SER 321 B Session

SI Session

Monday, October 30th 2023

4:00 - 5:00 pm MST

Agenda

- Making your Server Robust
- Protocol Organization
- Socket Programming

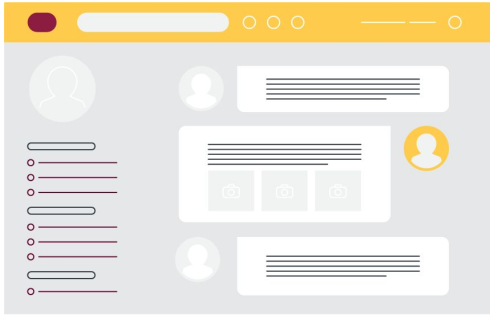
SI Session Expectations

Thanks for coming to the **SER 321** SI session. We have a packed agenda and we are going to try to get through as many of our planned example problems as possible. This session will be recorded and shared with others.

- If after this you want to see additional examples, please visit the drop-in tutoring center.
- We will post the link in the chat now and at the end of the session.
    - tutoring.asu.edu
- Please keep in mind we are recording this session and it will be made available for you to review 24-48 hours after this session concludes.
- Finally, please be respectful to each other during the session.

Interact with us:

Zoom Features



Zoom Chat

- Use the chat feature to interact with the presenter and respond to presenter's questions.
- Annotations are encouraged

SER 321

Making your Code Robust

For starters, what do we mean everytime we say, “**Your code needs to be robust,**” or, “**make sure your code is robust,**”?

Error Handling

SER 321

Making your Code Robust

For starters, what do we mean everytime we say, "**Your code needs to be robust,**" or, "**make sure your code is robust,**"?

Your server should ***not*** crash!

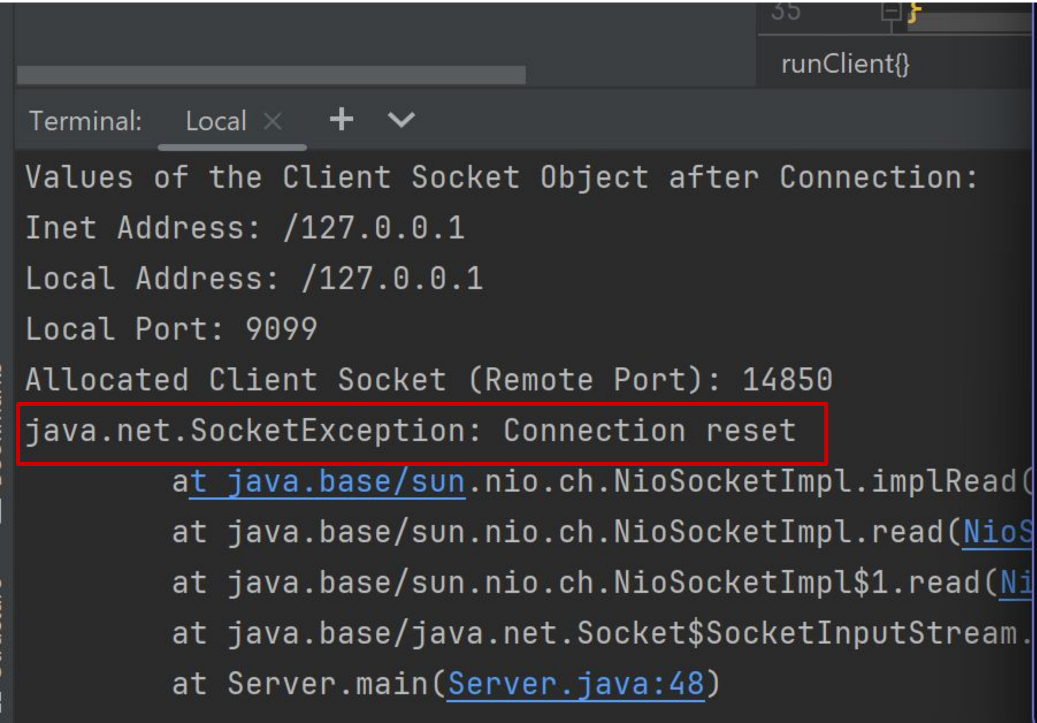
What if the client sends a bad request?

What if the client disconnects abruptly?

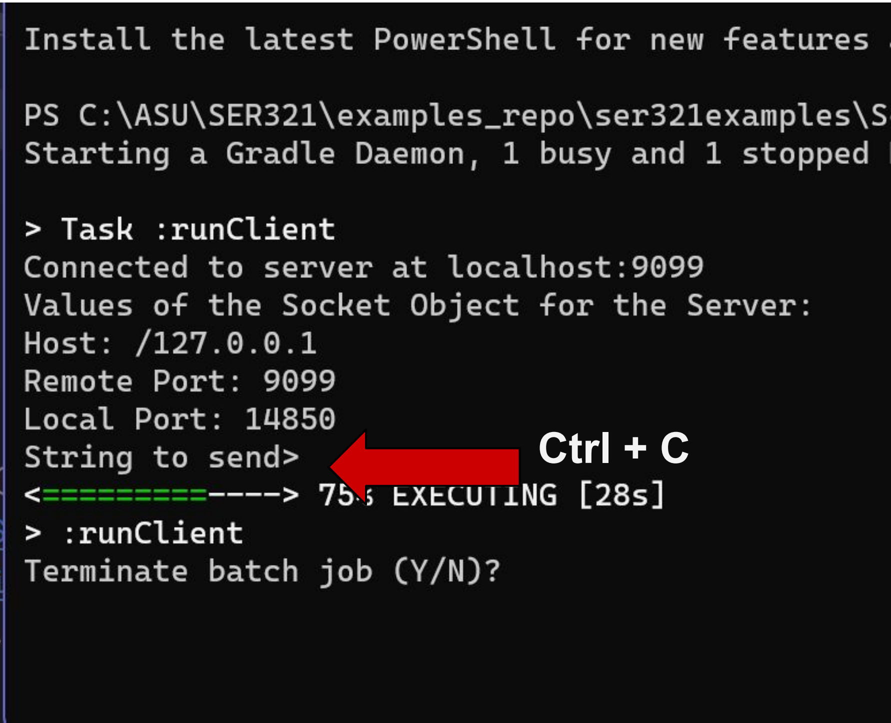
Echo_Java from course Repo

SER 321

Examples that need more Error Handling



```
Values of the Client Socket Object after Connection:
Inet Address: /127.0.0.1
Local Address: /127.0.0.1
Local Port: 9099
Allocated Client Socket (Remote Port): 14850
java.net.SocketException: Connection reset
        at java.base/sun.nio.ch.NioSocketImpl.implRead(
        at java.base/sun.nio.ch.NioSocketImpl.read(NioS
        at java.base/sun.nio.ch.NioSocketImpl$1.read(Ni
        at java.base/java.net.Socket$SocketInputStream.
        at Server.main(Server.java:48)
```



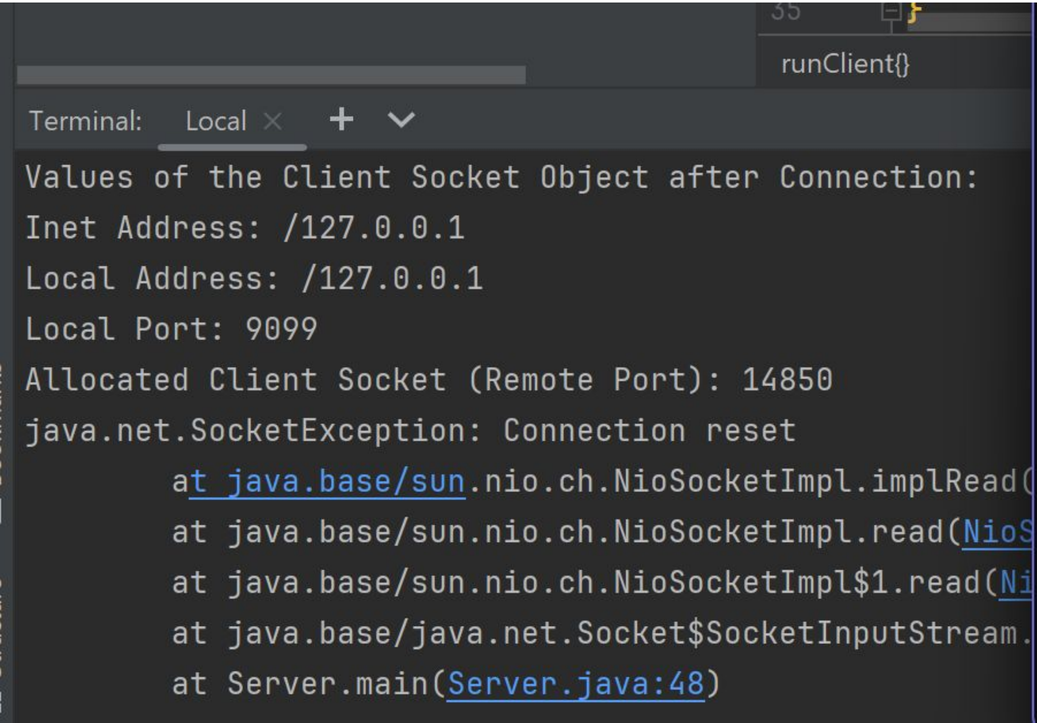
```
Install the latest PowerShell for new features

PS C:\ASU\SER321\examples_repo\ser321examples\S
Starting a Gradle Daemon, 1 busy and 1 stopped

> Task :runClient
Connected to server at localhost:9099
Values of the Socket Object for the Server:
Host: /127.0.0.1
Remote Port: 9099
Local Port: 14850
String to send>
<=========----> 75% EXECUTING [28s]
> :runClient
Terminate batch job (Y/N)?
```

Echo_Java from course Repo

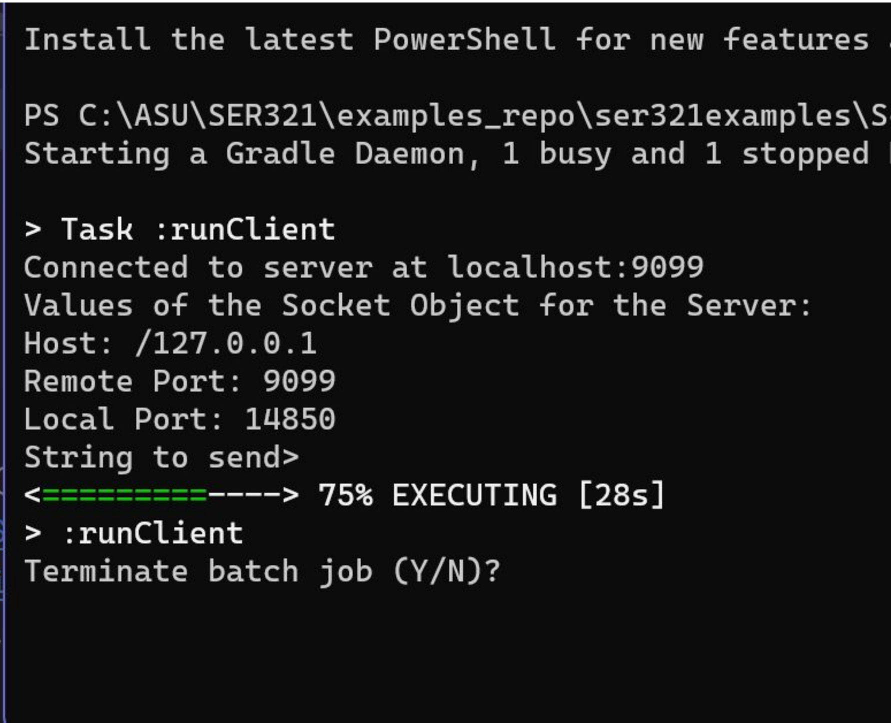
SER 321

Examples that need more Error Handling

Solutions or Ideas?



```
35
runClient{}
Terminal:   Local ×   +  ⌄
Values of the Client Socket Object after Connection:
Inet Address: /127.0.0.1
Local Address: /127.0.0.1
Local Port: 9099
Allocated Client Socket (Remote Port): 14850
java.net.SocketException: Connection reset
        at java.base/sun.nio.ch.NioSocketImpl.implRead(
        at java.base/sun.nio.ch.NioSocketImpl.read(NioS
        at java.base/sun.nio.ch.NioSocketImpl$1.read(Ni
        at java.base/java.net.Socket$SocketInputStream.
        at Server.main(Server.java:48)
```



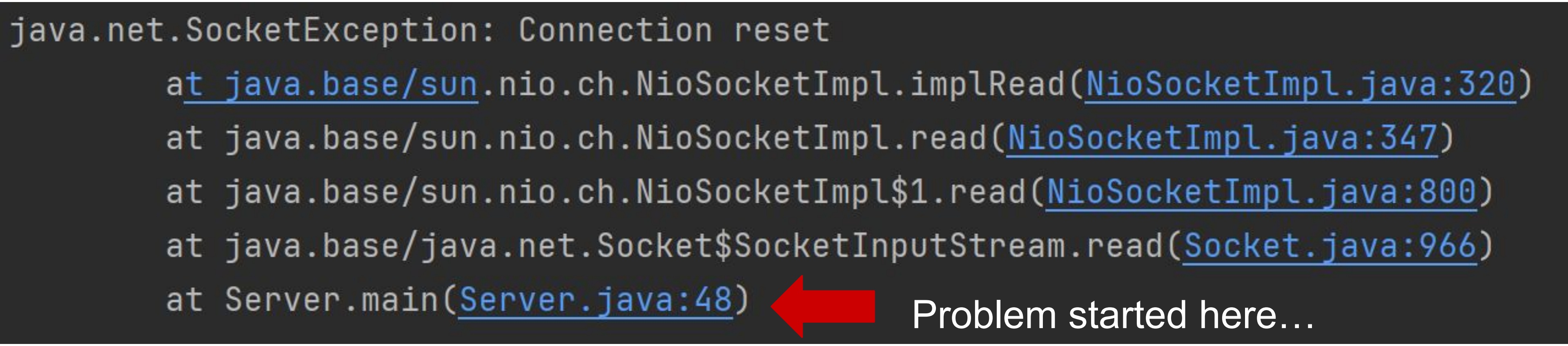
```
Install the latest PowerShell for new features

PS C:\ASU\SER321\examples_repo\ser321examples\S
Starting a Gradle Daemon, 1 busy and 1 stopped

> Task :runClient
Connected to server at localhost:9099
Values of the Socket Object for the Server:
Host: /127.0.0.1
Remote Port: 9099
Local Port: 14850
String to send>
<=========----> 75% EXECUTING [28s]
> :runClient
Terminate batch job (Y/N)?
```

Echo_Java from course Repo

SER 321

Examples that need more Error Handling

Solutions or Ideas?



```
java.net.SocketException: Connection reset
	at java.base/sun.nio.ch.NioSocketImpl.implRead(NioSocketImpl.java:320)
	at java.base/sun.nio.ch.NioSocketImpl.read(NioSocketImpl.java:347)
	at java.base/sun.nio.ch.NioSocketImpl$1.read(NioSocketImpl.java:800)
	at java.base/java.net.Socket$SocketInputStream.read(Socket.java:966)
	at Server.main(Server.java:48)
```

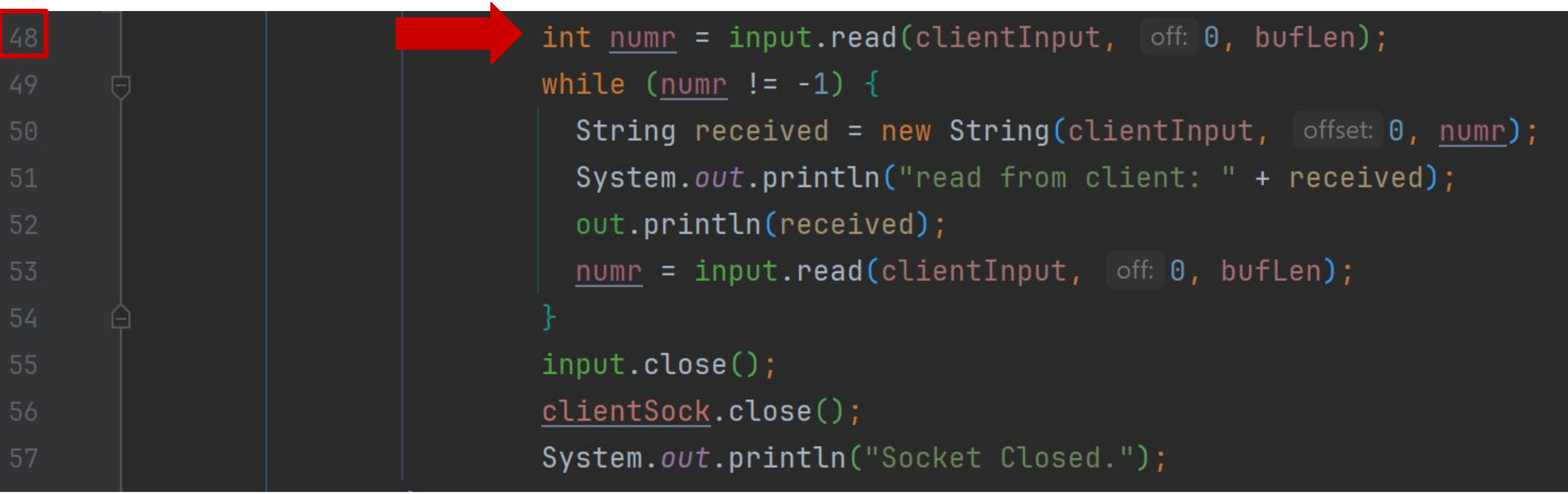
Problem started here…

Echo_Java from course Repo

SER 321

Examples that need more Error Handling

What happened?



```java
int numr = input.read(clientInput, off: 0, bufLen);
while (numr != -1) {
    String received = new String(clientInput, offset: 0, numr);
    System.out.println("read from client: " + received);
    out.println(received);
    numr = input.read(clientInput, off: 0, bufLen);
}
input.close();
clientSock.close();
System.out.println("Socket Closed.");
```

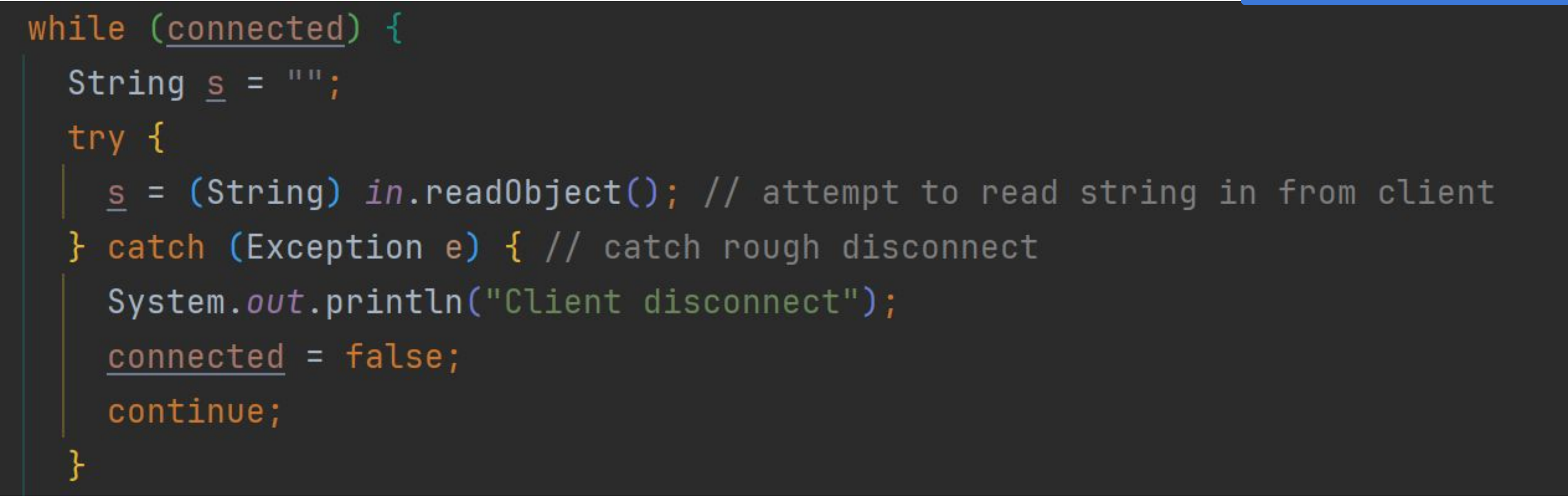
So how do we fix it?

Assignment 3 Part 1 Starter Code

SER 321

Examples that need more Error Handling

Thoughts?



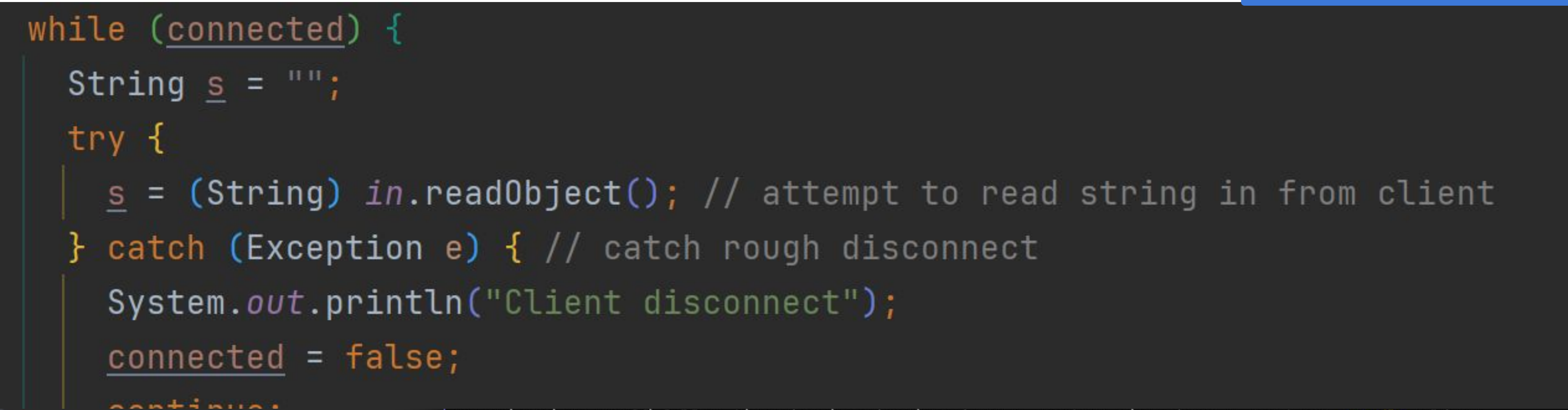
```java
while (connected) {
  String s = "";
  try {
    s = (String) in.readObject(); // attempt to read string in from client
  } catch (Exception e) { // catch rough disconnect
    System.out.println("Client disconnect");
    connected = false;
    continue;
  }
```

Assignment 3 Part 1 Starter Code

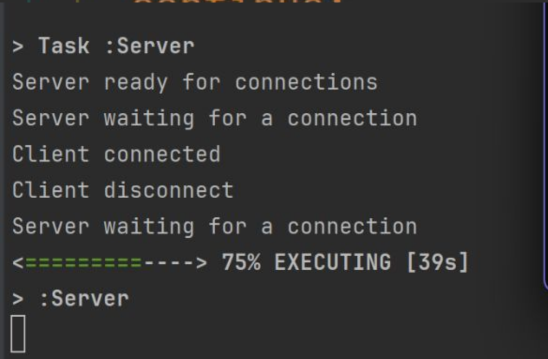
SER 321

Examples that need more Error Handling

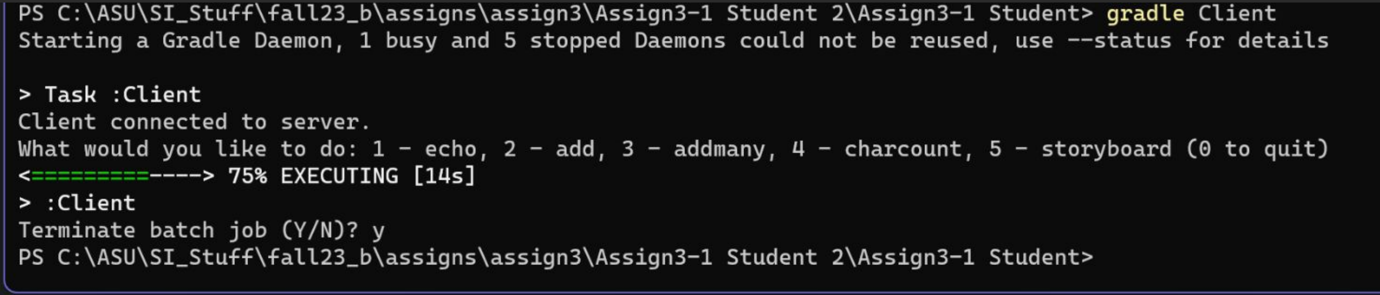
Thoughts?



```java
while (connected) {
  String s = "";
  try {
    s = (String) in.readObject(); // attempt to read string in from client
  } catch (Exception e) { // catch rough disconnect
    System.out.println("Client disconnect");
    connected = false;
    continue;
```



```
> Task :Server
Server ready for connections
Server waiting for a connection
Client connected
Client disconnect
Server waiting for a connection
<=========----> 75% EXECUTING [39s]
> :Server
```



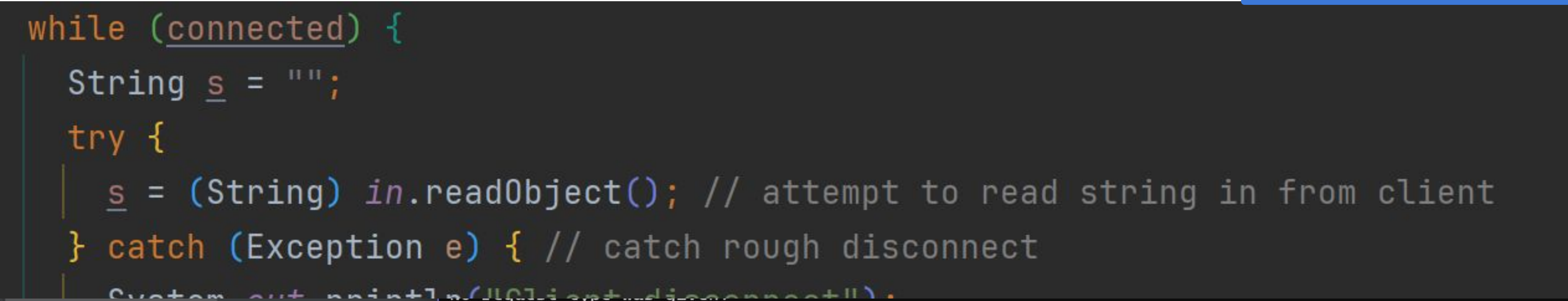
```
PS C:\ASU\SI_Stuff\fall23_b\assigns\assign3\Assign3-1 Student 2\Assign3-1 Student> gradle Client
Starting a Gradle Daemon, 1 busy and 5 stopped Daemons could not be reused, use --status for details

> Task :Client
Client connected to server.
What would you like to do: 1 - echo, 2 - add, 3 - addmany, 4 - charcount, 5 - storyboard (0 to quit)
<=========----> 75% EXECUTING [14s]
> :Client
Terminate batch job (Y/N)? y
PS C:\ASU\SI_Stuff\fall23_b\assigns\assign3\Assign3-1 Student 2\Assign3-1 Student>
```

