HW #2: Inventory Query on UDP, Due Date*: 11:59pm 02/08/2020, Cutoff Date: 11:59pm 02/10/2020**
Submission: (1) Upload all source code (.java) files to both Blackboard and the Virtual Servers (cs3700a and cs3700b), AND (2) set up and test java programs (.class files) on cs3700a and cs3700b (See Part II for details)
****Submission will NOT be accepted after the cutoff deadline**

An option of peer programming: You may choose to work on this assignment individually or in a team of two students. If you choose to work in a ***team of two students***, you **must** 1) **add** both team members' first and last names as the **comments** on Blackboard when submitting the .java files (both team members are required to submit the same .java files on Blackboard), and 2) put a ***team.txt*** file including both team members' names under your HW02/ on cs3700a (both team members are required to complete Task II under both home directories on cs3700a). For grading, I will randomly pick the submission in one of the two *home directories* of the team members on **cs3700a** and **cs3700b**, and then give both team members the same grade. If the team information is *missing* on Blackboard or cs3700a, two or more submissions with similar source codes with just variable name/comment changes will be considered to be a violation of the Integrity described in the course policies and both or all will be graded as 0.

Grading: Your programs will be graded via testing and points are associated with how much task it can complete. A program that cannot be compiled or crashes while running will receive up to 5% of the total points. A submission of .java files that are similar to any online Java program with only variable name changes will receive 0% of the total points.

Programming in Java is highly recommended, since all requirements in this assignment are guaranteed to be supported in Java. If you choose to use any other language such as Python or C/C++, it is YOUR responsibility, before the cutoff deadline, to (2) set up the compiling and running environment on cs3700a and cs3700b, (2) make sure that I can run/test your programs in your home directory on cs3700a and cs3700b, and (3) provide a README file under HW02/ on cs3700a to include the commands that I need to use.

Part I (85%): Write a client program and a server program to implement the following protocol on top of UDP.

- **Client Program**:
  1. Display a message to ask the User to input the DNS or IP of the machine on which the Server Program runs.
  2. Display the following table on the standard output:

Item ID	Item Description
00001	New Inspiron 15
00002	New Inspiron 17
00003	New Inspiron 15R
00004	New Inspiron 15z Ultrabook
00005	XPS 14 Ultrabook
00006	New XPS 12 UltrabookXPS

  3. Display a message on the standard output to ask the User to input an Item ID, and validate the user input. If the input is not a valid Item ID, ask the User to re-type it.
  4. Once getting a valid item ID from the User, send a request including this Item ID (e.g., 00005 or "00005") to the Server program to ask for a quote, and record the local time right before sending such request.
  5. Receive and interpret the response from the Server program, get the local time right after such response is received, and display the following information on the standard output, (e.g., if 00005 were provided by the User earlier on)

Item ID	Item Description	Unit Price	Inventory	RTT of Query
00005	XPS 14 Ultrabook	$999.99	261	… ms

  where "RTT of Query" is the difference between the time in Steps 5 and 4 in millisecond.
  6. Display a message on the standard output to ask the User whether to continue. If yes, repeat steps 2 through 5. Otherwise, close the socket and terminate the Client program.
- **Server Program**:
  1. Maintain the following information using an appropriate data structure of your choice (i.e., an array of a Class you defined). You do not have to place it in a file although you certainly can if you like.

Item ID	**Item Description**	**Unit Price**	**Inventory**
00001	New Inspiron 15	$379.99	157
00002	New Inspiron 17	$449.99	128
00003	New Inspiron 15R	$549.99	202
00004	New Inspiron 15z Ultrabook	$749.99	315
00005	XPS 14 Ultrabook	$999.99	261
00006	New XPS 12 UltrabookXPS	$1199.99	178

  2. Wait for receiving a packet from a Client.
  3. Once a packet is received from a Client, retrieve the information relevant to the requested Item ID from the data structure you used in Step 1 and send back such information to the Client.
  4. Repeat Steps 2 and 3 infinitely until an exception is caught.
  5. Close the datagram socket.

Part II (15%): Test your programs on the Virtual Servers in the cloud and your laptop/home computer.

Warning: to complete this part, especially when you work at home, you must first (1) **connect to GlobalProtect** using your NetID account (please read "**how to connect to GlobalProtect …**" at https://msudenver.edu/vpn/); then (2) **connect to the virtual servers cs3700a and cs3700b** using ***sftp*** and ***ssh*** command on MAC/Linux or ***PUTTY*** and ***PSFTP*** on Windows.

ITS only supports GlobalProtect on MAC and Windows machines. If your home computer has a different OS, it is your responsibility to figure out how to connect to cs3700a and cs3700b for programming assignments and submit your work by the cutoff deadline. Such issues cannot be used as an excuse to request any extension.

1. MAKE a directory "**HW02**" under your home directory on **cs3700a**.msdenver.edu and **cs3700b**.msudenver.edu, a subdirectory "**server**" under "**HW02**" on **cs3700a.msudenver.edu**, and a subdirectory "**client**" under "**HW02**" on **cs3700b.msudenver.edu**.
2. UPLOAD and COMPILE the ***server*** program under "**HW02/server**" and the ***client*** program under "**HW02/client**" on the VMs.
3. TEST ***the server program*** running on **cs3700a**.msudenver.edu together with ***a client program*** running on your laptop or lab computer and ***another client program***, simultaneously, running on **cs3700b**.msudenver.edu to test all the possible cases.
4. SAVE a file named ***testResultsClient.txt*** under "**HW02/client**" on **cs3700b**.msudenver.edu, which captures the outputs of your ***client*** program when you test it. You can use the following command to redirect the standard output (stdout) and the standard error (stderr) to a **file** on UNIX, Linux, or Mac, and view the contents of the file

```
java prog_name_args | tee testResultsClient.txt //copy stdout to the .txt file
cat file-name     //display the file's contents.
```