SimpleChat - Test Cases: SEG 2105

Nicolas Landreville - 8579308 - nland013@uottawa.ca

Phase 1

1001

System: Simple Chat
Server startup check
Severity: 1

Expected Result:

- The server reports that it is listening for clients by displaying the following message: Server listening for clients on port 5555

Actual Result:

- The server reports that it is listening for clients by displaying the following message: Server listening for clients on port 5555

1002

System: Simple Chat
Client startup check without a server
Severity: 1

Expected result:

- The client reports it cannot connect by displaying message: Error: Can't setup connection! Terminating client.

- The client terminates.

Actual result:

- The client reports it cannot connect by displaying message: Error: Can't setup connection! Terminating client.

- The client terminates.

1003

System: Simple Chat
Client connection with default arguments
Severity: 1

Expected result:

- The client displays no message and waits for user input.
- The server displays no message.

Actual result:

- The client displays no message and waits for user input.
- The server displays no message.

1004

System: Simple Chat
Data transfer and data echo
Severity: 1

Expected result:

- The message is echoed on the client side, but is preceded by a greater than symbol (">").
- The server displays a message similar to the following: Message received: <user input> from localhost(127.0.0.1)

Actual result:

- The message is echoed on the client side, but is preceded by a greater than symbol (">").
- The server displays a message similar to the following: Message received: <user input> from localhost(127.0.0.1)

1005

System: Simple Chat
Server termination check
Severity: 1

Expected result:

- The client waits for input. Upon receiving input, the client displays the message "> Could not send message to server. Terminating client."
- The clients exits.

Actual result:

- The client waits for input. Upon receiving input, the client displays the message "> Could not send message to server. Terminating client."
- The clients exits.

1006

System: Simple Chat
Server termination check
Severity: 1

Expected result:

- The server does not display any message regarding the disconnection, and then accepts the second connection.

Actual result:

- The server does not display any message regarding the disconnection, and then accepts the second connection.

1007

System: Simple Chat
Multiple client connections and data transfer
Severity: 1

Expected result:

- Each client connects properly.
- Every message typed into any one of the consoles is echoed on all of the client consoles and the server console in the same way as in Testcase 1004.

Actual result:

- Each client connects properly.
- Every message typed into any one of the consoles is echoed on all of the client consoles and the server console in the same way as in Testcase 1004.

1008

System: Simple Chat
Remote connection
Severity: 1

Expected result:

- Connection occurs as in Testcase 1003.

Actual result:

- Connection occurs as in Testcase 1003.

1009

System: Simple Chat
Data transfer to a remote connection
Severity: 1

Expected result:

- The connection occurs as in Testcase 1003.
- Any data typed is echoed as in Testcase 1004.

Actual result:

- The connection occurs as in Testcase 1003.
- Any data typed is echoed as in Testcase 1004.

1010

System: Simple Chat
Multiple remote connections
Severity: 1

Expected result:

- Connection occurs as in Testcase 1003.

Actual result:

- Connection occurs as in Testcase 1003.

Phase 2

2001

System: Simple Chat
Server startup check with default arguments
Severity: 1

Expected result:

- The server reports that it is listening for clients by displaying the following message: Server listening for clients on port 5555
- The server console awaits for user input

Actual results:

```
Server listening for connections on port 5555
```

No difference between expected results and actual results.

2002

System: Simple Chat
Server startup check with default arguments
Severity: 1

Program Arguments: null
Expected result:

- The client reports it cannot connect without a login by displaying: ERROR - No login ID specified. Connection aborted.
- The client terminates.

Actual results:

```
ERROR - No login ID specified.  Connection aborted.

Process finished with exit code 1
```

No difference between expected results and actual results.

2003

System: Simple Chat
Client startup check with a login and without a server
Severity: 1

Program Arguments: testUser

Expected result:

- The client reports it cannot connect to a server by displaying: Cannot open connection. Awaiting command.
- The client waits for user input

Actual results:

```
Invalid input. Using default host
Connecting to host: localhost
Invalid input. Using default port
Connecting to port: 5555
Cannot open connection.  Awaiting command.
```

Same values expected and received.

2004

System: Simple Chat
Client connection with default arguments
Severity: 1

Program Arguments: testUser

Expected result:

- The server displays the following messages in sequence:
  A new client is attempting to connect to the server.
  Message received #login <loginID> from null.
  <loginID> has logged on.
- The client waits for user input

Actual results:

```
SERVER:
Server listening for connections on port 5555
A new client is attempting to connect to the server.
Message received: #login testUser from null
testUser has logged on

CLIENT:
Invalid input. Using default host
Connecting to host: localhost
Invalid input. Using default port
Connecting to port: 5555
testUser has logged on
```

Order of message created by server and client are the same as in the expected results.

