**LAB#02**

|  |
| --- |
| Department of Software Engineering  Mehran University of Engineering and Technology, Jamshoro |

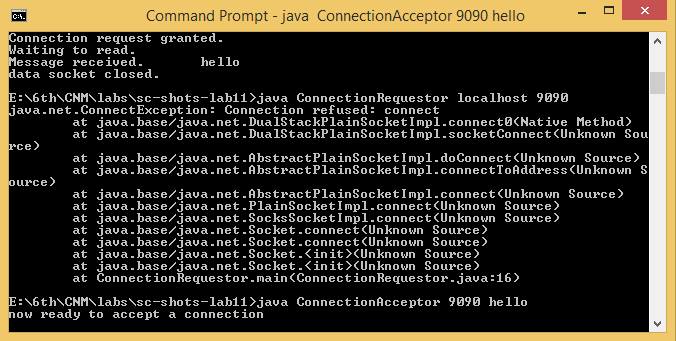
|  |  |  |  |
| --- | --- | --- | --- |
| Distributed Computing | | | |
| Instructor | Rabeea Jaffari | **Practical/Lab No.** | 2 |
| Date |  | **CLOs** |  |
| Signature |  | **Assessment Score** |  |

|  |  |
| --- | --- |
| Topic | Creating applications using Stream sockets. |
| Objectives | To learn creation of Stream sockets |

1. **Compile and run the above code, but reverse the order of program’s execution. Start the requestor first and then the acceptor. Describe and explain the outcome.**

**CODE:**

|  |  |
| --- | --- |
| **REQUESTOR:** | **ACCEPTOR:** |

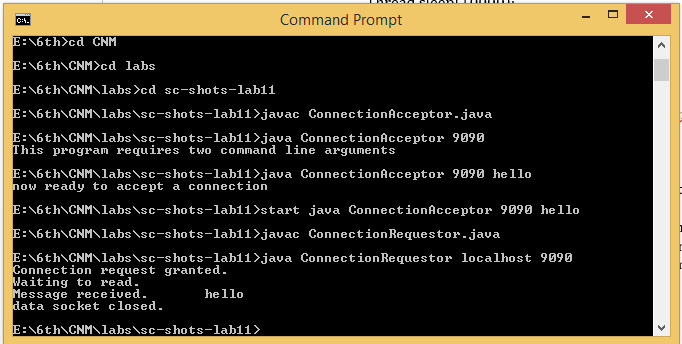
**OUTPUT: **

1. **Add a time delay of 10 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.**

