

# Submission Worksheet

CLICK TO GRADE

<https://learn.ethereallab.app/assignment/IT114-002-S2024/it114-sockets-part-1-3-checkpoint/grade/ns87>

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

## Submissions:

Submission Selection

1 Submission [active] 2/22/2024 1:28:35 AM

## 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

**i** 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

Small

Medium

Large

```

114 // Modified: Part3Copy 3: Client.java 3: Client 3: startNumberGuesser
115
116 public class Client {
117
118     //ACID: 0007
119     //DATE: 1/28/2004
120
121     public void startNumberGuesser() throws IOException {
122         if (!isConnected()) {
123             out.writeObject(obj:"start");
124         } else {
125             System.out.println("Not connected to the server");
126         }
127     }
128
129     public void stopNumberGuesser() throws IOException {
130         if (!isConnected()) {
131             out.writeObject(obj:"stop");
132         } else {
133             System.out.println("Not connected to the server");
134         }
135     }
136
137     public void makeGuess(int guess) throws IOException {
138         if (!isConnected()) {
139             out.writeObject("guess " + guess);
140         } else {
141             System.out.println("Not connected to the server");
142         }
143     }
144
145     public void flipCoin() throws IOException {
146         if (!isConnected()) {
147             out.writeObject(obj:"flip");
148         } else {
149             System.out.println("Not connected to the server");
150         }
151     }
152 }
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

## Demonstrate Baseline Code Working

## 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 ^



Task #1 - Points: 1

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

^COLLAPSE ^

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:

Feature in Number guess game

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

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.

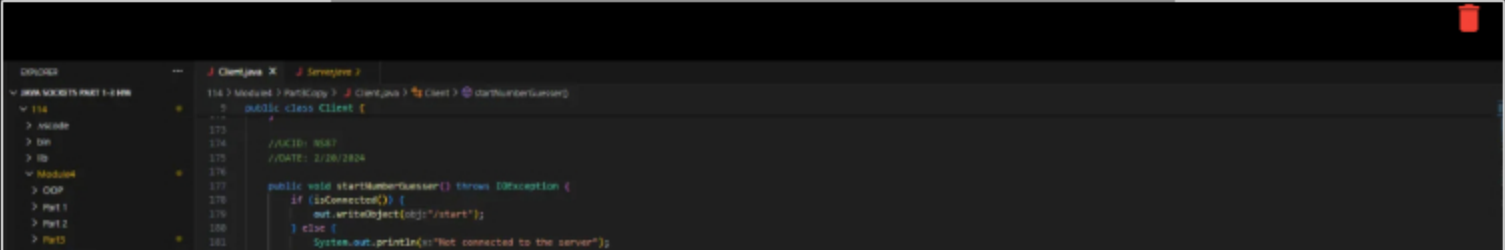
Task Screenshots:

Gallery Style: Large View

Small

Medium

Large







```

88 //DATE: 2/21/24
89 private void processNumberGuesserCommand(String command) {
90     if (command.equalsIgnoreCase(anotherString("/start")) {
91         server.broadcast(message:"Number guesser game has started! Guess a number between 0 and 50.", -1);
92         isNumberGuesserActive = true;
93     } else if (command.equalsIgnoreCase(anotherString("/stop")) {
94         server.broadcast(message:"Number guesser game has stopped. Guesses will now be treated as regular messages.", -1);
95         isNumberGuesserActive = false;
96     } else if (command.startsWith(prefix+"/guess")) {
97         int guessedNumber = Integer.parseInt(command.split(regex" ")[1]);
98     }
99 }
100
101
102 private void processCoinlessCommand(String command) {
103     if (command.equalsIgnoreCase(anotherString("/flip")) || command.equalsIgnoreCase(anotherString("/toss")) || command.equalsIgnoreCase(anotherString("/coin")) {
104         String result = (Math.random() < 0.5) ? "heads" : "tails";
105         server.broadcast("User[" + prefix + "] flipped a coin and got " + result + "!", -1);
106     }
107 }
108 }

```

PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS

Waiting for input

/start

Waiting for input

User[1]: Number guesser game has started! Guess a number between 0 and 50.

47

Waiting for input

User[21]: 47

47

Waiting for input

User[21]: /47

User[21]: /47

/flip

Waiting for input

User[21]: /flip

User[1]: User[21] flipped a coin and got tails!

0

Waiting for input

connect localhost:3000

Not connected to server

Waiting for input

connect localhost:3000

Client connected

Waiting for input

User[1]: Number guesser game has started! Guess a number between 0 and 50.

47

User[21]: 47

/47

Waiting for input

User[21]: /47

User[21]: /flip

User[1]: User[21] flipped a coin and got tails!

0

Client connected

Thread[22]: Thread created

Thread[22]: Thread starting

Thread[21]: Received from the client: /start

Checking command: /start

Checking command: Number guesser game has started! Guess a number between 0 and 50.

Thread[21]: Received from the client: 47

Checking command: 47

Thread[21]: Received from the client: /47

Checking command: /47

Thread[22]: Received from the client: /47

Checking command: /47

Thread[21]: Received from the client: /flip

Checking command: /flip

Checking command: User[21] flipped a coin and got tails!

0

## Numberguesser game (code and output) File: ServerThread.java

### Checklist Items (0)

#### Feature 2 (3 pts.)

^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:

Coin toss command (random heads or tails)  
Command should be something logical like /flip or /toss or /coin or similar

#### 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

```

114 //UCID: NS87
115 //DATE: 2/26/2024
116
117 public class Client {
118
119     public void startNumberGuesser() throws IOException {
120         if (!isConnected()) {
121             out.writeObject(obj("/start"));
122         } else {
123             System.out.println("Not connected to the server");
124         }
125     }
126
127     public void stopNumberGuesser() throws IOException {
128         if (!isConnected()) {
129             out.writeObject(obj("/stop"));
130         } else {
131             System.out.println("Not connected to the server");
132         }
133     }
134
135     public void makeGuess(int guess) throws IOException {
136         if (!isConnected()) {
137             out.writeObject(obj("/guess " + guess));
138         } else {
139             System.out.println("Not connected to the server");
140         }
141     }
142
143     public void flipCoin() throws IOException {
144         if (!isConnected()) {
145             out.writeObject(obj("/flip"));
146         } else {
147             System.out.println("Not connected to the server");
148         }
149     }
150 }
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

## Coin toss command (code and output) File: Client.java

## Checklist Items (0)

```

114 //UCID: NS87
115 //DATE: 2/26/2024
116
117 public class Server {
118
119     private void processCoinToss(String message, long clientId) {
120         if (message.equalsIgnoreCase("flip") || message.equalsIgnoreCase("toss") || message.equalsIgnoreCase("coin")) {
121             String result = (Math.random() < 0.5) ? "heads" : "tails";
122             broadcast("User[" + clientId + "] flipped a coin and got " + result + "!");
123         }
124     }
125
126     protected synchronized void broadcast(String message, long id) {
127         if (processCommand(message, id)) {
128             return;
129         }
130
131         message = String.format("User[%d]: %s", id, message);
132
133         Iterator<ServerThread> it = clients.iterator();
134         while (it.hasNext()) {
135             ServerThread client = it.next();
136             boolean wasSuccessful = client.send(message);
137             if (!wasSuccessful) {
138                 System.out.println("Error: format (Client) disconnected client[%d] from loop. client not[%d]");
139             }
140         }
141     }
142
143     public void start() {
144         // Start the server
145         // ...
146     }
147
148     public void stop() {
149         // Stop the server
150         // ...
151     }
152
153     public void makeGuess(int guess) {
154         // ...
155     }
156
157     public void flipCoin() {
158         // ...
159     }
160
161     public void tossCoin() {
162         // ...
163     }
164
165     public void coinCommand() {
166         // ...
167     }
168
169     public void coinTossCommand() {
170         // ...
171     }
172
173     public void coinTossCommand() {
174         // ...
175     }
176
177     public void coinTossCommand() {
178         // ...
179     }
180
181     public void coinTossCommand() {
182         // ...
183     }
184
185     public void coinTossCommand() {
186         // ...
187     }
188
189     public void coinTossCommand() {
190         // ...
191     }
192
193     public void coinTossCommand() {
194         // ...
195     }
196
197     public void coinTossCommand() {
198         // ...
199     }
200
201     public void coinTossCommand() {
202         // ...
203     }
204
205     public void coinTossCommand() {
206         // ...
207     }
208
209     public void coinTossCommand() {
210         // ...
211     }
212
213     public void coinTossCommand() {
214         // ...
215     }
216
217     public void coinTossCommand() {
218         // ...
219     }
220
221     public void coinTossCommand() {
222         // ...
223     }
224
225     public void coinTossCommand() {
226         // ...
227     }
228
229     public void coinTossCommand() {
230         // ...
231     }
232
233     public void coinTossCommand() {
234         // ...
235     }
236
237     public void coinTossCommand() {
238         // ...
239     }
240
241     public void coinTossCommand() {
242         // ...
243     }
244
245     public void coinTossCommand() {
246         // ...
247     }
248
249     public void coinTossCommand() {
250         // ...
251     }
252
253     public void coinTossCommand() {
254         // ...
255     }
256
257     public void coinTossCommand() {
258         // ...
259     }
260
261     public void coinTossCommand() {
262         // ...
263     }
264
265     public void coinTossCommand() {
266         // ...
267     }
268
269     public void coinTossCommand() {
270         // ...
271     }
272
273     public void coinTossCommand() {
274         // ...
275     }
276
277     public void coinTossCommand() {
278         // ...
279     }
280
281     public void coinTossCommand() {
282         // ...
283     }
284
285     public void coinTossCommand() {
286         // ...
287     }
288
289     public void coinTossCommand() {
290         // ...
291     }
292
293     public void coinTossCommand() {
294         // ...
295     }
296
297     public void coinTossCommand() {
298         // ...
299     }
300
301     public void coinTossCommand() {
302         // ...
303     }
304
305     public void coinTossCommand() {
306         // ...
307     }
308
309     public void coinTossCommand() {
310         // ...
311     }
312
313     public void coinTossCommand() {
314         // ...
315     }
316
317     public void coinTossCommand() {
318         // ...
319     }
320
321     public void coinTossCommand() {
322         // ...
323     }
324
325     public void coinTossCommand() {
326         // ...
327     }
328
329     public void coinTossCommand() {
330         // ...
331     }
332
333     public void coinTossCommand() {
334         // ...
335     }
336
337     public void coinTossCommand() {
338         // ...
339     }
340
341     public void coinTossCommand() {
342         // ...
343     }
344
345     public void coinTossCommand() {
346         // ...
347     }
348
349     public void coinTossCommand() {
350         // ...
351     }
352
353     public void coinTossCommand() {
354         // ...
355     }
356
357     public void coinTossCommand() {
358         // ...
359     }
360
361     public void coinTossCommand() {
362         // ...
363     }
364
365     public void coinTossCommand() {
366         // ...
367     }
368
369     public void coinTossCommand() {
370         // ...
371     }
372
373     public void coinTossCommand() {
374         // ...
375     }
376
377     public void coinTossCommand() {
378         // ...
379     }
380
381     public void coinTossCommand() {
382         // ...
383     }
384
385     public void coinTossCommand() {
386         // ...
387     }
388
389     public void coinTossCommand() {
390         // ...
391     }
392
393     public void coinTossCommand() {
394         // ...
395     }
396
397     public void coinTossCommand() {
398         // ...
399     }
400
401     public void coinTossCommand() {
402         // ...
403     }
404
405     public void coinTossCommand() {
406         // ...
407     }
408
409     public void coinTossCommand() {
410         // ...
411     }
412
413     public void coinTossCommand() {
414         // ...
415     }
416
417     public void coinTossCommand() {
418         // ...
419     }
420
421     public void coinTossCommand() {
422         // ...
423     }
424
425     public void coinTossCommand() {
426         // ...
427     }
428
429     public void coinTossCommand() {
430         // ...
431     }
432
433     public void coinTossCommand() {
434         // ...
435     }
436
437     public void coinTossCommand() {
438         // ...
439     }
440
441     public void coinTossCommand() {
442         // ...
443     }
444
445     public void coinTossCommand() {
446         // ...
447     }
448
449     public void coinTossCommand() {
450         // ...
451     }
452
453     public void coinTossCommand() {
454         // ...
455     }
456
457     public void coinTossCommand() {
458         // ...
459     }
460
461     public void coinTossCommand() {
462         // ...
463     }
464
465     public void coinTossCommand() {
466         // ...
467     }
468
469     public void coinTossCommand() {
470         // ...
471     }
472
473     public void coinTossCommand() {
474         // ...
475     }
476
477     public void coinTossCommand() {
478         // ...
479     }
480
481     public void coinTossCommand() {
482         // ...
483     }
484
485     public void coinTossCommand() {
486         // ...
487     }
488
489     public void coinTossCommand() {
490         // ...
491     }
492
493     public void coinTossCommand() {
494         // ...
495     }
496
497     public void coinTossCommand() {
498         // ...
499     }
500
501     public void coinTossCommand() {
502         // ...
503     }
504
505     public void coinTossCommand() {
506         // ...
507     }
508
509     public void coinTossCommand() {
510         // ...
511     }
512
513     public void coinTossCommand() {
514         // ...
515     }
516
517     public void coinTossCommand() {
518         // ...
519     }
520
521     public void coinTossCommand() {
522         // ...
523     }
524
525     public void coinTossCommand() {
526         // ...
527     }
528
529     public void coinTossCommand() {
530         // ...
531     }
532
533     public void coinTossCommand() {
534         // ...
535     }
536
537     public void coinTossCommand() {
538         // ...
539     }
540
541     public void coinTossCommand() {
542         // ...
543     }
544
545     public void coinTossCommand() {
546         // ...
547     }
548
549     public void coinTossCommand() {
550         // ...
551     }
552
553     public void coinTossCommand() {
554         // ...
555     }
556
557     public void coinTossCommand() {
558         // ...
559     }
560
561     public void coinTossCommand() {
562         // ...
563     }
564
565     public void coinTossCommand() {
566         // ...
567     }
568
569     public void coinTossCommand() {
570         // ...
571     }
572
573     public void coinTossCommand() {
574         // ...
575     }
576
577     public void coinTossCommand() {
578         // ...
579     }
580
581     public void coinTossCommand() {
582         // ...
583     }
584
585     public void coinTossCommand() {
586         // ...
587     }
588
589     public void coinTossCommand() {
590         // ...
591     }
592
593     public void coinTossCommand() {
594         // ...
595     }
596
597     public void coinTossCommand() {
598         // ...
599     }
600
601     public void coinTossCommand() {
602         // ...
603     }
604
605     public void coinTossCommand() {
606         // ...
607     }
608
609     public void coinTossCommand() {
610         // ...
611     }
612
613     public void coinTossCommand() {
614         // ...
615     }
616
617     public void coinTossCommand() {
618         // ...
619     }
620
621     public void coinTossCommand() {
622         // ...
623     }
624
625     public void coinTossCommand() {
626         // ...
627     }
628
629     public void coinTossCommand() {
630         // ...
631     }
632
633     public void coinTossCommand() {
634         // ...
635     }
636
637     public void coinTossCommand() {
638         // ...
639     }
640
641     public void coinTossCommand() {
642         // ...
643     }
644
645     public void coinTossCommand() {
646         // ...
647     }
648
649     public void coinTossCommand() {
650         // ...
651     }
652
653     public void coinTossCommand() {
654         // ...
655     }
656
657     public void coinTossCommand() {
658         // ...
659     }
660
661     public void coinTossCommand() {
662         // ...
663     }
664
665     public void coinTossCommand() {
666         // ...
667     }
668
669     public void coinTossCommand() {
670         // ...
671     }
672
673     public void coinTossCommand() {
674         // ...
675     }
676
677     public void coinTossCommand() {
678         // ...
679     }
680
681     public void coinTossCommand() {
682         // ...
683     }
684
685     public void coinTossCommand() {
686         // ...
687     }
688
689     public void coinTossCommand() {
690         // ...
691     }
692
693     public void coinTossCommand() {
694         // ...
695     }
696
697     public void coinTossCommand() {
698         // ...
699     }
700
701     public void coinTossCommand() {
702         // ...
703     }
704
705     public void coinTossCommand() {
706         // ...
707     }
708
709     public void coinTossCommand() {
710         // ...
711     }
712
713     public void coinTossCommand() {
714         // ...
715     }
716
717     public void coinTossCommand() {
718         // ...
719     }
720
721     public void coinTossCommand() {
722         // ...
723     }
724
725     public void coinTossCommand() {
726         // ...
727     }
728
729     public void coinTossCommand() {
730         // ...
731     }
732
733     public void coinTossCommand() {
734         // ...
735     }
736
737     public void coinTossCommand() {
738         // ...
739     }
740
741     public void coinTossCommand() {
742         // ...
743     }
744
745     public void coinTossCommand() {
746         // ...
747     }
748
749     public void coinTossCommand() {
750         // ...
751     }
752
753     public void coinTossCommand() {
754         // ...
755     }
756
757     public void coinTossCommand() {
758         // ...
759     }
760
761     public void coinTossCommand() {
762         // ...
763     }
764
765     public void coinTossCommand() {
766         // ...
767     }
768
769     public void coinTossCommand() {
770         // ...
771     }
772
773     public void coinTossCommand() {
774         // ...
775     }
776
777     public void coinTossCommand() {
778         // ...
779     }
780
781     public void coinTossCommand() {
782         // ...
783     }
784
785     public void coinTossCommand() {
786         // ...
787     }
788
789     public void coinTossCommand() {
790         // ...
791     }
792
793     public void coinTossCommand() {
794         // ...
795     }
796
797     public void coinTossCommand() {
798         // ...
799     }
800
801     public void coinTossCommand() {
802         // ...
803     }
804
805     public void coinTossCommand() {
806         // ...
807     }
808
809     public void coinTossCommand() {
810         // ...
811     }
812
813     public void coinTossCommand() {
814         // ...
815     }
816
817     public void coinTossCommand() {
818         // ...
819     }
820
821     public void coinTossCommand() {
822         // ...
823     }
824
825     public void coinTossCommand() {
826         // ...
827     }
828
829     public void coinTossCommand() {
830         // ...
831     }
832
833     public void coinTossCommand() {
834         // ...
835     }
836
837     public void coinTossCommand() {
838         // ...
839     }
840
841     public void coinTossCommand() {
842         // ...
843     }
844
845     public void coinTossCommand() {
846         // ...
847     }
848
849     public void coinTossCommand() {
850         // ...
851     }
852
853     public void coinTossCommand() {
854         // ...
855     }
856
857     public void coinTossCommand() {
858         // ...
859     }
860
861     public void coinTossCommand() {
862         // ...
863     }
864
865     public void coinTossCommand() {
866         // ...
867     }
868
869     public void coinTossCommand() {
870         // ...
871     }
872
873     public void coinTossCommand() {
874         // ...
875     }
876
877     public void coinTossCommand() {
878         // ...
879     }
880
881     public void coinTossCommand() {
882         // ...
883     }
884
885     public void coinTossCommand() {
886         // ...
887     }
888
889     public void coinTossCommand() {
890         // ...
891     }
892
893     public void coinTossCommand() {
894         // ...
895     }
896
897     public void coinTossCommand() {
898         // ...
899     }
900
901     public void coinTossCommand() {
902         // ...
903     }
904
905     public void coinTossCommand() {
906         // ...
907     }
908
909     public void coinTossCommand() {
910         // ...
911     }
912
913     public void coinTossCommand() {
914         // ...
915     }
916
917     public void coinTossCommand() {
918         // ...
919     }
920
921     public void coinTossCommand() {
922         // ...
923     }
924
925     public void coinTossCommand() {
926         // ...
927     }
928
929     public void coinTossCommand() {
930         // ...
931     }
932
933     public void coinTossCommand() {
934         // ...
935     }
936
937     public void coinTossCommand() {
938         // ...
939     }
940
941     public void coinTossCommand() {
942         // ...
943     }
944
945     public void coinTossCommand() {
946         // ...
947     }
948
949     public void coinTossCommand() {
950         // ...
951     }
952
953     public void coinTossCommand() {
954         // ...
955     }
956
957     public void coinTossCommand() {
958         // ...
959     }
960
961     public void coinTossCommand() {
962         // ...
963     }
964
965     public void coinTossCommand() {
966         // ...
967     }
968
969     public void coinTossCommand() {
970         // ...
971     }
972
973     public void coinTossCommand() {
974         // ...
975     }
976
977     public void coinTossCommand() {
978         // ...
979     }
980
981     public void coinTossCommand() {
982         // ...
983     }
984
985     public void coinTossCommand() {
986         // ...
987     }
988
989     public void coinTossCommand() {
990         // ...
991     }
992
993     public void coinTossCommand() {
994         // ...
995     }
996
997     public void coinTossCommand() {
998         // ...
999     }
1000

```

```
100     System.out.println(String.format("Removing disconnected client[%d] from list", client.getId()));
101     it.remove();
102     broadcast(message:"Disconnected", id);
103 }
104 }
105
106 if (!isGameActive) {
107     processGuess(message, id);
108 } else {
109     processCoinToss(message, id);
110 }
111 }
112 }
113 }
114 }
115 }
116 }
117 }
118 }
119 }
120 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Waiting for input  
/start  
Waiting for input  
User[1]: Number guesser game has started! Guess a number between 0 and 5  
0.  
47  
Waiting for input  
User[21]: 47  
/47  
Waiting for input  
User[21]: /47  
User[22]: /47  
/flip  
Waiting for input  
User[21]: /flip  
User[1]: User[21] flipped a coin and got Tails!

Waiting for input  
connect localhost: 3000  
Not connected to server  
Waiting for input  
connect localhost:3000  
Client connected  
Waiting for input  
User[1]: Number guesser game has started! Guess a number between 0 and 5  
0.  
User[21]: 47  
User[21]: /47  
/47  
Waiting for input  
User[22]: /47  
User[21]: /flip  
User[1]: User[21] flipped a coin and got Tails!

Client connected  
Thread[22]: Thread created  
Thread[22]: Thread starting  
Thread[21]: Received from the client: /start  
Checking command: /start  
Checking command: Number guesser game has started! Guess a number between  
n 0 and 50.  
Thread[21]: Received from the client: 47  
Checking command: 47  
Thread[21]: Received from the client: /47  
Checking command: /47  
Thread[22]: Received from the client: /47  
Checking command: /47  
Thread[21]: Received from the client: /flip  
Checking command: /flip  
Checking command: User[21] flipped a coin and got Tails!

## Coin toss command (code and output) File: Server.java

### Checklist Items (0)