2005

System: Simple Chat
Client Data transfer and data echo
Severity: 1

Program Arguments: testUser

Expected result:

- The message is echoed on the client side, but is preceded by the sender's loginID and the greater than symbol(">").
- The server displays a message similar to the following: Message received: <user input> from <loginID>

Actual results:

```
SERVER:
Message received: hello from 127.0.0.1 (127.0.0.1) with log-in ID testUser

CLIENT:
hello
testUser > hello
```

Expected and actual are identical, except server message which provides additional information.

2006

System: Simple Chat
Multiple local connections
Severity: 1

Program Arguments: testUser1
Program Arguments: testUser2
Program Arguments: testUser3
Program Arguments: testUser4

Expected result:

- All client messages are echoed as in Testcase 2005.
- All messages from the server console are echoed on the server console and to all clients, but are preceeded by "SERVER MESSAGE> ".

Actual results:

```
SERVER:
Message received: hello from 127.0.0.1 (127.0.0.1) with log-in ID testUser1
Message received: how from 127.0.0.1 (127.0.0.1) with log-in ID testUser2
Message received: are from 127.0.0.1 (127.0.0.1) with log-in ID testUser3
Message received: you? from 127.0.0.1 (127.0.0.1) with log-in ID testUser4
I am well
SERVER MSG > I am well

testUser1:
hello
testUser1 > hello
testUser2 > how
```

```
testUser3 > are
testUser4 > you?
SERVER MSG > I am well

testUser2:
testUser1 > hello
how
testUser2 > how
testUser3 > are
testUser4 > you?
SERVER MSG > I am well

testUser3:
testUser1 > hello
testUser2 > how
are
testUser3 > are
testUser4 > you?
SERVER MSG > I am well

TestUser4:
testUser1 > hello
testUser2 > how
testUser3 > are
you?
testUser4 > you?
SERVER MSG > I am well
```

In assignment instructions: ”SERVER MSG>”, in testcases :”SERVER MESSAGE>”. Does not modify meaning of message so keeping assignment value. Expected results and actual results are the same.

2007

System: Simple Chat
Server termination command check
Severity: 1

Expected result:

- The server quits.

Actual results:

```
Server listening for connections on port 5555
#quit

Process finished with exit code 0
```

Expected and actual are identical.

2008

System: Simple Chat
Server stop check
Severity: 2

Program Arguments: testUser
Program Arguments: testUser1

Expected result:

- The server displays: Server has stopped listening for connections.
- The client displays: WARNING - Server has stopped listening for connections.
- Data echoing still works normally.
- Attempting to connect other clients will hang the console as these new connections are put on backlog until the max backlog (default of 10) is reached.
- Restarting the server will cause the backlogged clients to connect properly.

Actual results:

```
SERVER:
#stop
Server has stopped listening for connections.
Message received: hello from 127.0.0.1 (127.0.0.1) with log-in ID testUser
#start
Server listening for connections on port 5555
A new client is attempting to connect to the server.
Message received: #login testUser1 from null
testUser1 has logged on


testUser:
testUser has logged on
WARNING - Server has stopped listening for connections.
hello
testUser > hello
testUser1 has logged on

testUser1:
testUser1 has logged on
```

Expected and actual are identical.

2009a

System: Simple Chat
Server close command check
Severity: 2

Program Arguments: testUser

Expected result:

- Server displays in sequence:
  Server has stopped listening for connections.
  ¡loginID¿ has disconnected.

- The client displays:
  WARNING - The server has stopped listening for connections
  SERVER SHUTTING DOWN! DISCONNECTING!
  Abnormal termination of connection.

  and then waits for input.

Actual results:

```
SERVER:
testUser has logged on
#close
testUser has disconnected
Server has stopped listening for connections.
#start
Server listening for connections on port 5555
A new client is attempting to connect to the server.




testUser:
testUser has logged on
WARNING - Server has stopped listening for connections.
SERVER SHUTTING DOWN! DISCONNECTING!
Abnormal termination of connection.
Connection closed.
```

Expected and actual are identical, except the last line from the client. This is due to the fact that when the logOff method is called for the client it automatically prints that message as confirmation.

2009b

System: Simple Chat
Server close command check
Severity: 1

Program Arguments: testUser

Expected result:

- The server closes, restarts and then displays: Server listening for connections on port 5555.
- The client connects normally as described in Testcase 2004.

Actual results:

```
SERVER:
Server listening for connections on port 5555
#close
Server has stopped listening for connections.
#start
Server listening for connections on port 5555
A new client is attempting to connect to the server.
Message received: #login testUser from null
testUser has logged on


testUser:
Invalid input. Using default host
Connecting to host: localhost
Invalid input. Using default port
Connecting to port: 5555
testUser has logged on
```

Expected and actual are identical, except the last line from the client. This is due to the fact that when the logOff method is called for the client it automatically prints that message as confirmation.

2010

System: Simple Chat
Client terminates.
Severity: 1

Program Arguments: testUser

Expected result:

- Client terminates.

