Course: IT114-002-S2025
Assignment: IT114 Module 4 Sockets Part3 Challenge
Student: Hitarth P. (hp627)
Status: Submitted | Worksheet Progress: 100.00%
Potential Grade: 10.00/10.00 (100.00%)
Received Grade: 0.00/10.00 (0.00%)
Grading Link: https://learn.ethereallab.app/assignment/v3/IT114-002-S2025/it114-module-4-sockets-part3-challenge/grading/hp627

Instructions

1. Ensure you read all instructions and objectives before starting.
2. Create a new branch from `main` called `M4-Homework`
    1. `git checkout main` (ensure proper starting branch)
    2. `git pull origin main` (ensure history is up to date)
    3. `git checkout -b M4-Homework` (create and switch to branch)
3. Copy the template code from here: GitHub Repository - M4 Homework
    - It includes Sockets `Part1`, `Part2`, and `Part3`. Put all into an M4 folder or similar if you don't have them yet (adjust `package` reference at the top if you chose a different folder name).
    - Make a copy of `Part3` and call it `Part3HW`
        - [ ] Fix the `package` and `import` references at the top of each file in this new folder
    - Immediately record to history
        - [ ] `git add .`
        - [ ] `git commit -m "adding M4 HW baseline files"`
        - [ ] `git push origin M4-Homework`
        - [ ] Create a Pull Request from `M4-Homework` to `main` and keep it open
4. Fill out the below worksheet
    - Each Problem requires the following as you work
        - [ ] Ensure there's a comment with your UCID, date, and brief summary of how the problem was solved
        - [ ] Code solution (add/commit periodically as needed)
        - [ ] Hint: Note how `/reverse` is handled
5. Once finished, click "Submit and Export"
6. Locally add the generated PDF to a folder of your choosing inside your repository folder and move it to Github
    1. `git add .`
    2. `git commit -m "adding PDF"`
    3. `git push origin M4-Homework`
    4. On Github merge the pull request from `M4-Homework` to `main`
7. Upload the same PDF to Canvas
8. Sync Local
    1. `git checkout main`
    2. `git pull origin main`

Section #1: (3 pts.) Challenge 1 - Coin Flip

Task #1 (3 pts.) - Implement a Coin Flip Command

Combo Task:

Weight: *100%*
Objective: *Implement a Coin Flip Command*

Details:

- `Client` must capture the user entry and generate a valid command per the lesson details
  - Command format must be `/flip`
- `ServerThread` must receive the data and call the correct method on `Server`
- `Server` must expose a method for the logic and send the result to everyone
  - The message must be in the format of `<who> flipped a coin and got <result>` and be from the Server
- Add code to solve the problem (add/commit as needed)

