

# Submission Worksheet

CLICK TO GRADE

<https://learn.ethereallab.app/assignment/IT114-002-S2024/it114-sockets-part-1-3-checkpoint/grade/pd438>

IT114-002-S2024 - [IT114] Sockets Part 1-3-Checkpoint

## Submissions:

Submission Selection

1 Submission [active] 2/19/2024 4:41:59 PM

## Instructions

^ COLLAPSE ^

Create a new branch for this assignment

Go through the socket lessons and get each part implemented (parts 1-3)

You'll probably want to put them into their own separate folders/packages (i.e., Part1, Part2, Part3) These are for your reference

Part 3, below, is what's necessary for this HW

<https://github.com/MattToegel/IT114/tree/Module4/Module4/Part3>

Create a new folder called Part3HW (copy of Part3)

Make sure you have all the necessary files from Part3 copied here and fix the package references at the top of each file

Add/commit/push the branch

Create a pull request to main and keep it open

Implement **two** of the following **server-side** activities for all connected clients (majority of the logic should be processed server-side and broadcasted/sent to all clients if/when applicable)

Simple number guesser where all clients can attempt to guess while the game is active

Have a /start command that activates the game allowing guesses to be interpreted

Have a /stop command that deactivates the game, guesses will be treated as regular messages (i.e., guess messages are ignored)

Have a guess command that include a value that is processed to see if it matches the hidden number (i.e., /guess 5)

Guess should only be considered when the game is active

The response should include who guessed, what they guessed, and whether or not it was correct (i.e., Bob guessed 5 but it was not correct)

No need to implement complexities like strikes

Coin toss command (random heads or tails)

Command should be something logical like /flip or /toss or /coin or similar

The result should mention *who* did *what* and got what *result* (i.e., Bob Flipped a coin and got heads)

Dice roller given a command and text format of "/roll #d#" (i.e., roll 2d6)

Command should be in the format of /roll #d# (i.e., roll 1d10)

The result should mention *who* did *what* and got what *result* (i.e., Bob rolled 1d10 and got 7)

Math game (server outputs a basic equation, first person to guess it correctly gets congratulated and a new equation is given)

Have a /start command that activates the game allowing equation to be answered

Have a /stop command that deactivates the game, answers will be treated as regular messages (i.e., any game related commands when stopped will be ignored)

Have an answer command that include a value that is processed to see if it matches the hidden number (i.e., /answer 15)

the hidden number (i.e., /answer 15)  
 The response should include who answered, what they answered, and whether or not it was correct (i.e., Bob answered 5 but it was not correct)  
 Private message (a client can send a message targetting another client where only the two can see the messages)  
 Command can be /pm, /dm followed by the user's name or an @ preceding the users name (clearly note which)  
 The server should properly check the target audience and send the response to the original sender and to the receiver (no one else should get the message)  
 Alternatively (make note if you do this and show evidence) you can add support to private message multiple people at once. Evidence should show a larger number of clients than the target list of the private message to show it works. Note to grader: if this is accomplished add 0.5 to total final grade on Canvas  
 Message shuffler (randomizes the order of the characters of the given message)  
 Command should be /shuffle or /randomize (clearly mention what you chose) followed by the message to shuffle (i.e., /shuffle hello everybody)  
 The message should be sent to all clients showing it's from the user but randomized  
 Example: Bob types /command hello and everyone receives Bob: lleho  
 Fill in the below deliverables  
 Save the submission and generated output PDF  
 Add the PDF to the Part3HW folder (local)  
 Add/commit/push your changes  
 Merge the pull request  
 Upload the same PDF to Canvas

Branch name: M4-Sockets3-Homework

Tasks: 7 Points: 10.00

Baseline (2 pts.)

