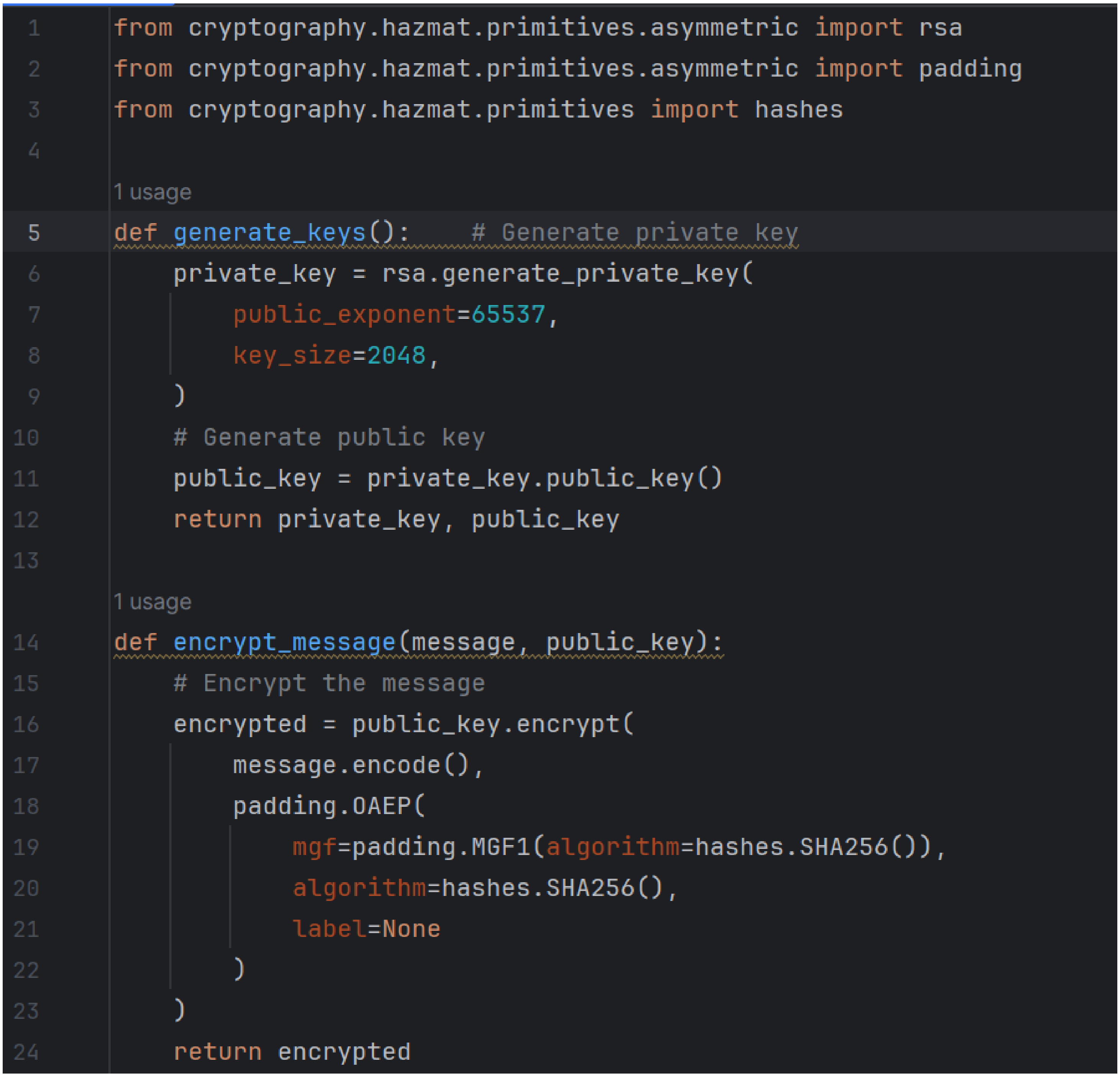
New Era University

College of Informatics and Computer Studies

Name: Manalili, James I.

