

Name: _____

J#: _____

Jackson State University
CSC 435 Computer Networks, Spring 2019
Instructor: Dr. Natarajan Meghanathan
Exam 1 (Take Home)

Due on: Feb. 20th, 2019: 4 PM (Hardcopy in Class)

Maximum Points: 100

**Print this exam and answer in the space provided. Use additional sheets, if needed.
For Socket programming questions, print the code and output on your own paper and
staple together with the answers for the other questions.**

Submit everything together (as one submission) in class

1) (25 pts) Consider the use of the Classless Interdomain Routing (CIDR). Let there be a network PQR that requires the support of 1023 hosts and we assign contiguous class C network address spaces starting from 202.89.16.0, (choose additional network address spaces, as needed).

Compute the following for network PQR:

- (1) Subnet mask (2) Network address
- (3) Broadcast address (4) Range of valid IP addresses
- (5) Efficiency of IP address assignment.
- (6) Identify the class C network address spaces required.

Name: _____

J#: _____

Name: _____

J#: _____

2) (25 pts) Consider implementing subnetting to support three departments within an organization. The three departments P, Q and R need support for 30, 40 and 60 hosts. Choose an appropriate class-based address space to do the subnetting. Compute the following for each of the three subnets. Also, compute the efficiency of your IP address assignment. Show all the work in detail.

a) Subnet mask b) Subnet address c) Subnet broadcast address d) Valid range of unicast IP addresses

Subnet	Subnet Mask	Subnet Prefix (Subnet Address)	Broadcast IP Address	Range of Valid IP Addresses
P				
Q				
R				

Name: _____

J#: _____

Name: _____

J#: _____

3) (50 points) Implement a "Remote calculator" application comprising of two programs (processes) using connectionless sockets (terminology: sender/receiver processes) as well as using connection-oriented sockets (terminology: client/server processes). The two implementations (using connectionless and connection-oriented sockets) are independent from each other and are worth for a maximum of 25 points, each.

Synopsis: A client/sender process accepts inputs (*two integers and an arithmetic operation*) from the user and sends these inputs to a server/receiver process. The server/receiver process performs the arithmetic operation on the two values received across the socket, and sends back the computed value (as a result of the arithmetic operation) to the client/sender process. The client/sender process displays (to the user) the final result received across the socket (from the server/receiver).

You are free to choose your own design/logic to implement your programs. You need to clearly explain (in the report and the video) the approach you took for the implementation.

Port Number: Let the server/receiver to process run on a port number corresponding to the last 4 digits of your J#. If the last 4 digits of your J# is less than 1024, add 10000 to it, and run on the resulting port number.

Arithmetic Operations: The four arithmetic operations to be implemented are: addition, subtraction, multiplication and division.

Note: For operations (subtraction and division) that are not commutative, you can assume the following order: the second input (right operand) is subtracted from the first input (left operand); and likewise, the first input (numerator) is divided by the second input (denominator). If the receiver/sender finds out that the denominator is zero, appropriate error message should be sent to the client/sender.

Server/Receiver Design: Make sure, your *server is running continuously and can handle requests from infinitely many clients* (one at a time) and your client program should be run each time you want to do a computation. The server/receiver process can be stopped by pressing CTRL+Z or CTRL+C (by doing so, the socket will also be closed at the server/receiver side. So, do not worry about closing the socket if you are running the server/receiver in an infinite loop).

For each client/sender served, the server/receiver should locally print (on screen) the IP address and port number (of the client/sender) from which the request for computation came.

Where to run the programs: You could run on your personal computer, using the localhost option.

What to Submit:

Your code for the client/server (for connection-oriented sockets) and for the sender/receiver (for connectionless sockets) and snapshots of at least three client/sender processes (placed suitably on your monitor screen) showing the passage of inputs and the result obtained as well as a snapshot of the server printing the IP address/port number of the different clients/senders from which the inputs for the arithmetic operation were received.