```
114 > Modules > PortCopy > 2 ServerThread.java > 2 ServerThread > 2 processNumberGuesserCommand(String)
115 public class ServerThread extends Thread {
116     public void run() {
117     }
118 }
119
120 private void cleanup() {
121 }
122
123 //AUCSID: NS87
124 //DATE: 2/21/24
125 private void processNumberGuesserCommand(String command) {
126     if (command.equalsIgnoreCase("start")) {
127         server.broadcast(message:"Number guesser game has started! Guess a number between 0 and 50.", -1);
128         isNumberGuesserActive = true;
129     } else if (command.equalsIgnoreCase("stop")) {
130         server.broadcast(message:"Number guesser game has stopped. Guesses will now be treated as regular messages.", -1);
131         isNumberGuesserActive = false;
132     } else if (command.startsWith("pass")) {
133         int guessNumber = Integer.parseInt(command.split(" ")[1]);
134     }
135 }
136
137 private void processCoinTossCommand(String command) {
138     if (command.equalsIgnoreCase("toss") || command.equalsIgnoreCase("flip") || command.equalsIgnoreCase("coin")) {
139         String result = (Math.random() < 0.5) ? "heads" : "tails";
140         server.broadcast("User[" + getId() + "] flipped a coin and got " + result + "!", -1);
141     }
142 }
143 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Waiting for input  
/start  
Waiting for input  
User[1]: Number guesser game has started! Guess a number between 0 and 5  
0.  
47  
Waiting for input  
User[21]: 47  
/47  
Waiting for input  
User[21]: /47  
User[22]: /47  
/flip  
Waiting for input  
User[21]: /flip  
User[1]: User[21] flipped a coin and got Tails!

Waiting for input  
connect localhost: 3000  
Not connected to server  
Waiting for input  
connect localhost:3000  
Client connected  
Waiting for input  
User[1]: Number guesser game has started! Guess a number between 0 and 5  
0.  
User[21]: 47  
User[21]: /47  
/47  
Waiting for input  
User[22]: /47  
User[21]: /flip  
User[1]: User[21] flipped a coin and got Tails!

Client connected  
Thread[22]: Thread created  
Thread[22]: Thread starting  
Thread[21]: Received from the client: /start  
Checking command: /start  
Checking command: Number guesser game has started! Guess a number between  
n 0 and 50.  
Thread[21]: Received from the client: 47  
Checking command: 47  
Thread[21]: Received from the client: /47  
Checking command: /47  
Thread[22]: Received from the client: /47  
Checking command: /47  
Thread[21]: Received from the client: /flip  
Checking command: /flip  
Checking command: User[21] flipped a coin and got Tails!

## Coin toss command (code and output) File: ServerThread.java

### 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^



^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?

#### Checklist

\*The checkboxes are for your own tracking

#	Points	Details
<input type="checkbox"/> #1	1	An issue or learning is clearly stated
<input type="checkbox"/> #2	1	Response is a few reasonable sentences

Response:

During the assignment, I introduced features like a number guesser and a flip command to enhance the functionality of the Java client-server application. The number guesser allows clients to interactively guess a number while flip command enables clients to flip a coin.

^COLLAPSE ^

### 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/Noaman4/Java-Sockets-Part-1-3-HW/pull/1>

End of Assignment