Client.py

```
import xmlrpc.client

def main():
    # Connect to the server
    server = xmlrpc.client.ServerProxy("http://localhost:9000/")


    try:
          num = int(input("Enter an integer to calculate its factorial: "))
        # Call the remote function
        result = server.factorial(num)
        print(f"The factorial of {num} is: {result}")

    except Exception as e:
        print(f"Error: {e}")

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

Server.py

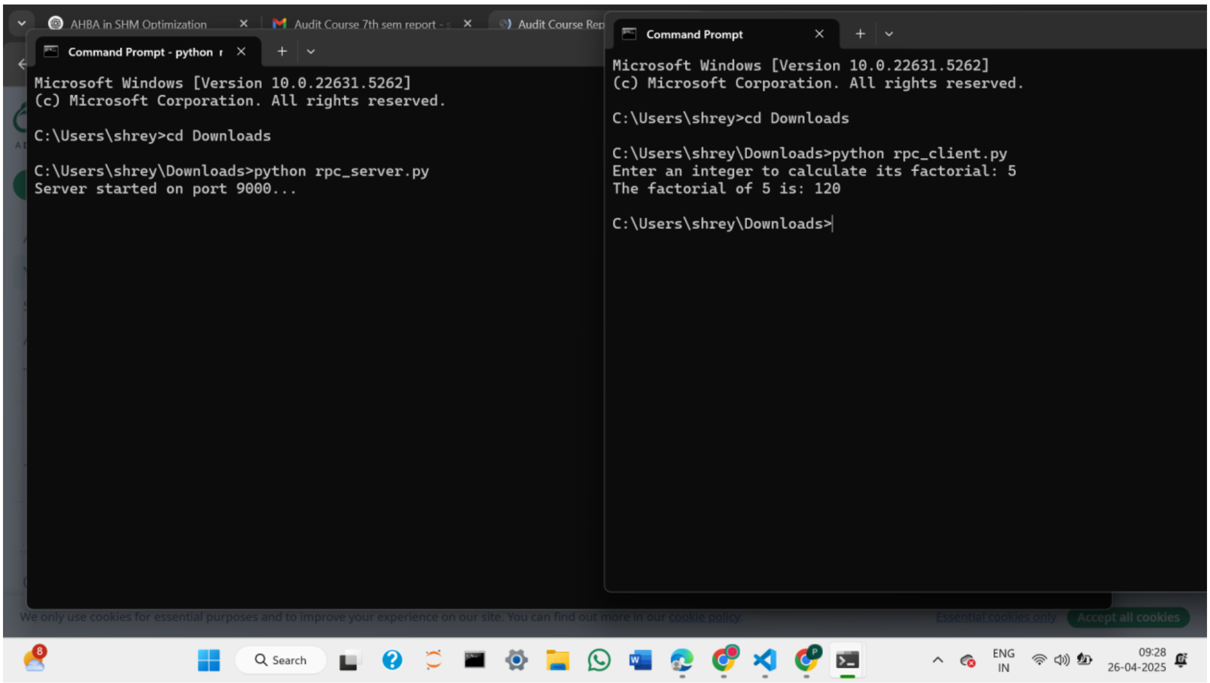
```
from xmlrpc.server import SimpleXMLRPCServer
import math

def calculate_factorial(n):
    if n < 0:
        return "Factorial is not defined for negative numbers."
    return math.factorial(n)

# Create and configure server
server = SimpleXMLRPCServer(("localhost", 9000), logRequests=True)
print("Server started on port 9000...")


# Register the factorial function
server.register_function(calculate_factorial, "factorial")


# Run the server
try:
    server.serve_forever()
except KeyboardInterrupt:
    print("\nServer shutting down.")
```



AHBA in SHM Optimization
Audit Course 7th sem report
Audit Course Rep
Command Prompt - python r
Command Prompt
We only use cookies for essential purposes and to improve your experience on our site. You can find out more in our cookie policy.
Essential cookies only
Accept all cookies
Search
ENG
IN
09:28
26-04-2025

```
Microsoft Windows [Version 10.0.22631.5262]
(c) Microsoft Corporation. All rights reserved.

C:\Users\shrey>cd Downloads

C:\Users\shrey\Downloads>python rpc_server.py
Server started on port 9000...
```

```
Microsoft Windows [Version 10.0.22631.5262]
(c) Microsoft Corporation. All rights reserved.

C:\Users\shrey>cd Downloads

C:\Users\shrey\Downloads>python rpc_client.py
Enter an integer to calculate its factorial: 5
The factorial of 5 is: 120

C:\Users\shrey\Downloads>
```

```
# # Set up the server
# def start_server():
#     server = SimpleXMLRPCServer(("localhost", 8000))
#     print("Server is running on port 8000...")

#     # Register the function
#     server.register_function(calculate_factorial, "factorial")

#     # Keep the server running
#     server.serve_forever()

# if __name__ == "__main__":
#     start_server()


## factorial
# def factorial(n):

#     if n < 0:
#         return "Error: Factorial of a negative number is undefined."
#     elif n == 0 or n == 1:
#         return 1
#     else:
#         result = 1
#         for i in range(2, n + 1):
#             result *= i
#         return result
```