CSC 435/524 Computer Networks
Instructor: Dr. Natarajan Meghanathan

Sample Questions for Module 2 - Java Socket Programming

1) Define the term "socket". What are the two types of sockets that we used in the course and briefly explain their use?

2) Identify the Java class as well as the appropriate method or constructor in that class and the corresponding arguments to the function you would use for ***connectionless sockets*** to do or obtain the following:
   a. To get the IP address of the remote machine from which the message came.

   b. To get the port number of the remote process from which the message came.

   c. To get the IP address of the local host from which the message was sent.

   d. To get the port number of the local process from which the message was sent.

   e. To set a timeout for the socket to stop waiting for an incoming message at the receiver end.

   f. To set the destination IP address of the packet being sent.

   g. To open a socket at the local host at the specific IP address interface and port number

3) Consider the following Java socket programs to send and receive a message. Currently, with these programs, the sender can only send a message and the receiver can only receive the message. Modify these codes such that the sender can send any number of messages (one-by-one) and each of these messages are to be received and displayed at the receiver side. Make sure your modified versions can compile without any error.

```
import java.net.*;
import java.io.*;

class datagramReceiver{
  public static void main(String[ ] args){
    try{
      int MAX_LEN = 40;
      int localPortNum = Integer.parseInt(args[0]);
      DatagramSocket mySocket  = new DatagramSocket(localPortNum);
      byte[] buffer = new byte[MAX_LEN];
      DatagramPacket packet = new DatagramPacket(buffer, MAX_LEN);
      mySocket.receive(packet);
      String message = new String(buffer);
      System.out.println(message);
      mySocket.close( );
    }
    catch(Exception e){e.printStackTrace( );}
  }
}
```

Receiver Socket Program

```
import java.net.*;
import java.io.*;

class datagramSender{
  public static void main(String[ ] args){
    try{
       InetAddress receiverHost = InetAddress.getByName(args[0]);
       int receiverPort = Integer.parseInt(args[1]);
       String message = args[2];
       DatagramSocket mySocket = new DatagramSocket( );
       byte[] buffer = message.getBytes( );
       DatagramPacket packet = new DatagramPacket(buffer,
                    buffer.length, receiverHost, receiverPort);
       mySocket.send(packet);
       mySocket.close( );
    }
    catch(Exception e){ e.printStackTrace( ); }
  }
}
```

Sender Socket Program

4) Why is it not needed for a "sender only" program to instantiate a Datagram Socket on a specific port number? Is there a port number assigned to such sockets? If so, how?

5) What strategy does a sender-receiver program uses to make sure it gets a reply from the receiver-sender program? How does the receiver-sender program finds out to what address and port number it can send the reply to? Indicate the appropriate methods used as part of your answer.

6) What is the need for using the flush( ) method with connection-oriented sockets?

7) Name the I/O classes and their appropriate methods that you would employ to send and receive information using connection-oriented sockets?

8) What is the difference between the ServerSocket class and Socket class, with regards to their use?

9) The following server program greets only one client and stops. Extend it to a server program (write the complete code with appropriate modifications to the given code below) such that the server can greet any number of clients. Your modified server program should compile without any error.

```
import java.net.*;
import java.io.*;
class connectionServer{
   public static void main(String[ ] args){
      try{
           String message = "Hello, Welcome to the World of Java Socket Programming";
           int serverPortNumber = 3456;
           ServerSocket connectionSocket = new ServerSocket(serverPortNumber);
           Socket dataSocket = connectionSocket.accept( );
           PrintStream socketOutput = new PrintStream(dataSocket.getOutputStream( ));
           socketOutput.println(message);
           socketOutput.flush( );
           dataSocket.close( );
           connectionSocket.close( );
           }
       catch(Exception e){
           e.printStackTrace( );
         }
     }
 }
```

10) What is the interface that an user-defined class needs to implement so that an object of that class can be transmitted across a socket? What I/O classes and appropriate methods of these classes would you use to send and receive objects across connection-oriented sockets?

11) What is the advantage of a concurrent server over an iterative server? How is this advantage realized? Briefly explain.

12) What strategy (in the form of program statements/methods) would you employ to make sure that a multicast sender receiver program waits to send any message until all the receivers are started?

13) What two methods of the Thread class would you use to begin the execution of a thread? Which of these two methods do you need to mandatorily override to impart some functionality to the thread and why would you need to do so?

14) We have seen at least two distinct functions (methods) that are blocking calls. What are those two functions and explain in what context they are typically used?

15) Why multicast sockets are run as connectionless datagam sockets rather than connection-oriented stream sockets?

16) When can a multicast sender also receive the message that it sent? When would it not receive the message?

17) Are the individual receiver IP addresses used for multicasting a message? What does a receiver need to do to receive a multicast message? Also, indicate the appropriate Java method to accomplish this.

18) If you want a multicast process to be able to send and receive message at any time, how many Multicast Sockets do you need to open for that process and why? How many port numbers would you need to use to accomplish this? Justify your answer.