Aim: Implement AES technique to encrypt and decrypt message, Use java or python code

CODE:

from Crypto.Cipher import AES

from Crypto.Random import get\_random\_bytes

import base64

IV\_LEN = 12  *#* 12-byte nonce for GCM

KEY\_LEN = 32 *#* 32 bytes = 256-bit key

def generate\_key():

    return get\_random\_bytes(KEY\_LEN)

def encrypt(plaintext: str, key: bytes) -> str:

    nonce = get\_random\_bytes(IV\_LEN)

    cipher = AES.new(key, AES.MODE\_GCM, nonce=nonce)

    ciphertext, tag = cipher.encrypt\_and\_digest(plaintext.encode("utf-8"))

*#* Pack as: nonce || tag || ciphertext (then base64)

    packed = nonce + tag + ciphertext

    return base64.b64encode(packed).decode("ascii")

def decrypt(b64\_packed: str, key: bytes) -> str:

    packed = base64.b64decode(b64\_packed)

    nonce = packed[:IV\_LEN]

    tag = packed[IV\_LEN:IV\_LEN+16]

    ciphertext = packed[IV\_LEN+16:]

    cipher = AES.new(key, AES.MODE\_GCM, nonce=nonce)

    plaintext = cipher.decrypt\_and\_verify(ciphertext, tag)

    return plaintext.decode("utf-8")

if \_\_name\_\_ == "\_\_main\_\_":

    key = generate\_key()

    msg = "Death is an old friend"

    b64 = encrypt(msg, key)

    print("Packed (base64 nonce+tag+ciphertext):", b64)

    print("Decrypted:", decrypt(b64, key))

OUPTUT:

