved.
Hassan: 12345
Ali: 98765
Contact 'Ali' deleted.
Hassan: 12345
```