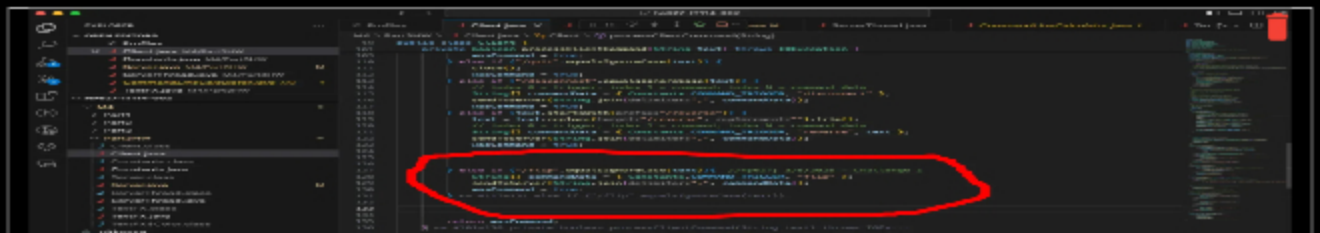
Item:#1

Weight: *40%*

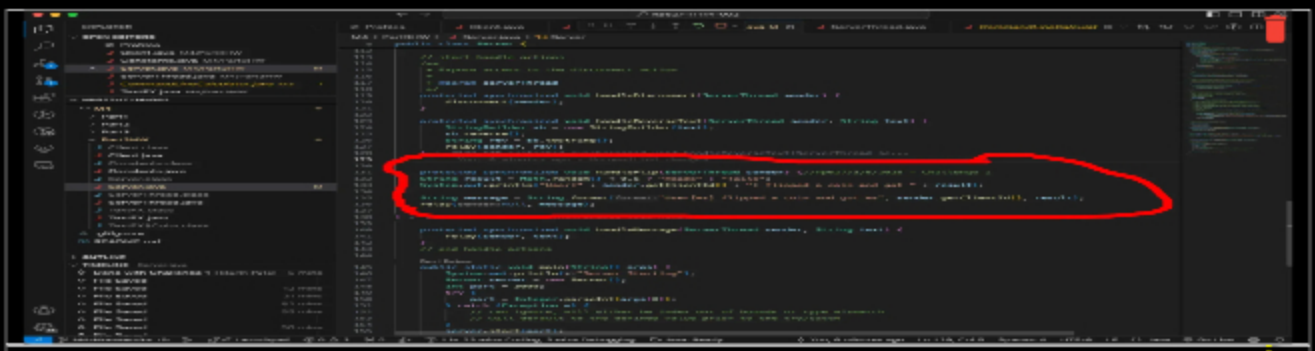
Details:
Multiple screenshots are expected

1. Snippet of relevant code showing solution (with ucid/date comment) from `Client`
   - Should only need to edit `processClientCommands()`
2. Snippet of relevant code showing solution (with ucid/date comment) from `ServerThread`
   - Should only need to edit `processCommand()`
3. Snippet of relevant code showing solution (with ucid/date comment) from `Server`
   - Should only need to create a new method and pass the result message to `relay()`
4. Show 5 examples of the command being seen across all terminals (2+ Clients and 1 Server)
   1. This can be captured in one screenshot if you split the terminals side by side

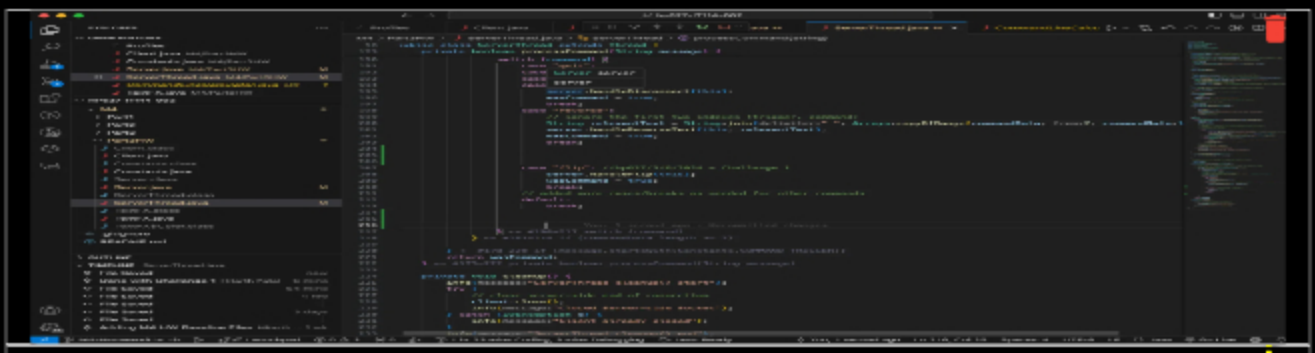
Image Prompt



Client.java



Server.java



ServerThread.java

Saved: 3/10/2025 5:23:40 PM

Item:#2

Weight: *20%*

Details:
Direct link to the file in the homework related branch from Github (should end in `.java`)

Url Prompt

URL #1
https://github.com/hitarthpat/hp627-IT114-002/main/M4/Part3HW/Client.java

URL
https://github.com/hitarthpat/hp62

URL #2
https://github.com/hitarthpat/hp627-IT114-002/main/M4/Part3HW/Server.java

URL
https://github.com/hitarthpat/hp62

URL #3

URL

https://github.com/hitarthpat/hp627-IT114-002/main/M4/Part3HW/ServerThread.java

https://github.com/hitarthpat/hp62

Saved: 3/10/2025 5:23:40 PM

Item:#3

Weight: *40%*

Details:
Briefly explain how the code solves the challenge (note: this isn't the same as what the code does)

Text Prompt

Your Response:

The code lets the user solve the challenge by flipping a coin online just by writing /flip. The sever side uses a random number generator to get "Heads" Or "Tails" and creates a message that gives the result of the coin flip. It broadcast the results of the coin flip to everyone in the networks.