Assignment 3 Part 1 Starter Code

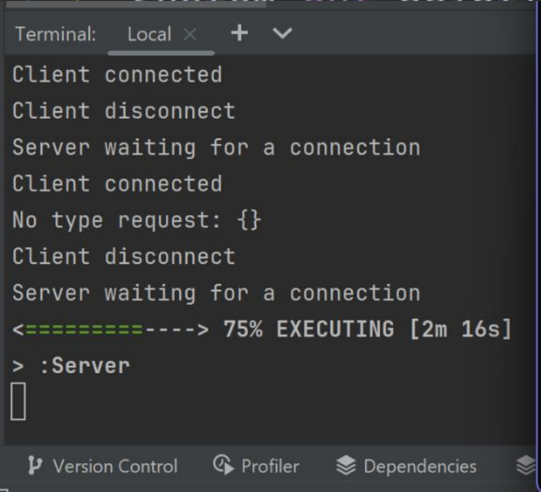
SER 321

Examples that need more Error Handling

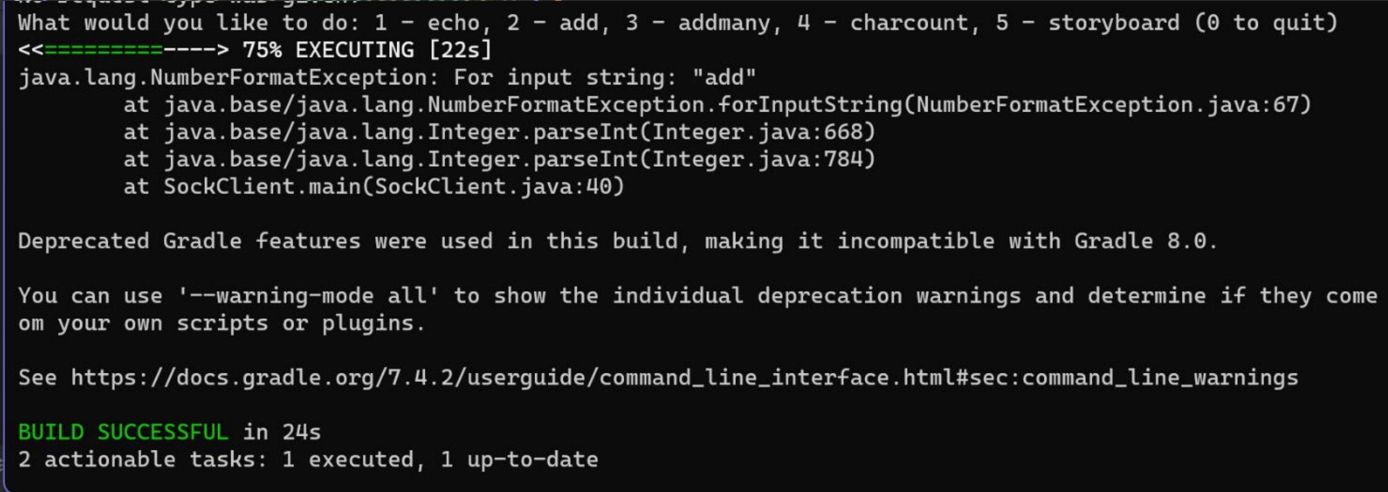
Thoughts?



```
while (connected) {
  String s = "";
  try {
    s = (String) in.readObject(); // attempt to read string in from client
  } catch (Exception e) { // catch rough disconnect
```



```
Client connected
Client disconnect
Server waiting for a connection
Client connected
No type request: {}
Client disconnect
Server waiting for a connection
<=========----> 75% EXECUTING [2m 16s]
> :Server
```



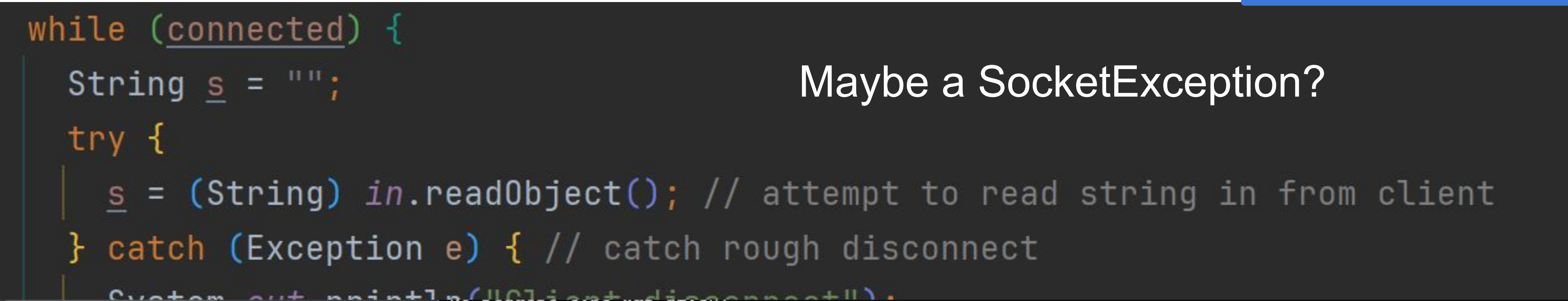
```
What would you like to do: 1 - echo, 2 - add, 3 - addmany, 4 - charcount, 5 - storyboard (0 to quit)
<<=========----> 75% EXECUTING [22s]
java.lang.NumberFormatException: For input string: "add"
        at java.base/java.lang.NumberFormatException.forInputString(NumberFormatException.java:67)
        at java.base/java.lang.Integer.parseInt(Integer.java:668)
        at java.base/java.lang.Integer.parseInt(Integer.java:784)
        at SockClient.main(SockClient.java:40)

Deprecated Gradle features were used in this build, making it incompatible with Gradle 8.0.

You can use '--warning-mode all' to show the individual deprecation warnings and determine if they come
om your own scripts or plugins.

See https://docs.gradle.org/7.4.2/userguide/command_line_interface.html#sec:command_line_warnings

BUILD SUCCESSFUL in 24s
2 actionable tasks: 1 executed, 1 up-to-date
```

Assignment 3 Part 1 Starter Code

SER 321

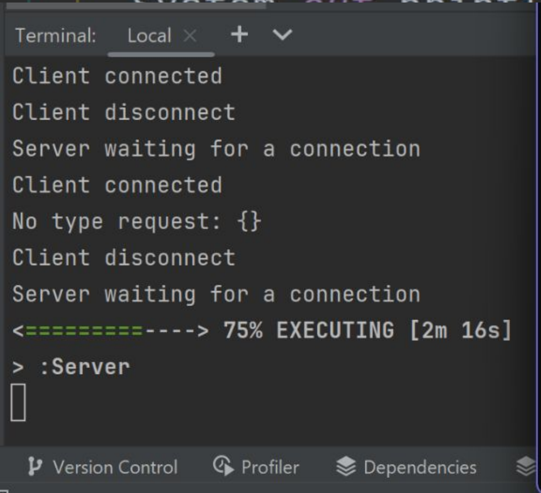
Examples that need more Error Handling

Thoughts?

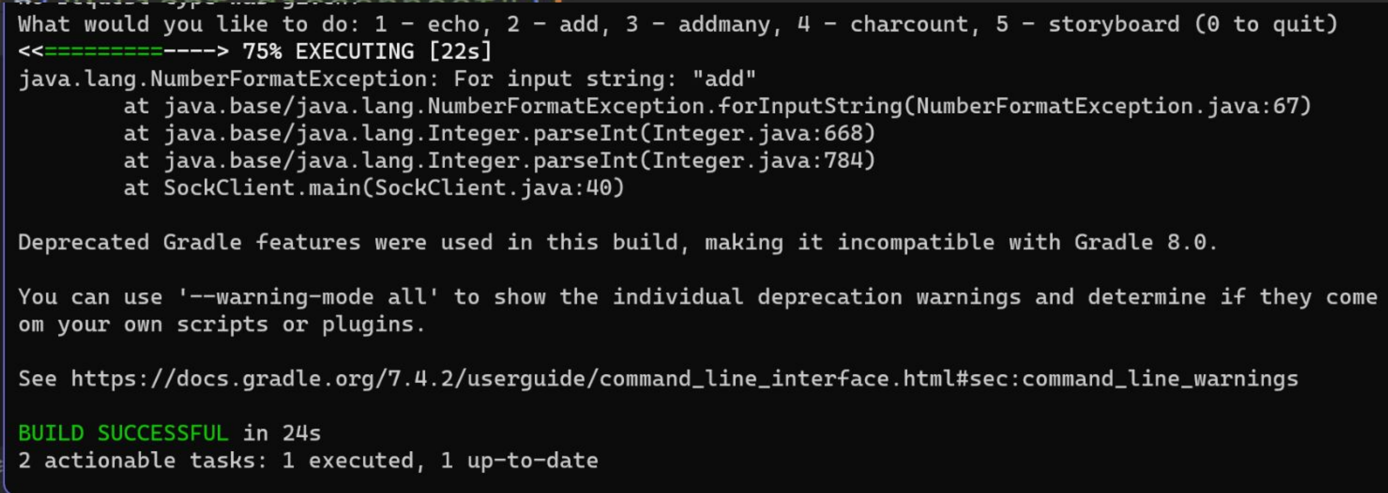


```
while (connected) {
  String s = "";
  try {
    s = (String) in.readObject(); // attempt to read string in from client
  } catch (Exception e) { // catch rough disconnect
    System.out.println("Client disconnect");
```

Maybe a SocketException?



```
Client connected
Client disconnect
Server waiting for a connection
Client connected
No type request: {}
Client disconnect
Server waiting for a connection
<=========----> 75% EXECUTING [2m 16s]
> :Server
```

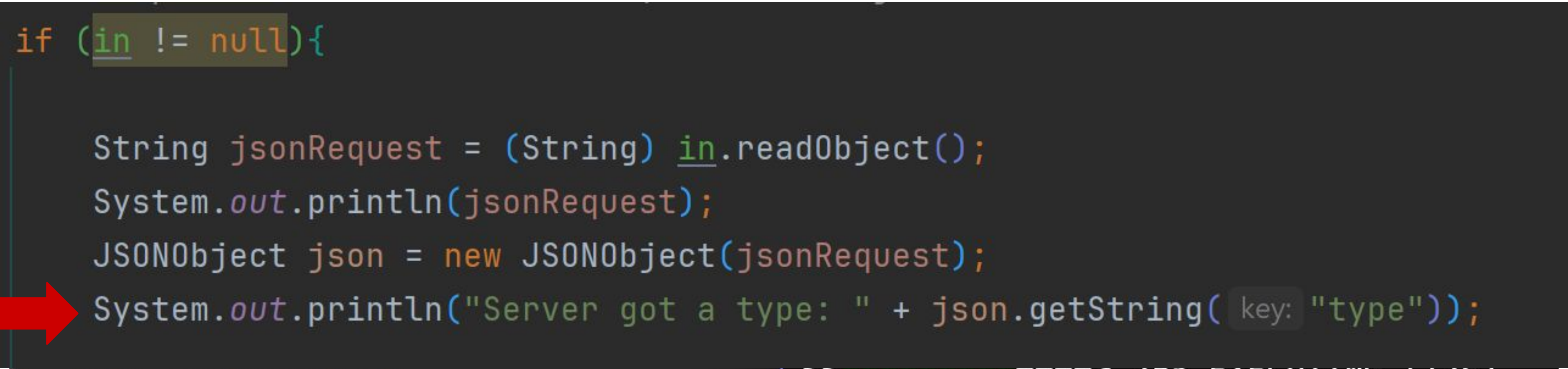


```
What would you like to do: 1 - echo, 2 - add, 3 - addmany, 4 - charcount, 5 - storyboard (0 to quit)
<<=========----> 75% EXECUTING [22s]
java.lang.NumberFormatException: For input string: "add"
        at java.base/java.lang.NumberFormatException.forInputString(NumberFormatException.java:67)
        at java.base/java.lang.Integer.parseInt(Integer.java:668)
        at java.base/java.lang.Integer.parseInt(Integer.java:784)
        at SockClient.main(SockClient.java:40)

Deprecated Gradle features were used in this build, making it incompatible with Gradle 8.0.

You can use '--warning-mode all' to show the individual deprecation warnings and determine if they come
om your own scripts or plugins.

See https://docs.gradle.org/7.4.2/userguide/command_line_interface.html#sec:command_line_warnings

BUILD SUCCESSFUL in 24s
2 actionable tasks: 1 executed, 1 up-to-date
```

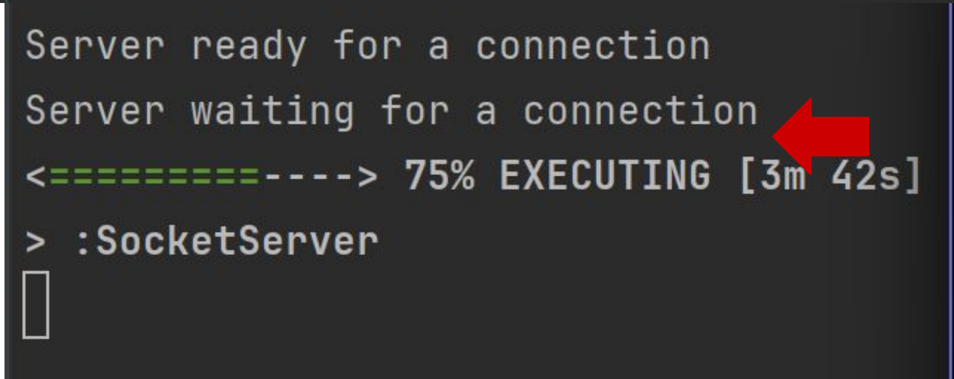
JavaSocketJSONProtocol from course Repo

SER 321

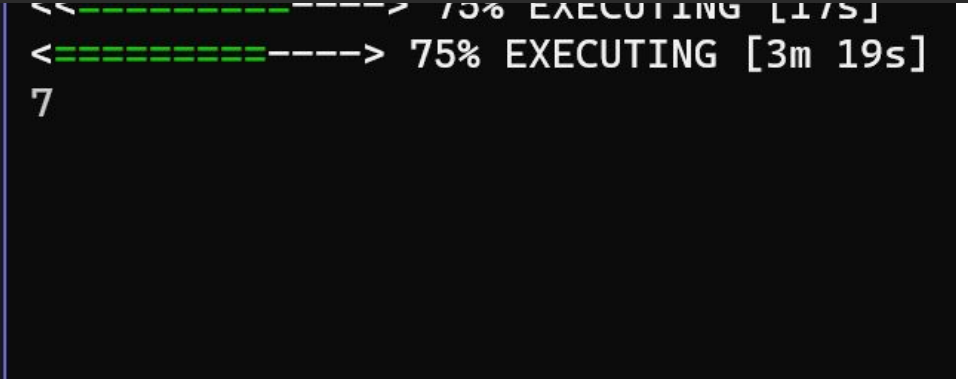
Examples that need more Error Handling



```
if (in != null){

    String jsonRequest = (String) in.readObject();
    System.out.println(jsonRequest);
    JSONObject json = new JSONObject(jsonRequest);
    System.out.println("Server got a type: " + json.getString( key: "type"));
```



```
Server ready for a connection
Server waiting for a connection
<=========----> 75% EXECUTING [3m 42s]
> :SocketServer
```

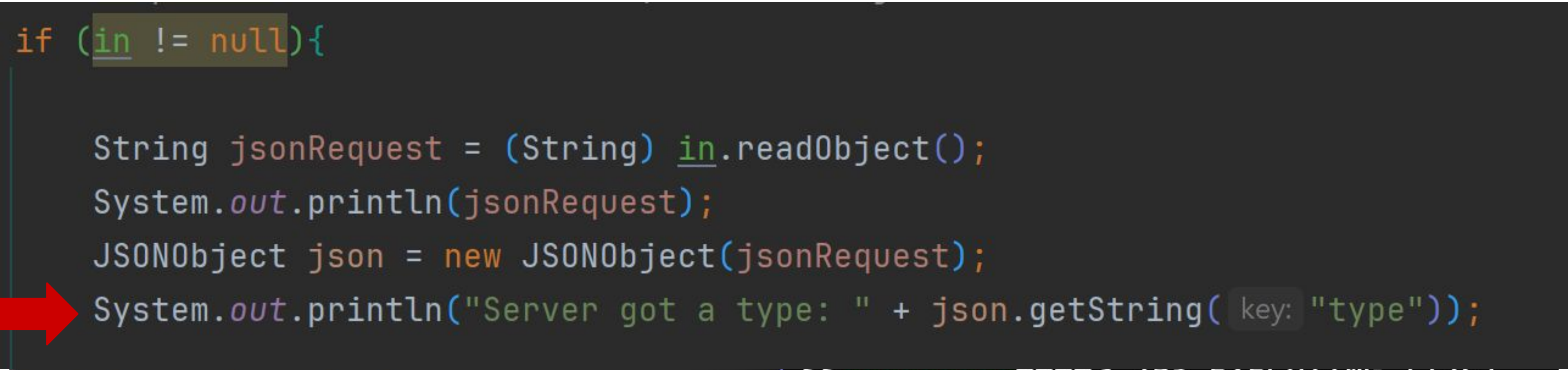


```
<<=========----> 75% EXECUTING [17s]
<=========----> 75% EXECUTING [3m 19s]
7
```

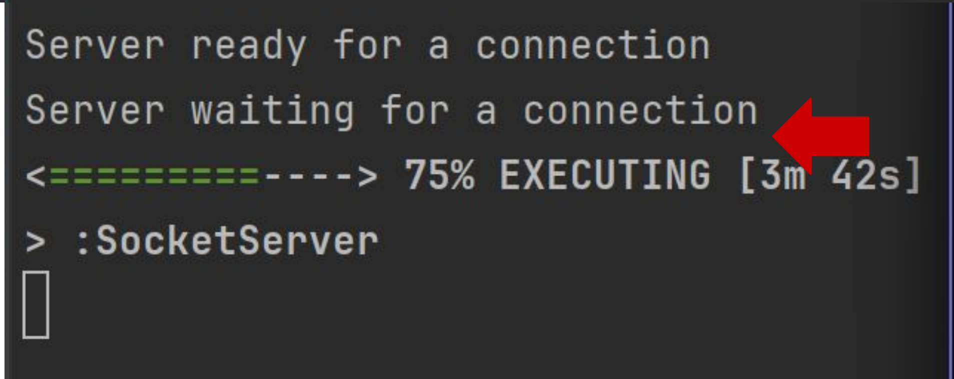
JavaSocketJSONProtocol from course Repo

SER 321

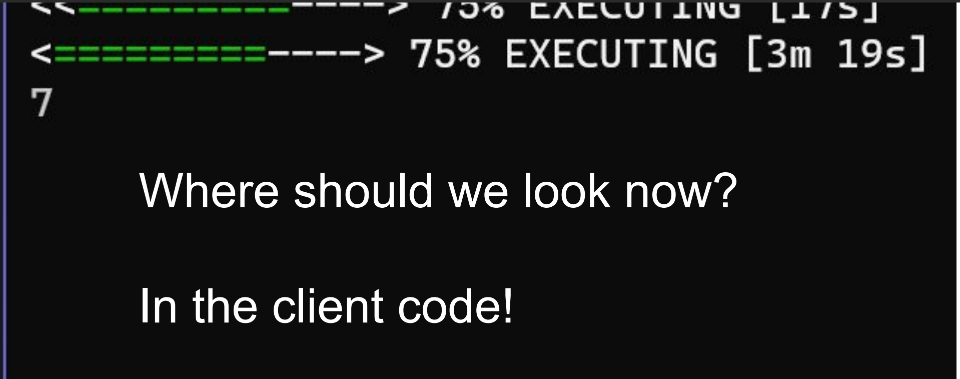
Examples that need more Error Handling



```
if (in != null){

    String jsonRequest = (String) in.readObject();
    System.out.println(jsonRequest);
    JSONObject json = new JSONObject(jsonRequest);
    System.out.println("Server got a type: " + json.getString( key: "type"));
```



```
Server ready for a connection
Server waiting for a connection
<=========----> 75% EXECUTING [3m 42s]
> :SocketServer
```



```
<=========----> 75% EXECUTING [17s]
<=========----> 75% EXECUTING [3m 19s]
7
```

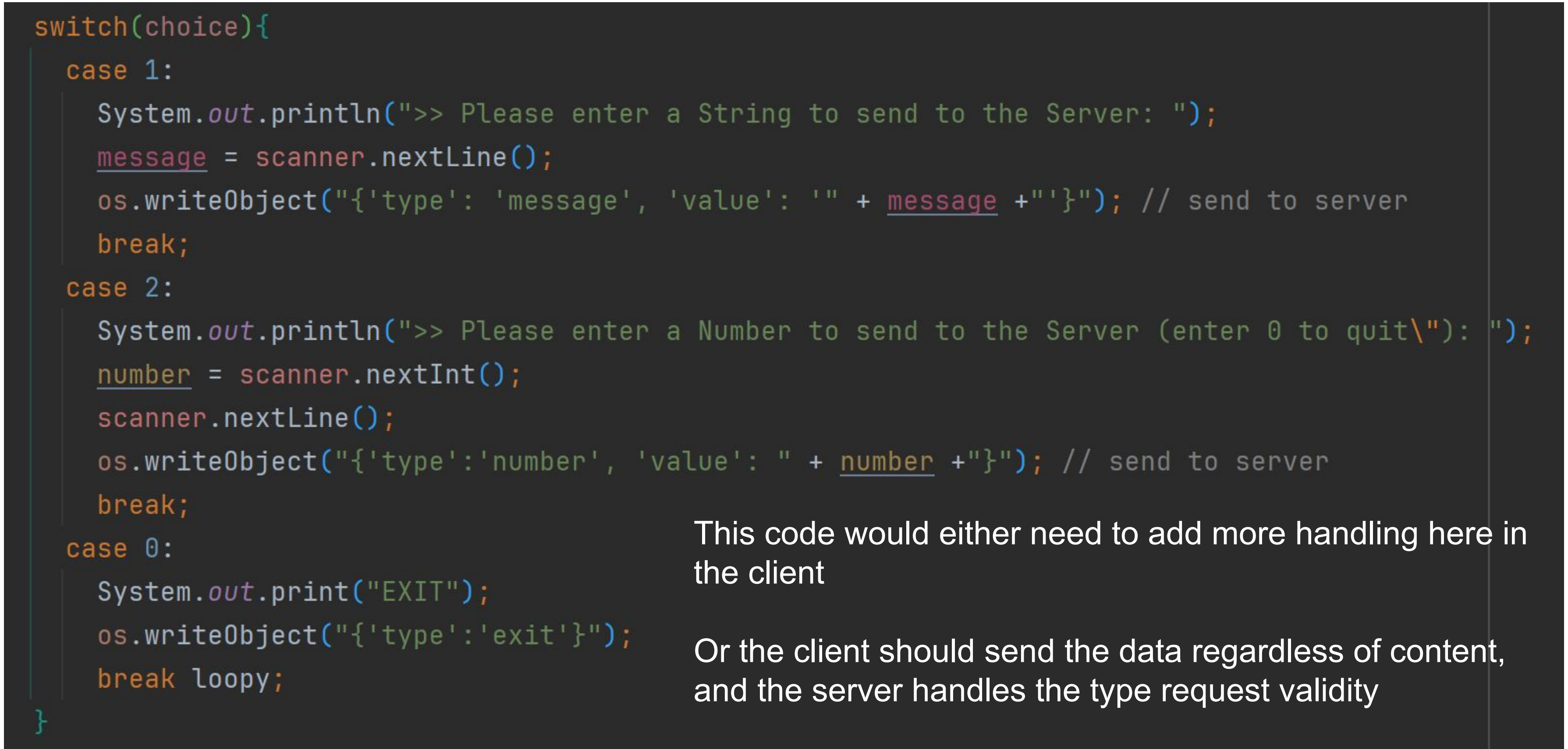
Where should we look now?

In the client code!

JavaSocketJSONProtocol from course Repo

SER 321

Examples that need more Error Handling



```
switch(choice){
  case 1:
    System.out.println(">> Please enter a String to send to the Server: ");
    message = scanner.nextLine();
    os.writeObject("{'type': 'message', 'value': '" + message +"'}"); // send to server
    break;
  case 2:
    System.out.println(">> Please enter a Number to send to the Server (enter 0 to quit\"): ");
    number = scanner.nextInt();
    scanner.nextLine();
    os.writeObject("{'type':'number', 'value': " + number +"}"); // send to server
    break;
  case 0:
    System.out.print("EXIT");
    os.writeObject("{'type':'exit'}");
    break loopy;
}
```

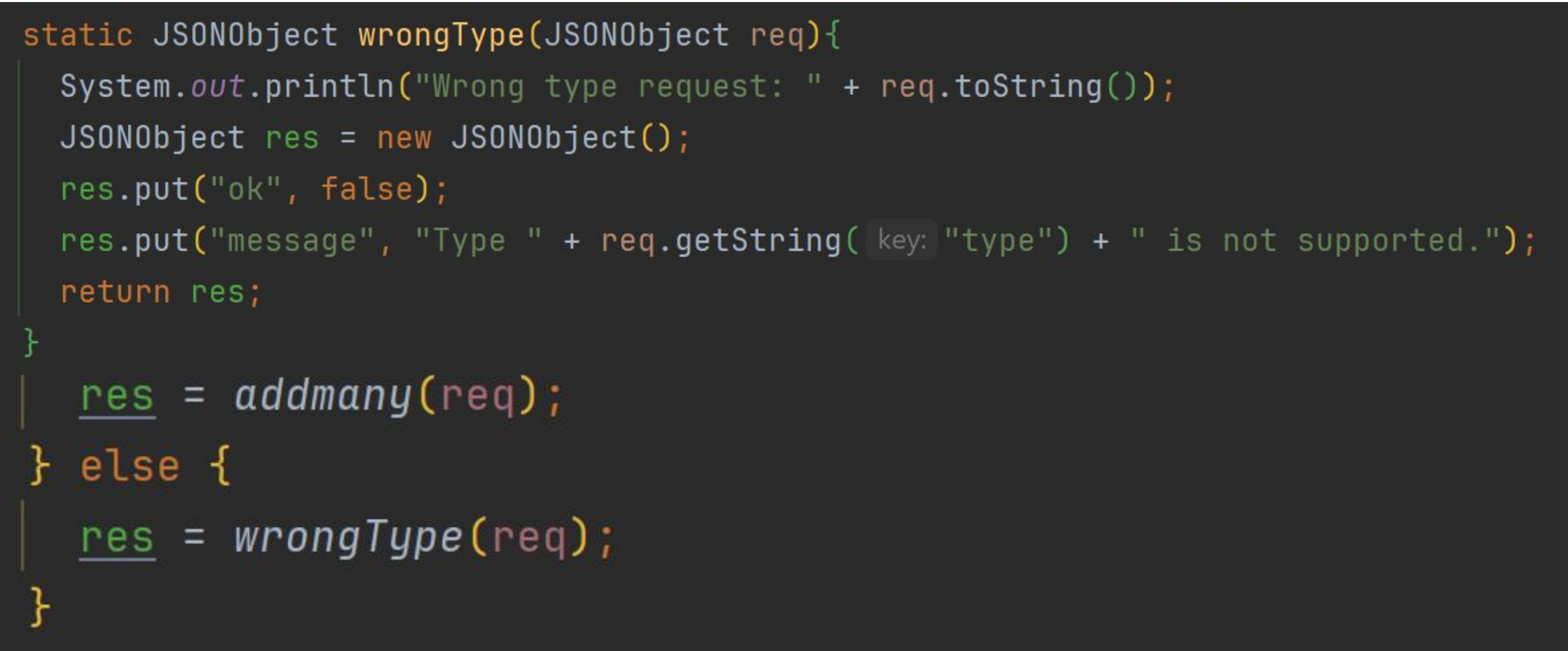
This code would either need to add more handling here in the client

Or the client should send the data regardless of content, and the server handles the type request validity

Assignment 3 Part 1 Starter Code

SER 321

Examples that need more Error Handling



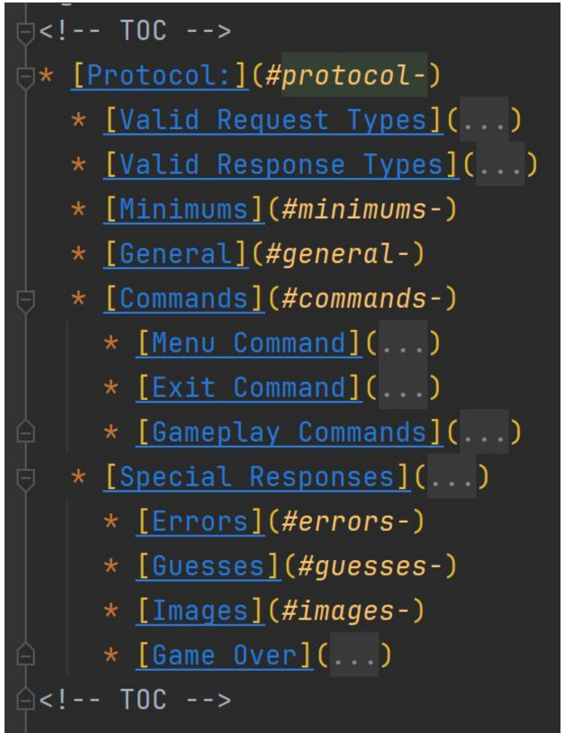
```
static JSONObject wrongType(JSONObject req){
  System.out.println("Wrong type request: " + req.toString());
  JSONObject res = new JSONObject();
  res.put("ok", false);
  res.put("message", "Type " + req.getString( key: "type") + " is not supported.");
  return res;
}
```

```
    res = addmany(req);
} else {
    res = wrongType(req);
}
```

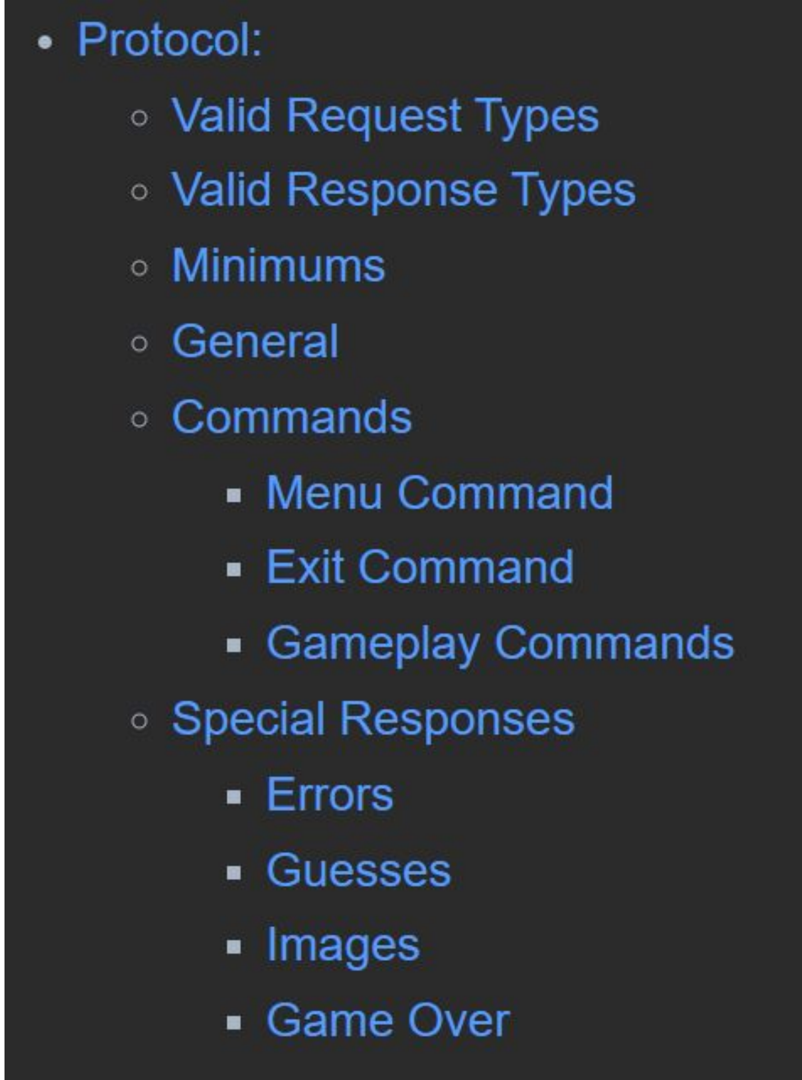
SER 321

Protocol Organization

Markdown Table of Contents is *super* helpful!



```
<!-- TOC -->
* [Protocol:](#protocol-)
  * [Valid Request Types](...)
  * [Valid Response Types](...)
  * [Minimums](#minimums-)
  * [General](#general-)
  * [Commands](#commands-)
    * [Menu Command](...)
    * [Exit Command](...)
    * [Gameplay Commands](...)
  * [Special Responses](...)
    * [Errors](#errors-)
    * [Guesses](#guesses-)
    * [Images](#images-)
    * [Game Over](...)
<!-- TOC -->
```



- Protocol:
  - Valid Request Types
  - Valid Response Types
  - Minimums
  - General
  - Commands
    - Menu Command
    - Exit Command
    - Gameplay Commands
  - Special Responses
    - Errors
    - Guesses
    - Images
    - Game Over

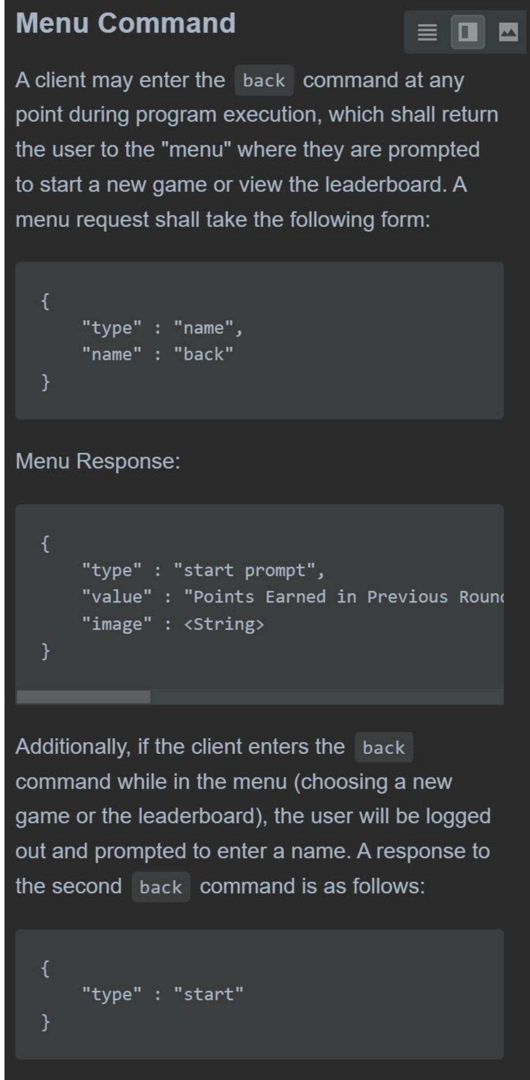
SER 321
Protocol Organization

Make Categories!

Be descriptive!

If you find yourself getting lost in the file, try printing it out!

Print it out and color code it!



Menu Command

A client may enter the `back` command at any point during program execution, which shall return the user to the "menu" where they are prompted to start a new game or view the leaderboard. A menu request shall take the following form:

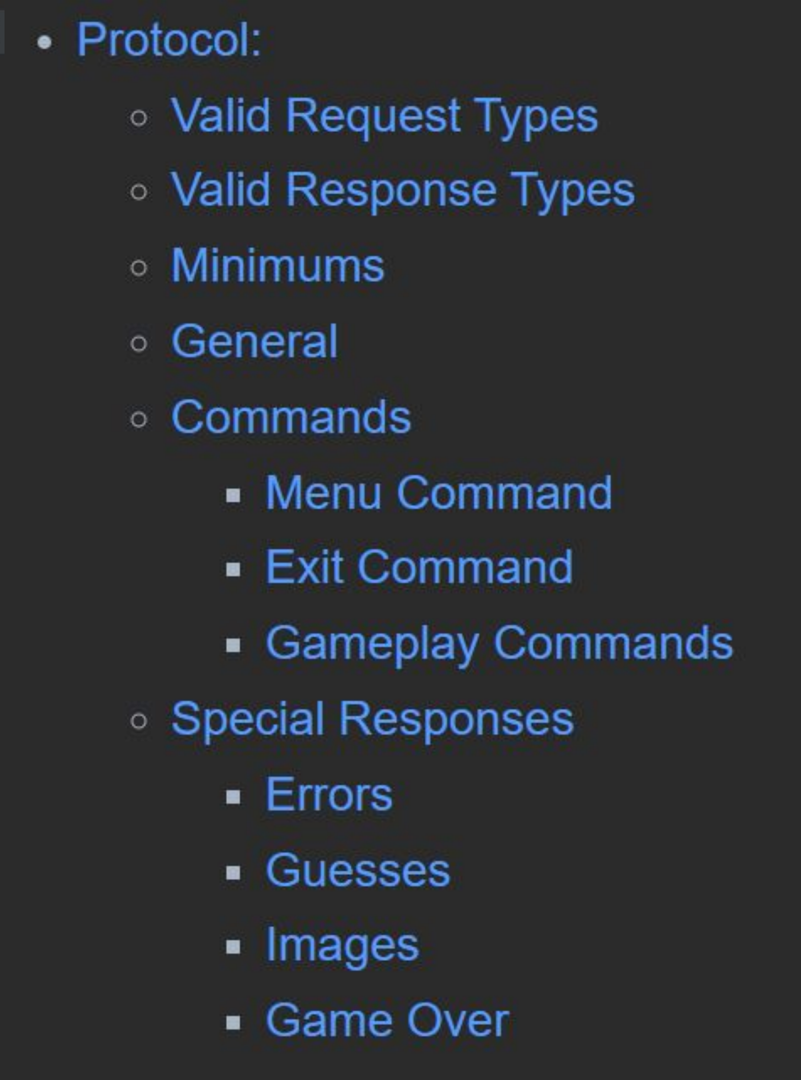
```
{
    "type" : "name",
    "name" : "back"
}
```

Menu Response:

```
{
    "type" : "start prompt",
    "value" : "Points Earned in Previous Roun
    "image" : <String>
}
```

Additionally, if the client enters the `back` command while in the menu (choosing a new game or the leaderboard), the user will be logged out and prompted to enter a name. A response to the second `back` command is as follows:

```
{
    "type" : "start"
}
```



- Protocol:
  - Valid Request Types
  - Valid Response Types
  - Minimums
  - General
  - Commands
    - Menu Command
    - Exit Command
    - Gameplay Commands
  - Special Responses
    - Errors
    - Guesses
    - Images
    - Game Over

Questions?