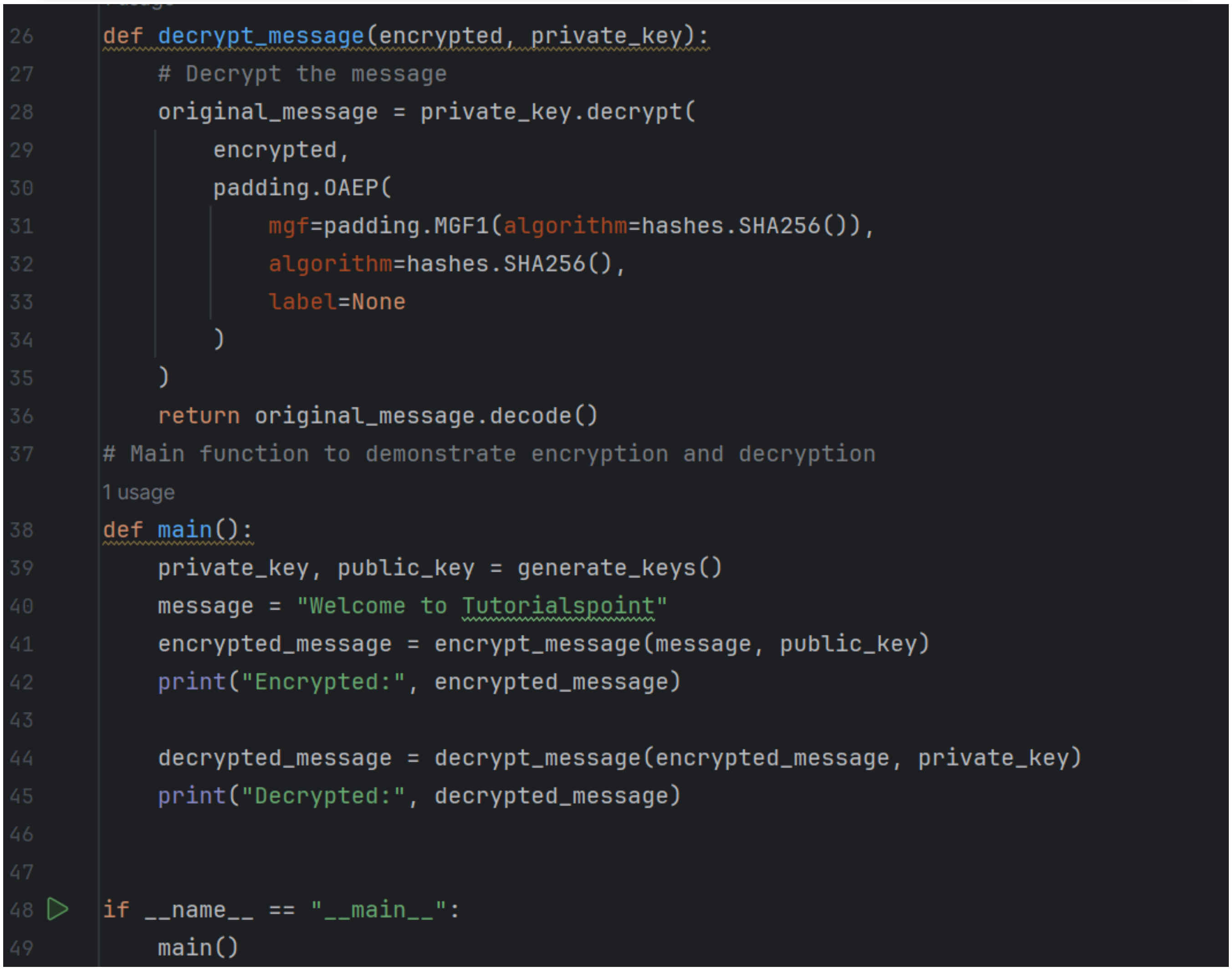
Section: 3 – BSCS – 1



```
from cryptography.hazmat.primitives.asymmetric import rsa
from cryptography.hazmat.primitives.asymmetric import padding
from cryptography.hazmat.primitives import hashes

1 usage
def generate_keys():     # Generate private key
    private_key = rsa.generate_private_key(
        public_exponent=65537,
        key_size=2048,
    )
    # Generate public key
    public_key = private_key.public_key()
    return private_key, public_key

1 usage
def encrypt_message(message, public_key):
    # Encrypt the message
    encrypted = public_key.encrypt(
        message.encode(),
        padding.OAEP(
            mgf=padding.MGF1(algorithm=hashes.SHA256()),
            algorithm=hashes.SHA256(),
            label=None
        )
    )
    return encrypted
```

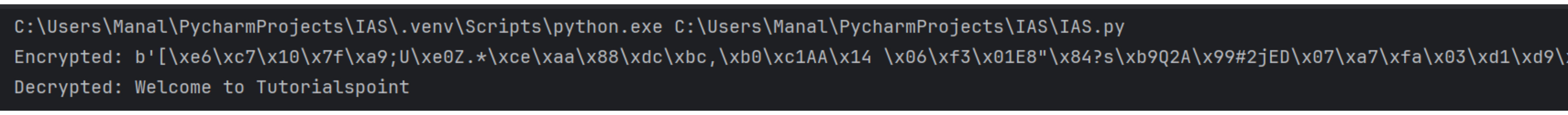


```
def decrypt_message(encrypted, private_key):
    # Decrypt the message
    original_message = private_key.decrypt(
        encrypted,
        padding.OAEP(
            mgf=padding.MGF1(algorithm=hashes.SHA256()),
            algorithm=hashes.SHA256(),
            label=None
        )
    )
    return original_message.decode()
# Main function to demonstrate encryption and decryption
1 usage
def main():
    private_key, public_key = generate_keys()
    message = "Welcome to Tutorialspoint"
    encrypted_message = encrypt_message(message, public_key)
    print("Encrypted:", encrypted_message)

    decrypted_message = decrypt_message(encrypted_message, private_key)
    print("Decrypted:", decrypted_message)


if __name__ == "__main__":
    main()
```

OUTPUT:



```
C:\Users\Manal\PycharmProjects\IAS\.venv\Scripts\python.exe C:\Users\Manal\PycharmProjects\IAS\IAS.py
Encrypted: b'[\xe6\xc7\x10\x7f\xa9;U\xe0Z.*\xce\xaa\x88\xdc\xbc,\xb0\xc1AA\x14 \x06\xf3\x01E8"\x84?s\xb9Q2A\x99#2jED\x07\xa7\xfa\x03\xd1\xd9\
Decrypted: Welcome to Tutorialspoint
```