Saved: 3/10/2025 5:23:40 PM

Section #2: (3 pts.) Challenge 2 - Private Message

Task #1 (3 pts.) - Implement a Private Message Command

Combo Task:

Weight: *100%*
Objective: *Implement a Private Message Command*

Details:

- `Client` must capture the user entry and generate a valid command per the lesson details
  - Command format must be `/pm <target id>`
- `ServerThread` must receive the data and call the correct method on `Server`
- `Server` must expose a method for the logic

- The message must be in the format of `PM from <who>: <message>` and be from the Server
- The result must only be sent to the original sender and to the receiver/target

- Add code to solve the problem (add/commit as needed)

Item:#1

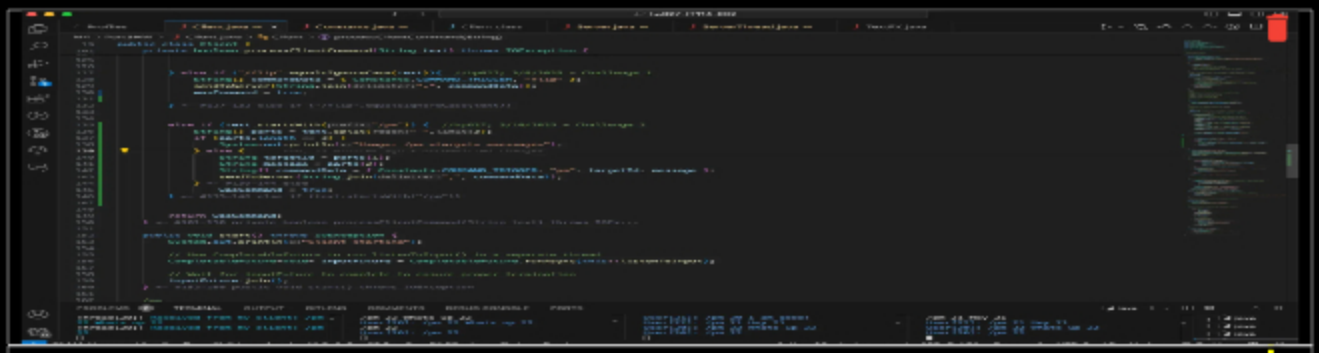
Weight: *40%*

Details:
Multiple screenshots are expected

1. Snippet of relevant code showing solution (with ucid/date comment) from `Client`
   - Should only need to edit `processClientCommands()`
2. Snippet of relevant code showing solution (with ucid/date comment) from `ServerThread`
   - Should only need to edit `processCommand()`
3. Snippet of relevant code showing solution (with ucid/date comment) from `Server`
   - Should only need to create a new method and pass the result message to `relay()`
4. Show 3 examples of the command being seen across all terminals (3+ Clients and 1 Server)
   1. This can be captured in one screenshot if you split the terminals side by side
   2. Note: Only the sender and the receiver should see the private message (show variations across different users)

Image Prompt

Code Output



Client.java

Server.java

ServerThread.java

Constants.java

Saved: 3/10/2025 5:34:32 PM

Item:#2

Weight: *20%*

Details:

Direct link to the file in the homework related branch from Github (should end in `.java`)

Url Prompt

URL #1

https://github.com/hitarthpat/hp627-

URL

https://github.com/hitarthpat/hp62

URL #2
https://github.com/hitarthpat/hp627-IT114-002/main/M4/Part3HW/Constants.java

URL
https://github.com/hitarthpat/hp62

URL #3
https://github.com/hitarthpat/hp627-IT114-002/main/M4/Part3HW/Server.java

URL
https://github.com/hitarthpat/hp62

URL #4
https://github.com/hitarthpat/hp627-IT114-002/main/M4/Part3HW/ServerThread.java

URL
https://github.com/hitarthpat/hp62

Saved: 3/10/2025 5:34:32 PM

Section #3: (3 pts.) Challenge 3 - Shuffle Message

Task #1 (3 pts.) - Implement a Shuffle Message Command

Combo Task:

Weight: *100%*
Objective: *Implement a Shuffle Message Command*

Details:

- `Client` must capture the user entry and generate a valid command per the lesson details
  - Command format must be `/shuffle <message>`
- `ServerThread` must receive the data and call the correct method on `Server`
- `Server` must expose a method for the logic and send the result to everyone
  - The message must be in the format of `Shuffled from <who>: <shuffled_message>` and be from the Server
- Add code to solve the problem (add/commit as needed)