Survey:

https://bit.ly/asn_survey

Upcoming Events

SI Sessions:

- Thursday, November 2nd 2023 at 7:00 pm MST
- Sunday, November 5th 2023 at 7:00 pm MST
- Monday, November 6th 2023 at 4:00 pm MST

Review Sessions:

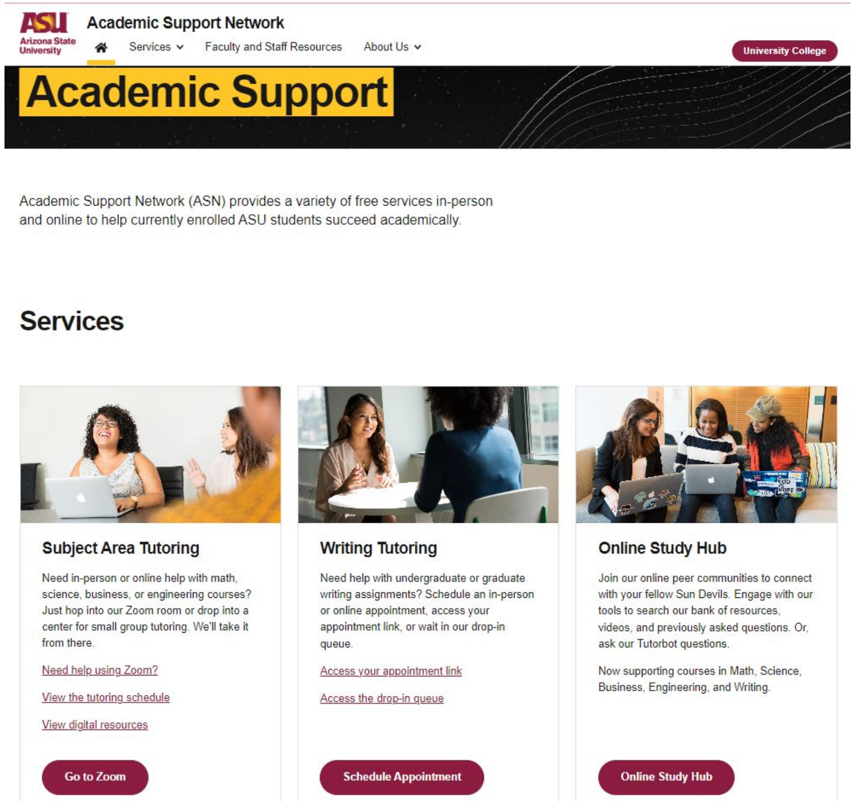
- Must be **48 hours before** the final opens
- Final is typically available the last three days of class (11/29 - 12/1)
- Absolute last day to hold Review Session would be **Monday, November 27th**

I will be posting a poll concerning the Review Session schedule next week!

More Questions?

Check out our other resources!

tutoring.asu.edu



Academic Support Network

Services · Faculty and Staff Resources · About Us · University College

Academic Support

Academic Support Network (ASN) provides a variety of free services in-person and online to help currently enrolled ASU students succeed academically.

Services

Subject Area Tutoring

Need in-person or online help with math, science, business, or engineering courses? Just hop into our Zoom room or drop into a center for small group tutoring. We'll take it from there.

Need help using Zoom?

View the tutoring schedule

View digital resources

Go to Zoom

Writing Tutoring

Need help with undergraduate or graduate writing assignments? Schedule an in-person or online appointment, access your appointment link, or wait in our drop-in queue.

Access your appointment link

Access the drop-in queue

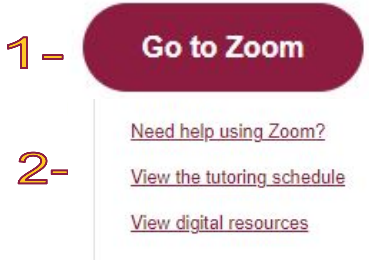
Schedule Appointment

Online Study Hub

Join our online peer communities to connect with your fellow Sun Devils. Engage with our tools to search our bank of resources, videos, and previously asked questions. Or, ask our Tutorbot questions.

Now supporting courses in Math, Science, Business, Engineering, and Writing.

Online Study Hub

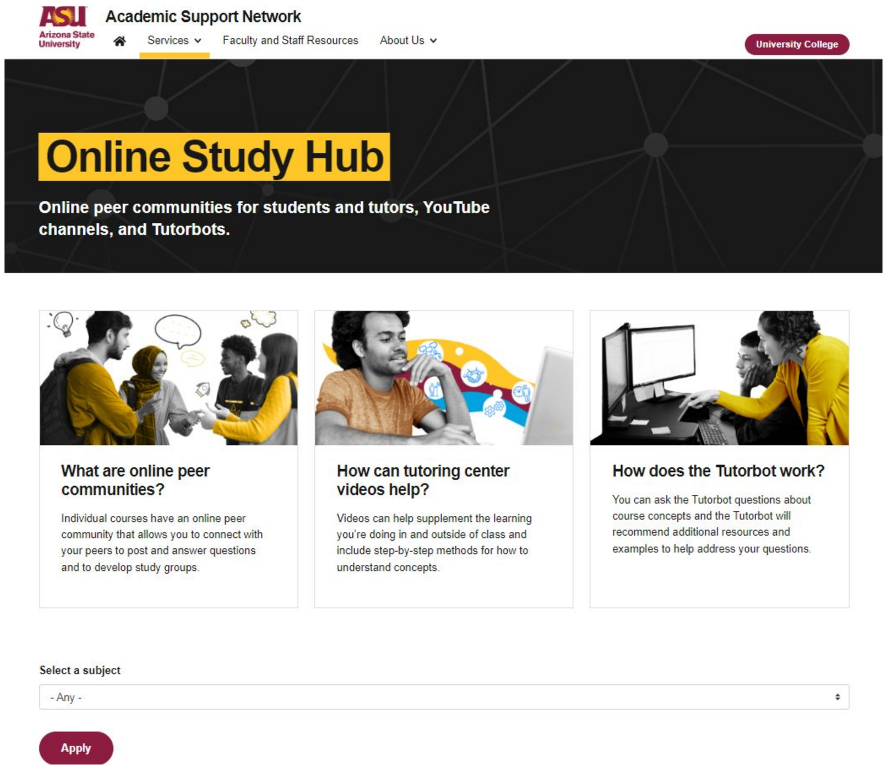


1. Click on 'Go to Zoom' to log onto our Online Tutoring Center.
2. Click on 'View the tutoring schedule' to see when tutors are available for specific courses.

More Questions?

Check out our other resources!

tutoring.asu.edu/online-study-hub



Online Study Hub

Online peer communities for students and tutors, YouTube channels, and Tutorbots.

What are online peer communities?

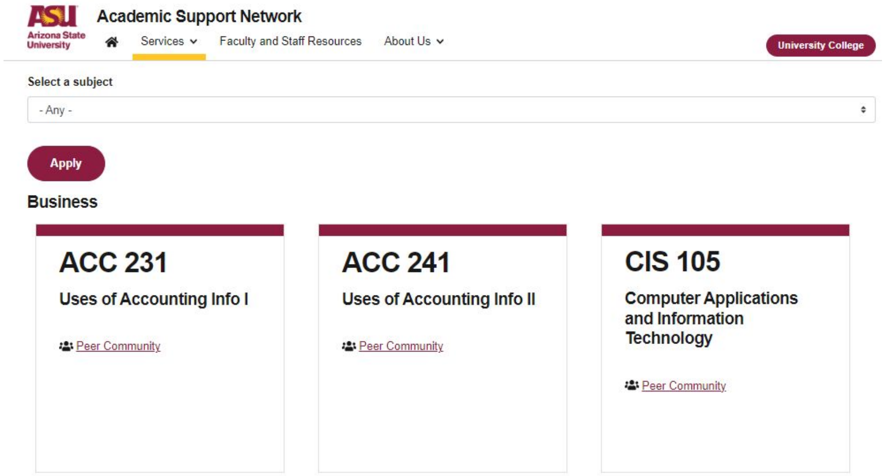
Individual courses have an online peer community that allows you to connect with your peers to post and answer questions and to develop study groups.

How can tutoring center videos help?

Videos can help supplement the learning you're doing in and outside of class and include step-by-step methods for how to understand concepts.

How does the Tutorbot work?

You can ask the Tutorbot questions about course concepts and the Tutorbot will recommend additional resources and examples to help address your questions.



Business

ACC 231
Uses of Accounting Info I
Peer Community

ACC 241
Uses of Accounting Info II
Peer Community

CIS 105
Computer Applications and Information Technology
Peer Community

Don't forget to check out the Online Study Hub for additional resources!

Additional Resources

CoureRepo