Actual results:

```
testUser:
Cannot open connection.  Awaiting command.
#quit
Connection closed.
```

No difference between expected and actual results.

2011

System: Simple Chat
Client logoff check.
Severity: 1

Program Arguments: testUser

Expected result:

- Client disconnects and displays:
  Connection closed. (Under NT, it will display
  Abnormal termination of connection.)

Actual results:

```
testUser:
testUser has logged on
#logoff
Connection closed.
```

Identical expected and actual results.

2012

System: Simple Chat
Client host and port setup commands check.
Severity: 1

Program Arguments: testUser

Expected result:

- Client disconnects and displays:
  Connection closed. (Under NT, it will display
  Abnormal termination of connection.)

Actual results:

```
testUser:
testUser has logged on
#logoff
Connection closed.
```

Identical expected and actual results.

2013

System: Simple Chat
Starting a server on a non-default port.
Severity: 1

Program Arguments: 1234

Expected result:

- The server displays: Server listening for connections on port 1234.

Actual results:

```
Server listening for connections on port 1234
```

Identical expected and actual results.

2014

System: Simple Chat
Connecting a client to a non-default host or port
Severity: 1

Program Arguments: 1234
Client Argument: testUser localhost 1234

Expected result:

- The connection occurs normally.

Actual results:

```
A new client is attempting to connect to the server.
Message received: #login testUser from null
testUser has logged on
```

Identical expected and actual results.

2015

System: Simple Chat
Multiple remote clients disconnections and reconnections
Severity: 1

Client Argument: testUser
Client Argument: testUser1

Expected result:

- The first set of connections occur normally.
- When the server is closed, all clients are disconnected.
- The server displays the following message when the #setport command is used: port set to: <newport>.
- The server restarts and displays: Server listening for connections on port <newport>.
- The clients change port as in Testcase 2012.

- The clients reconnect normally.
- The clients are disconnected when the server quits.

Actual results:

```
Server:
A new client is attempting to connect to the server.
Message received: #login testUser from null
testUser has logged on
A new client is attempting to connect to the server.
Message received: #login testUser1 from null
testUser1 has logged on
Message received: hello there from 127.0.0.1 (127.0.0.1) with log-in ID testUser
Message received: General Kenobi from 127.0.0.1 (127.0.0.1) with log-in ID testUser1
#close
testUser has disconnected
testUser1 has disconnected
Server has stopped listening for connections.
#setport 1212
New port is: 1212
#start
Server listening for connections on port 1212
A new client is attempting to connect to the server.
Message received: #login testUser from null
testUser has logged on
A new client is attempting to connect to the server.
Message received: #login testUser1 from null
testUser1 has logged on
#quit
Server has stopped listening for connections.
testUser has disconnected
testUser1 has disconnected

testUser1:
Connecting to port: 5555
testUser1 has logged on
testUser > hello there
General Kenobi
testUser1 > General Kenobi
SERVER SHUTTING DOWN! DISCONNECTING!
Abnormal termination of connection.
Connection closed.
#setport 1212
New port: 1212
#login
Connection established.
testUser1 has logged on
Server has shut down.
Connection closed.
```

Identical expected and actual results.

2016

System: Simple Chat
Client changing hosts
Severity: 1

Expected result:

- The two servers start up normally.
- The client connects to the first server normally.
- When the client disconnects it displays: Connection closed. (Exception: NT will display Abnormal termination of connection)
- When the client disconnects, the server displays: <loginID> has disconnected.
- The client changes host as in Testcase 2012.
- The client reconnects normally as in Testcase 2015.

Actual results:

```
Server1:
A new client is attempting to connect to the server.
Message received: #login testUser1 from null
testUser1 has logged on
testUser1 has disconnected

Server2:
A new client is attempting to connect to the server.
Message received: #login testUser1 from null
testUser1 has logged on
```

Identical expected and actual results.

2017

System: Simple Chat
Client changing hosts
Severity: 1

Expected result:

- In both cases, all remaining clients and the server get the following message: <loginID> has disconnected.
- The clients display: Connection Closed.

Actual results:

```
Server:
A new client is attempting to connect to the server.
Message received: #login testUser from null
testUser has logged on
A new client is attempting to connect to the server.
Message received: #login testUser1 from null
testUser1 has logged on
```

```
A new client is attempting to connect to the server.
Message received: #login testUser2 from null
testUser2 has logged on
testUser has disconnected
testUser1 has disconnected

testUser:
testUser has logged on
testUser1 has logged on
testUser2 has logged on
#quit
Connection closed.

testUser1:
testUser1 has logged on
testUser2 has logged on
testUser has disconnected
#logoff
Connection closed.

testUser2:
testUser2 has logged on
testUser has disconnected
testUser1 has disconnected
```

Identical expected and actual results.

2018

No access to other systems, did not run test.

2019

No access to other systems, did not run test.