**CODE:**

|  |  |
| --- | --- |
| **REQUESTOR:** | **ACCEPTOR:** |

**OUTPUT:**

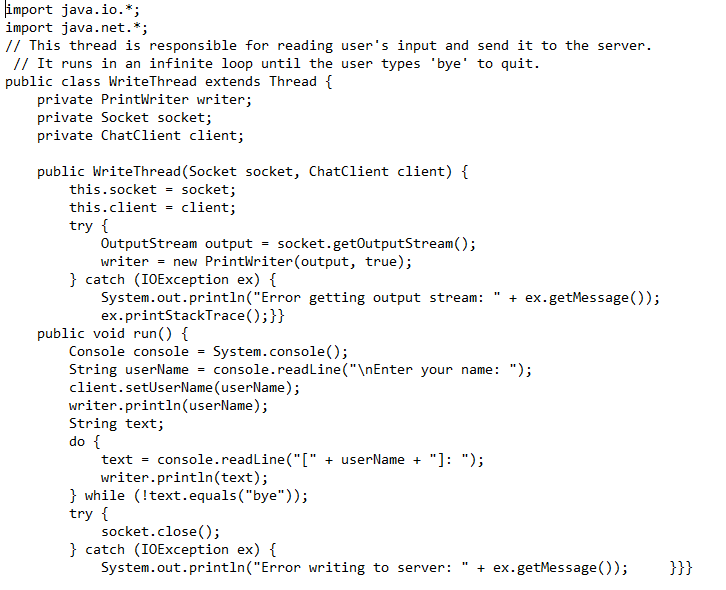
****

1. **Implement a simple chat program using stream sockets.**

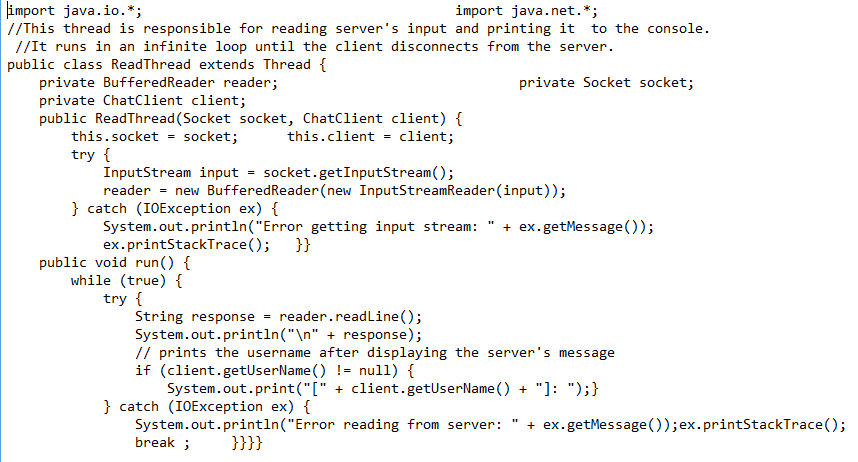
**CODE:**

|  |  |
| --- | --- |
| **SERVER:** | **CLIENT:** |

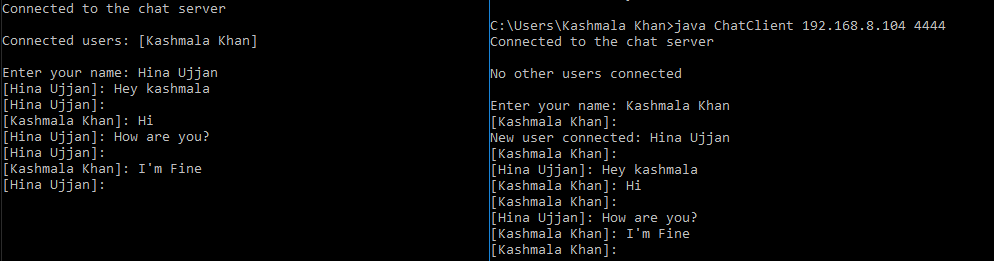
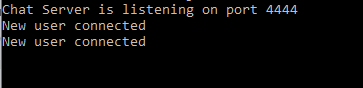
**THREADS:**





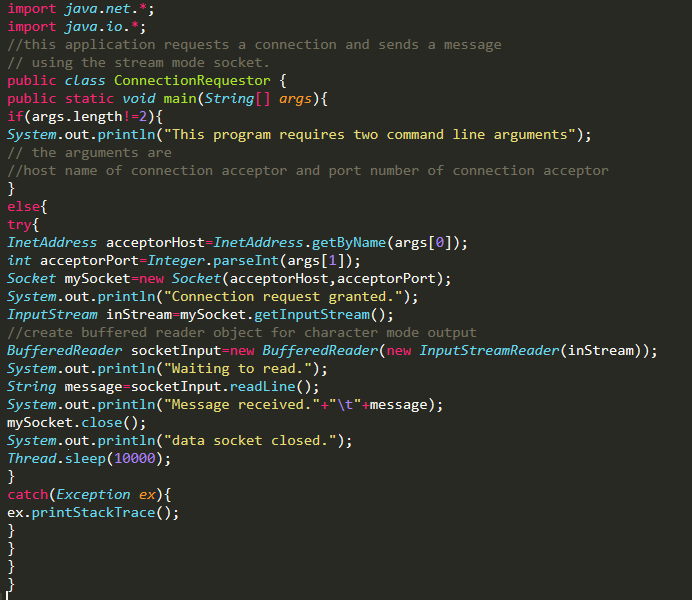


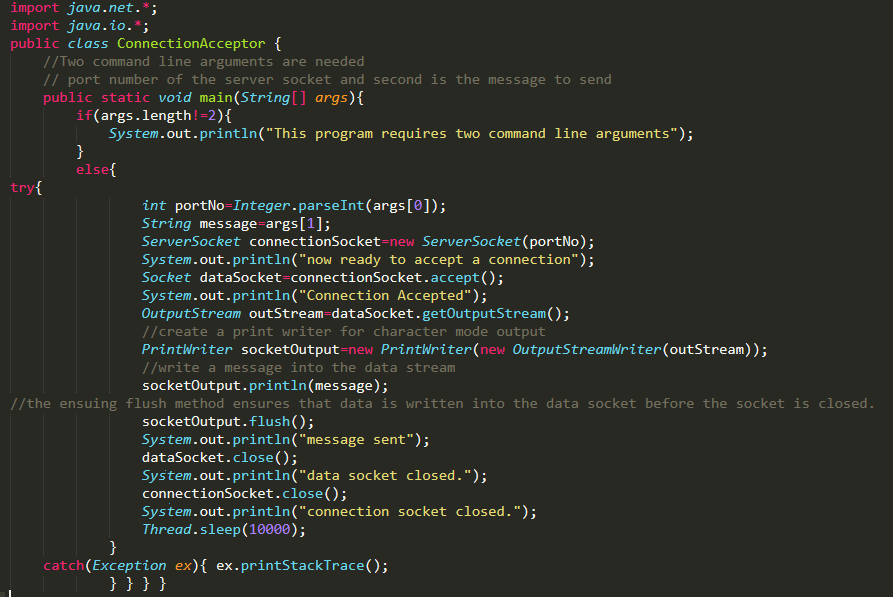
**OUTPUT:**

****

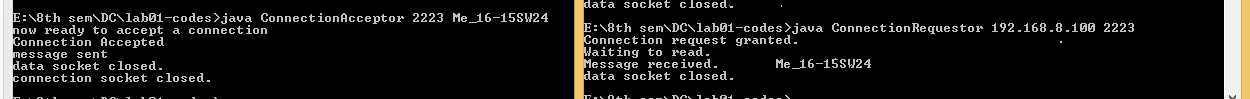
1. **Compile and run the above code. Start the acceptor first and then the requestor with appropriate command line arguments. Describe and explain the output.**

