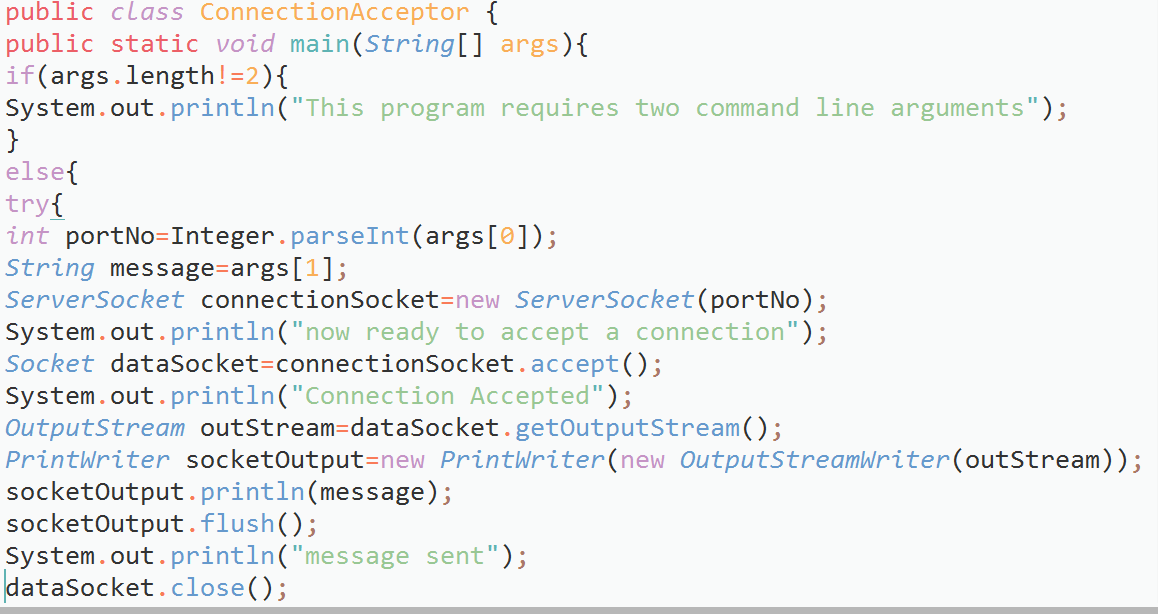
**Lab # 02**

**Objective: To become familiar with Stream Socket API.**

**Task 1: Compile and run the above code. Start the acceptor first and then the requestor with appropriate command line arguments. Describe and explain the output.**

**Code:**

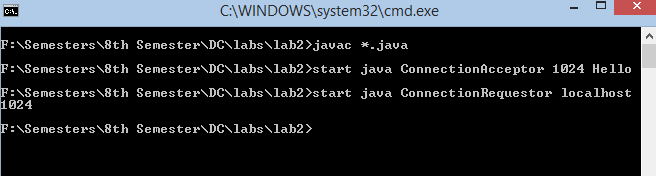
**Acceptor:**

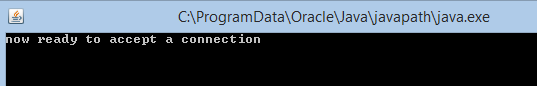
****

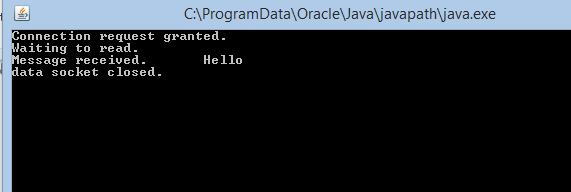
**Requestor:**

****

**Output:**

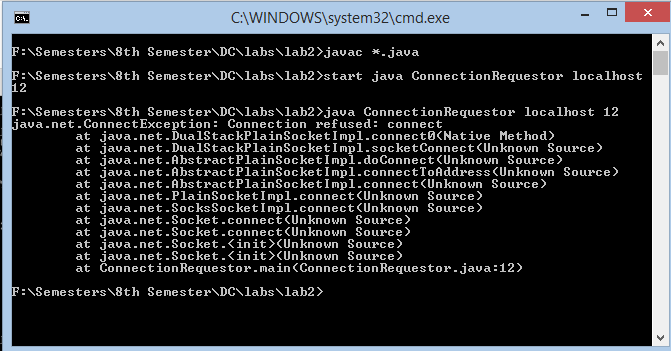
****

****

****

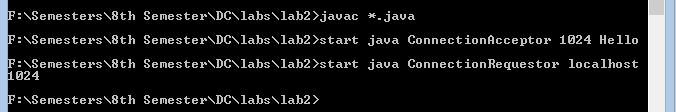
**Description: First we start the acceptor that accepts the request and responds accordingly and then requestor that request any service. In above case acceptor starts and prepares its service that is a message in this case and then requestor is started in order to make a request that required the name and port number of acceptor in order to establish a connection and get the data or service after it receives the message the acceptor flushes all the data (some like session ends) and close the socket and ends the connection. We can say here the acceptor class is working as a server while requestor is working as receiver.**