^COLLAPSE ^

Task #1 - Points: 1

Text: Demonstrate Baseline Code Working

### Details:

This can be a single screenshot if everything fits, or can be multiple screenshots

### Checklist

\*The checkboxes are for your own tracking

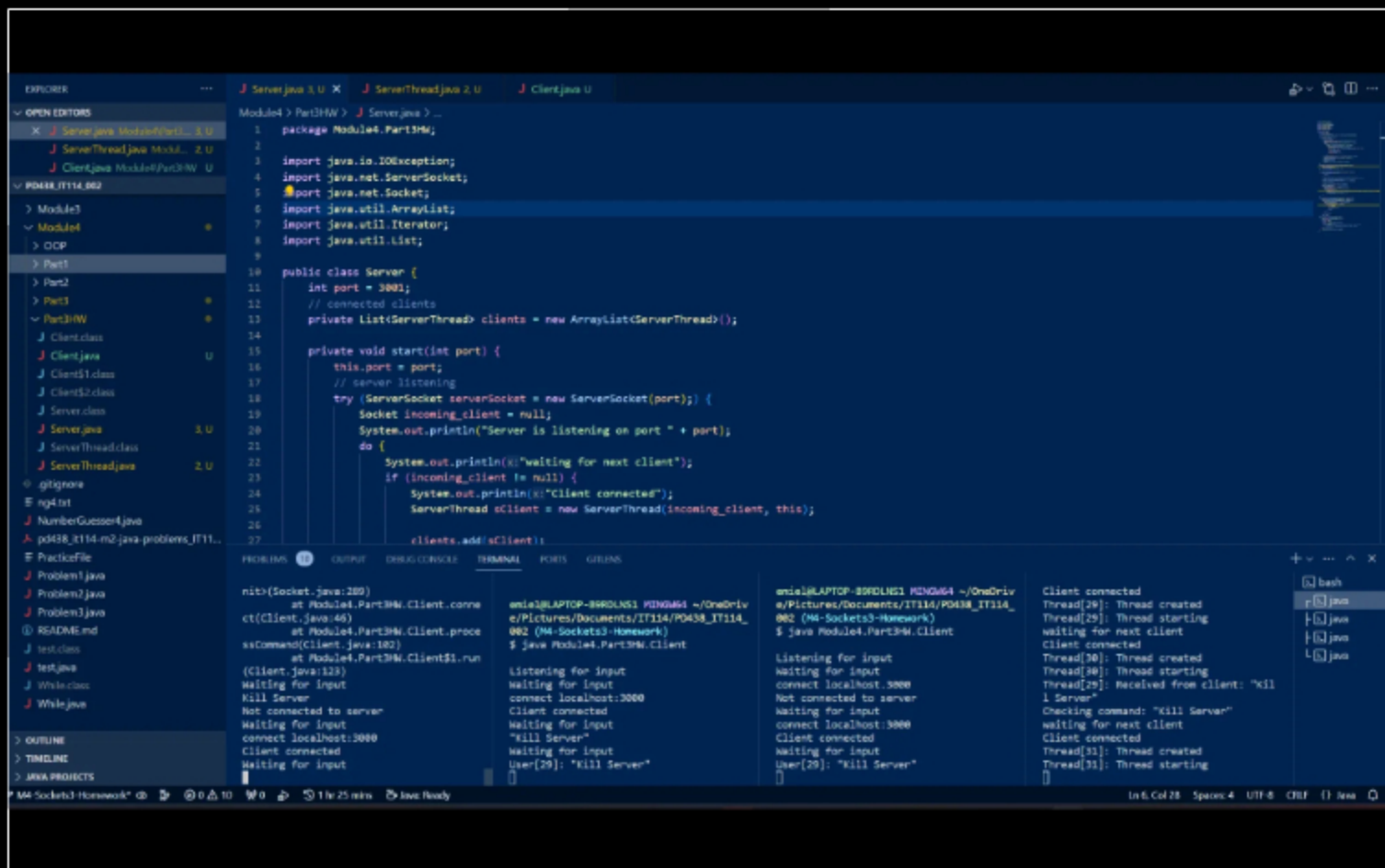
#	Points	Details
<input type="checkbox"/> #1	1	Server terminal/instance is clearly shown/noted
<input type="checkbox"/> #2	1	At least 3 client terminals should be visible and noted
<input type="checkbox"/> #3	1	Each client should correctly receive all broadcasted/shared messages
<input type="checkbox"/> #4	1	Captions clearly explain what each screenshot is showing
<input type="checkbox"/> #5	1	Include a screenshot showing you grabbed Parts 1-3 correctly and have them in your repository alongside Part3HW

Task: Screenshots

Small

Medium

Large



Grabbed parts 1-3 correctly. Also showed display of how it coompletely works.