**CODE:**

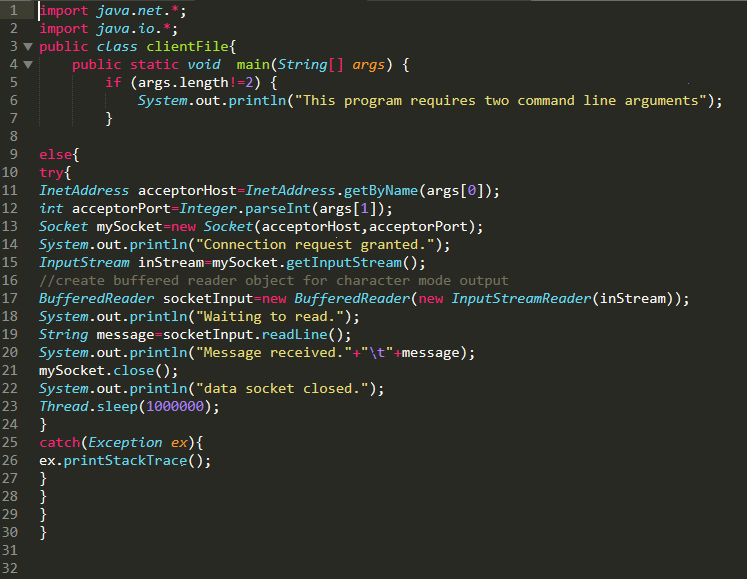


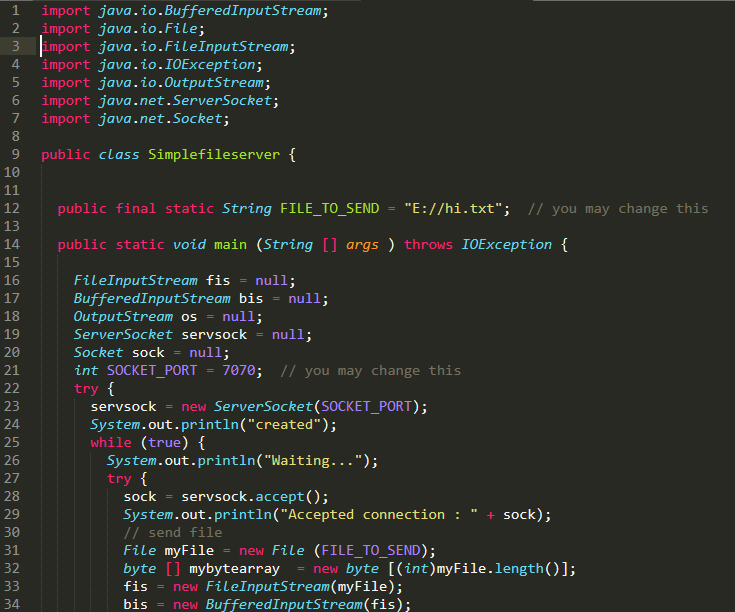


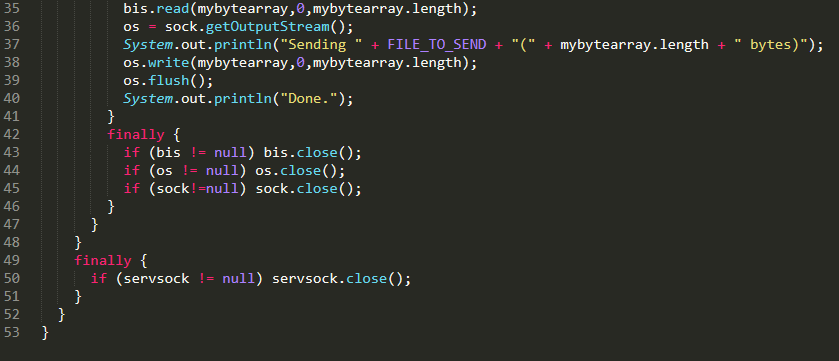
**OUTPUT:**



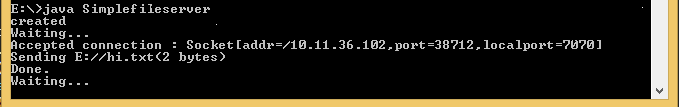
1. **Modify the sample code to send complete files between the client to the server.**

**CODE:** 

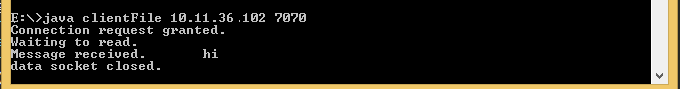




**SENT:**



**RECEIVED:**



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

**CODE:**