Item:#1

Weight: *40%*

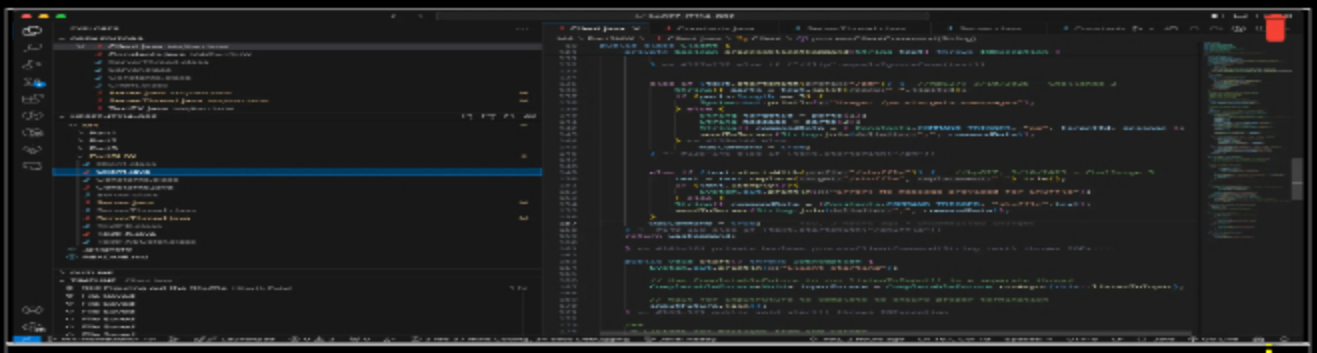
Details:

Multiple screenshots are expected

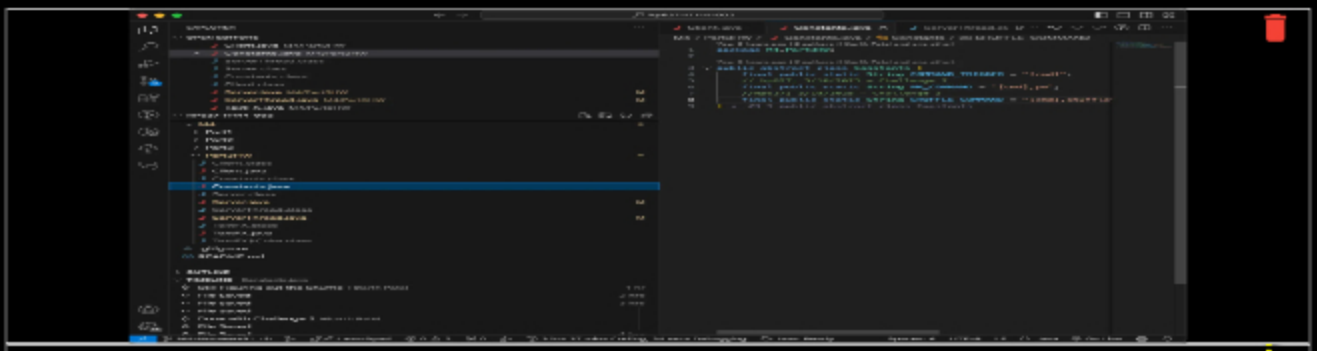
1. Snippet of relevant code showing solution (with ucid/date comment) from `Client`
   - Should only need to edit `processClientCommands()`
2. Snippet of relevant code showing solution (with ucid/date comment) from `ServerThread`
   - Should only need to edit `processCommand()`
3. Snippet of relevant code showing solution (with ucid/date comment) from `Server`
   - Should only need to create a new method and do similar logic to `relay()`
4. Show 3 examples of the command being seen across all terminals (2+ Clients and 1 Server)
   1. This can be captured in one screenshot if you split the terminals side by side

Image Prompt

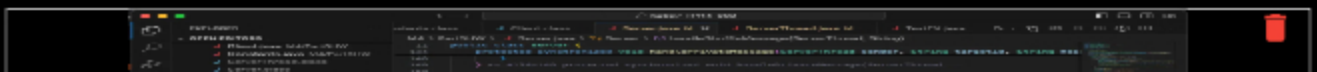
Code



Client.java



Constant.java



Server.java

ServerThread.java

Saved: 3/10/2025 5:48:41 PM

Item:#2

Weight: *20%*

Details:
Direct link to the file in the homework related branch from Github (should end in `.java`)

Url Prompt

URL #1
https://github.com/hitarthpat/hp627-IT114-002/main/M4/Part3HW/Client.java

URL
https://github.com/hitarthpat/hp62

URL #2
https://github.com/hitarthpat/hp627-IT114-002/main/M4/Part3HW/Constants.java