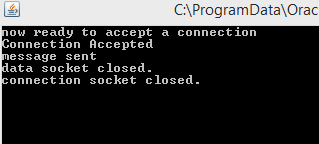
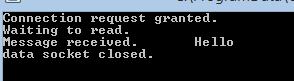
**Task2: Now run the code again, but reverse the order of program’s execution. Start the requestor first and then the acceptor. Describe and explain the outcome.**

****

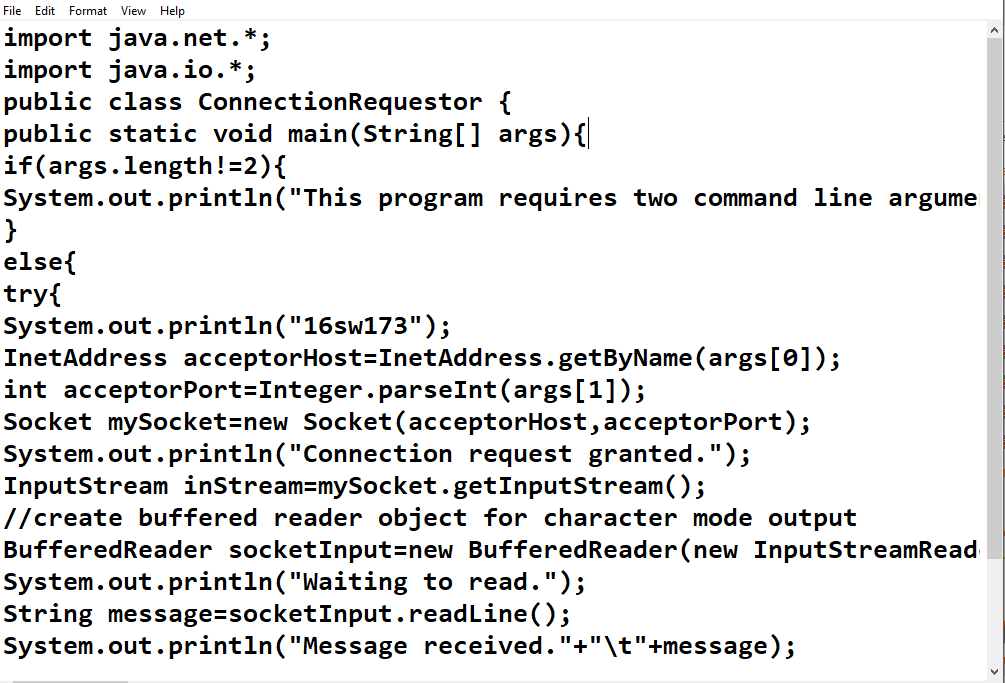
**Description: Exception occurs as there is no acceptor to accept the requestors connection.**

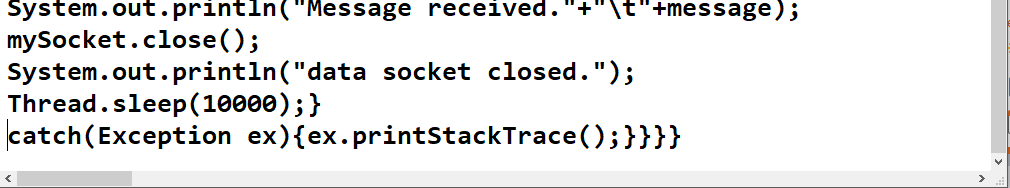
**Task 3: Add a time delay of 5 seconds in the ConnectionAcceptor process just before the message is written to the socket, then run the program. This will show you the blocking at the receiver. Show a trace of the output of the processes.**

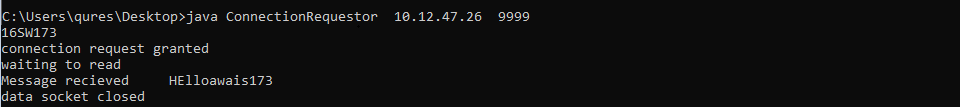
****

** **

**Task 4:Modify the sample code to include two way communication between the client and the server.**

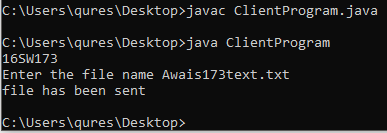
****

****

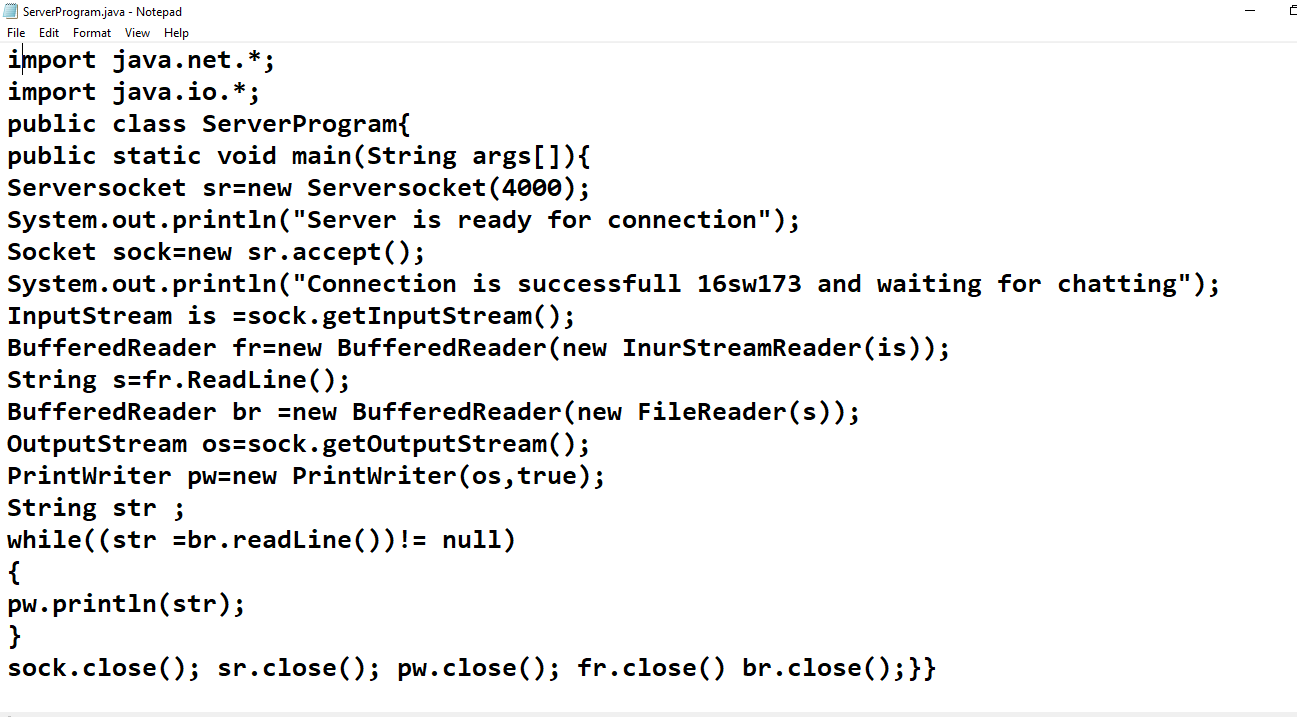
****

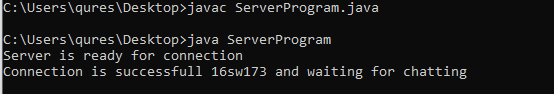
**Task 5 :Modify the sample code to send complete files between the client to the server.**

****

****

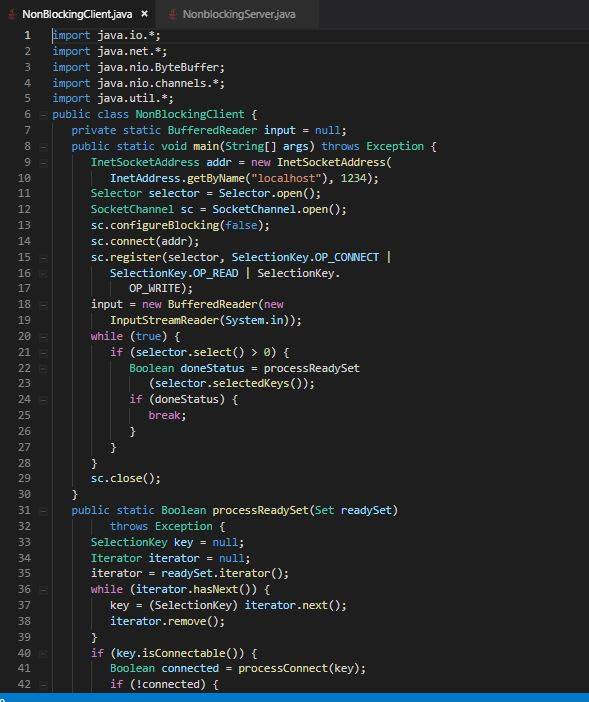
**Server Side**

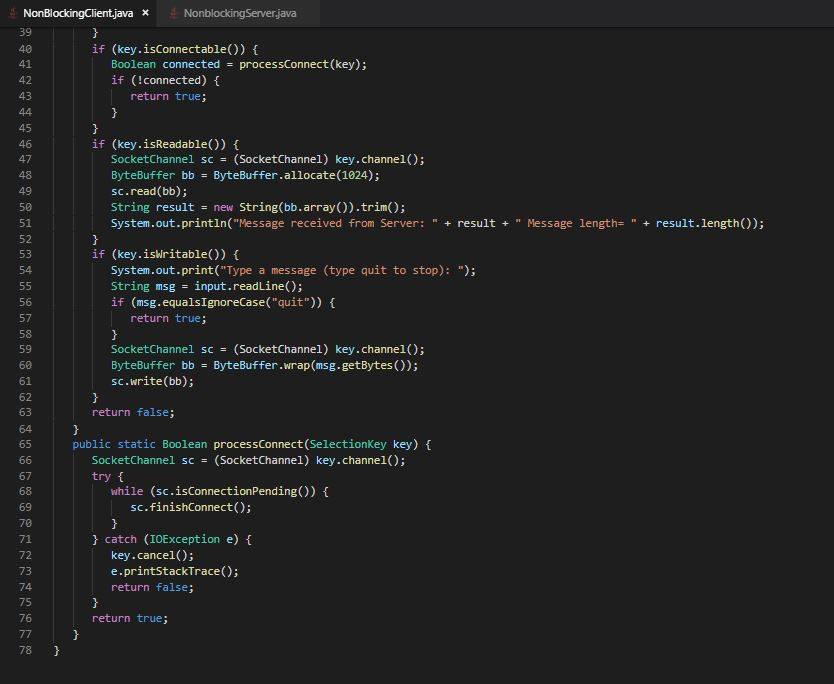
****

****

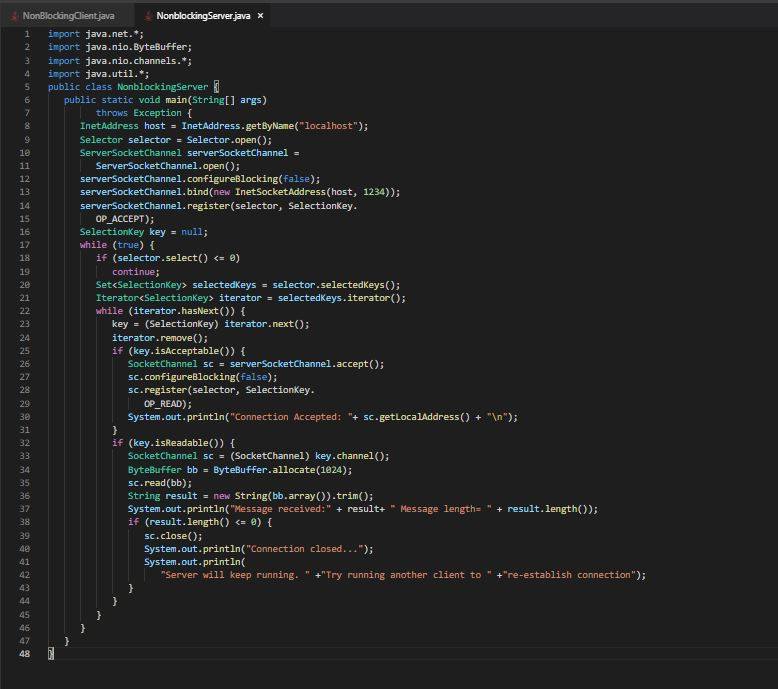
**Task 6 :Explore the non-blocking java socket API in the nio package and implement a sample program.**

**Client**

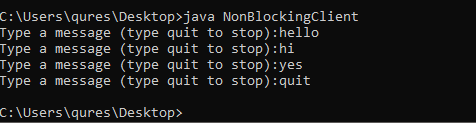


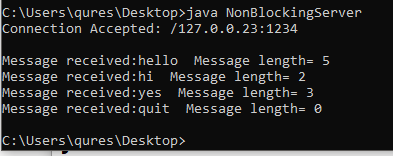


**Server**



**Output**

****

****