CS 3341 Fall 2025

Programming Homework – Client-Server Computing

Due: 11/12 (Wed) 11:59pm ***(Absolutely no extensions!)***

For this program I would like you to implement a client-server application of a very simple two-player game.

The HCF/LCM game

The game is a competition between two players. Each round player one will select a number between 50 – 99 that is not prime. Then player two will pick a number between 60-99 that is not prime. After the numbers are selected, the game is scored in the following way:

- Player one's score will be the highest common factor (HCF) of the two numbers
- Player two's score will be the last digit of the lowest common multiple of the two numbers (LCM).

Each game consists of two rounds where each player takes turns to be player one/two, and the player with the higher cumulative score over the two rounds will be the winners (ties are allowed).

The following situation will lead to a player receiving 0 point for a round, and the opponent getting 100 point. (If both players make the same choices, both will get 0 point).

- The player picked a prime number
- The player picked a number outside the range
- The player picked the same numbers in both rounds.

Task

Your task is to implement the game where a server will be the host, and two clients will connect to the server and play a game against one another.

- The server should be started at the background, and it will listen for client connecting. (Use port 24175 for this program)
- The client will send in a request to the server requesting a game (for this program, you always assume the client and server is on the same machine (so you should use "127.0.0.1" as the IP address)
- Once the server accepts a connection from the client, it should inform the client which player (player one or player two) that he/she is.

- The client should then send an ID (for this program, a single string) to the server for the record.
- Once the server collects the names, it should print the statement "<player one ID> vs. <player two ID>: Game start".
- Then the server should send a request to ask the two players to send in the first number.
- Once the server receives the numbers, it should calculate the score for this round. And then sent the score of both players, together with the numbers both player picked, back to the client.
- The server should then print the result of round 1: including the number selected, and the score of each player.
- Once the client receives the information, it should send in a number of round 2
- Then the server should calculate the score for round 2, and send the following tp the players
    - The score of round 2 for each player
    - The total score for each player
    - Whether the player win/lose/draw the match (sending 1/-1/0 respectively)
- The server should then print the result of round 2: including the number selected, and the score of each player, and then print out who is the winner.
- The client, after receiving the information about who wins, should print a statement (you can select what statement to print, it must be different for the 3 cases, and please, no inappropriate language).
- After that, the client should disconnect from the server, and quit
- The server should also quit.

You should write two programs (in either Java or C++): server.cpp/server.java and client.cpp/client.java to implement the server and client. You are welcomed to implement extra classes/functions and include them in your submission. You should use processes for C++, threads for Java.

What to hand in

You should zip all the source code into a zip file and hand in the program. You should comment on the code in the following way:

"For each of the bullet point above, indicate where in the code that implement that bullet point" (You can copy the bullet point and paste it (as a comment) before the block of code you used to implement this line.)

Each comment is worth 1 point out of 100. So, if you do not comment, the best you can get is 86.

A note on testing

Since you are writing your program as a client-server program, that means testing in your IDE is not realistic. You will have to compile the client and server program separately, save it to a local directory, and then open 2-3 windows. Have one window run the server program, and two other windows run the client program.

Extra credit (15%)

You can hand in a version of the code that modified the behavior of the server program in the following way:

- Once the server accepted both players, it should create a new child process/thread that handles the rest of the game,
- After that, the server process/thread should return and listen to the next player
- The server will keep running until 10 games have finished.

Once again, comment on the code on these extra points.

Additional info

- Here is the list of all non-prime number between 50-99: 50, 51, 52, 54, 55, 56, 57, 58, 60, 62, 63, 64, 65, 66, 68, 69, 70, 72, 74, 75, 76, 77, 78, 80, 81, 82, 84, 85, 86, 87, 88, 90, 91, 92, 93, 94, 95, 96, 98, 99,
- A version of HCF function/method in C++/Java is provided for you. Also, give two numbers a, b, LCM(a, b) = a * b / HCF(a, b)