URL
https://github.com/hitarthpat/hp62

URL #3

URL

Section #4: (1 pt.) Misc

Task #1 (0.33 pts.) - Github Details

Combo Task:

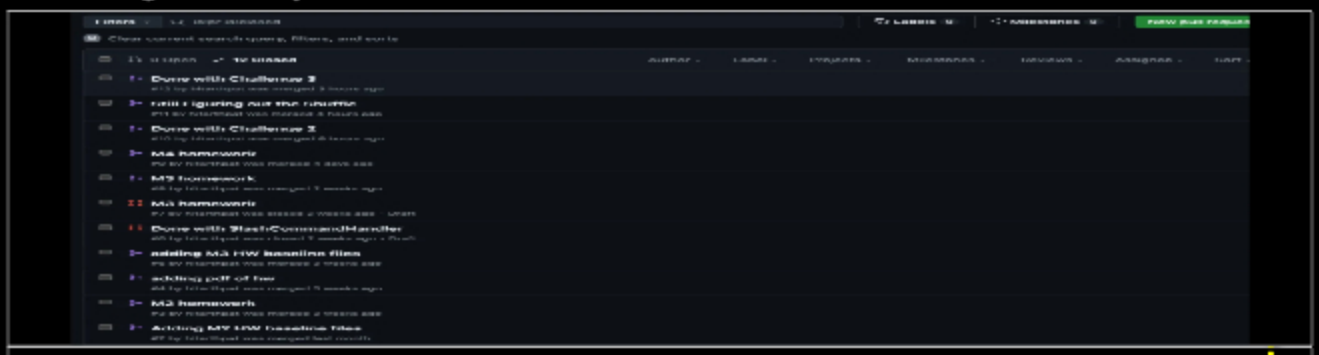
Weight: *33.33%*
Objective: *Github Details*

Item:#1

Weight: *60%*

Details:
From the Commits tab of the Pull Request screenshot the commit history Following minimum should be present

Image Prompt



Pull

Saved: 3/10/2025 5:51:48 PM

Item:#2

Weight: *40%*

Details:
Include the link to the Pull Request (should end in `/pull/#`)

Url Prompt

URL #1
https://github.com/hitarthpat/hp627-IT114-002/pull/2?q=is%3Apr+is%3Aclosed

URL
https://github.com/hitarthpat/hp62

Saved: 3/10/2025 5:51:48 PM

Task #2 (0.33 pts.) - WakaTime - Activity

Weight: *33.33%*
Objective: *WakaTime - Activity*

Details:

- Visit the WakaTime.com Dashboard
- Click `Projects` and find your repository
- Capture the overall time at the top that includes the repository name
- Capture the individual time at the bottom that includes the file time
- Note: The duration isn't relevant for the grade and the visual graphs aren't necessary

Image Prompt

Wakatime

Saved: 3/10/2025 5:52:49 PM

Task #3 (0.33 pts.) - Reflection

Sub-Tasks:

Task #1 (0.33 pts.) - What did you learn?

Weight: *33.33%*
Objective: *What did you learn?*

Details:
Briefly answer the question (at least a few decent sentences)

Text Prompt

Your Response:

I learned how to use the strucuture which was already provided to me to create changes for the challenges like creating a coin flip, word shuffler, and more. It taught me how to work with threads while also not messing up the connection which was already provided to me.

Saved: 3/10/2025 5:57:03 PM

Task #2 (0.33 pts.) - What was the easiest part of the assignr

Weight: *33.33%*
Objective: *What was the easiest part of the assignment?*

Details:
Briefly answer the question (at least a few decent sentences)

Text Prompt

Your Response:

The easiest part of the challenge was doing the coin flip features. Where I had to generate random numbers to check if its less than 0.5 to get the heads and tails.

Saved: 3/10/2025 5:59:13 PM

Task #3 (0.33 pts.) - What was the hardest part of the assign

Weight: *33.33%*
Objective: *What was the hardest part of the assignment?*

Details:
Briefly answer the question (at least a few decent sentences)

Briefly answer the question (at least a few decent sentences)

Text Prompt

Your Response:

The hardest part of the assignment was using the threads not know if I would mess it up. Making sure that I keep all the threads same has it was and also managing the input and the outputs of the user and the servers.

Saved: 3/10/2025 6:01:09 PM