### Checklist Items (5)

#1 Server terminal/instance is clearly shown/noted

#2 At least 3 client terminals should be visible and noted

#3 Each client should correctly receive all broadcasted/shared messages

#4 Captions clearly explain what each screenshot is showing

#5 Include a screenshot showing you grabbed Parts 1-3 correctly and have them in your repository alongside Part3HW

Feature 1 (3 pts.)

^COLLAPSE ^

^COLLAPSE ^

Task #1 - Points: 1

Text: What feature did you pick? Briefly explain how you implemented it

Checklist		*The checkboxes are for your own tracking
#	Points	Details
<input type="checkbox"/> #1	1	Feature is clearly stated (best to copy/paste it from above)
<input type="checkbox"/> #2	1	Explanation sufficiently and concisely describes implementation (should be aligned with code snippets in related task)

Response:

What I have decided to choose to complete this assignment is the head tails game and also private message. The reason why i picked these two is because of how simplistic these were. I wanted the user to be able to play the game while to be able to message the person. The way i implemented it, was to place it in the serverthread and the server so that it can be displayed.

^COLLAPSE ^

Task #2 - Points: 1

Text: Add screenshot(s) showing the implemented feature working (code and output)

**Details:**  
Add screenshots of the relevant code changes AND relevant output during runtime

Checklist		*The checkboxes are for your own tracking
#	Points	Details
<input type="checkbox"/> #1	1	Output is clearly shown and captioned
<input type="checkbox"/> #2	1	Code shows relevant snippets that accomplish feature, UCID and date are present in all code screenshots. Relevant captions are included for each screenshot of the code.

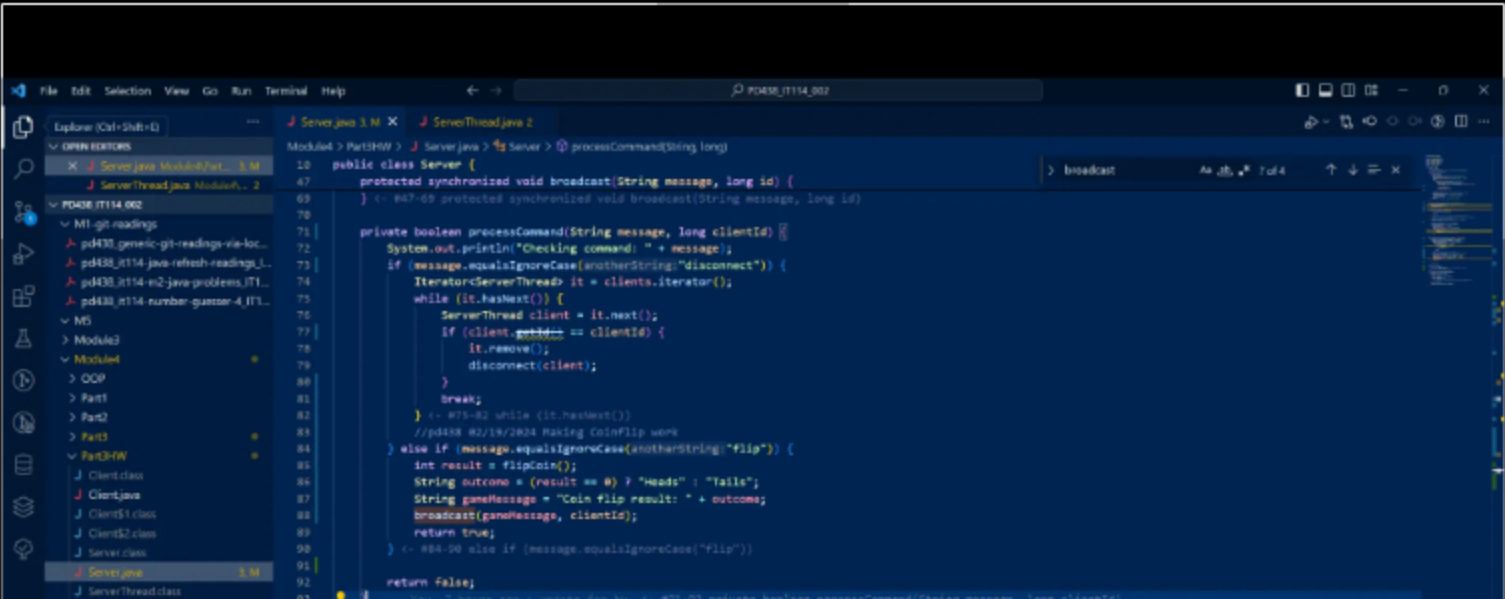
Task Screenshots:

Gallery Style: Large View

Small

Medium

Large



Showing how the code works. shows results as well the user having the number

## Checklist Items (2)

#1 Output is clearly shown and captioned

#2 Code shows relevant snippets that accomplish feature, UCID and date are present in all code screenshots. Relevant captions are included for each screenshot of the code.

### Feature 2 (3 pts.)

#### Task #1 - Points: 1

Text: What feature did you pick? Briefly explain how you implemented it

### Checklist

\*The checkboxes are for your own tracking

#	Points	Details
<input type="checkbox"/> #1	1	Feature is clearly stated (best to copy/paste it from above)
<input type="checkbox"/> #2	1	Explanation sufficiently and concisely describes implementation (should be aligned with code snippets in related task)

Response:

The other feature that I have picked, is the private message. I want to implement the ability for the client to private message while playing the coin flip game.

#### Task #2 - Points: 1

Text: Add screenshot(s) showing the implemented feature working (code and output)

### Details:

Add screenshots of the relevant code changes AND relevant output during runtime

### Checklist

\*The checkboxes are for your own tracking

#	Points	Details
---	--------	---------



#1	1	Output is clearly shown and captioned
#2	1	Code shows relevant snippets that accomplish feature, UCID and date are present in all code screenshots. Relevant captions are included for each screenshot of the code.

## Task Screenshots:

**Gallery Style: Large View**

Small
Medium
Large

The screenshot displays an IDE with a Java project. The main editor shows the `ServerThread.java` file. The code includes comments explaining the immutability of `String` objects and how they are handled in the server. The terminal output shows the server listening on port 3000 and handling client connections. The left sidebar shows the project structure with various Java files and packages.

Shows attempt to have user interaction with user 29 and 30.

## Checklist Items (2)

#1 Output is clearly shown and captioned

#2 Code shows relevant snippets that accomplish feature, UCID and date are present in all code screenshots. Relevant captions are included for each screenshot of the code.

Misc (2 pts.)

^COLLAPSE ^

Task #1 - Points: 1

Text: Reflection: Did you have an issues and how did you resolve them? If no issues, what did you learn during this assignment that you found interesting?

#	Points	Details
#1	1	An issue or learning is clearly stated
#2	1	Response is a few reasonable sentences


Response:

I had some issues completing the assignment. I had placed the code but VScode would give me kickback and then the program does not work. When i tried placing the code it says that there are no issues. But when i compile and make it work, it does not work so I am definitely confused.



Task #2 - Points: 1

Text: Pull request link

 Details:

URL should end with /pull/# and be related to this assignment

URL #1

[https://github.com/PD438/PD438\\_IT114\\_002/pull/7](https://github.com/PD438/PD438_IT114_002/pull/7)

End of Assignment