SER 321 B Session

SI Session

Thursday October 26th 2023

7:00 - 8:00 pm MST

Agenda

Client Socket

Assignment 3.1 - Protocol

Assignment 3 GUI

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

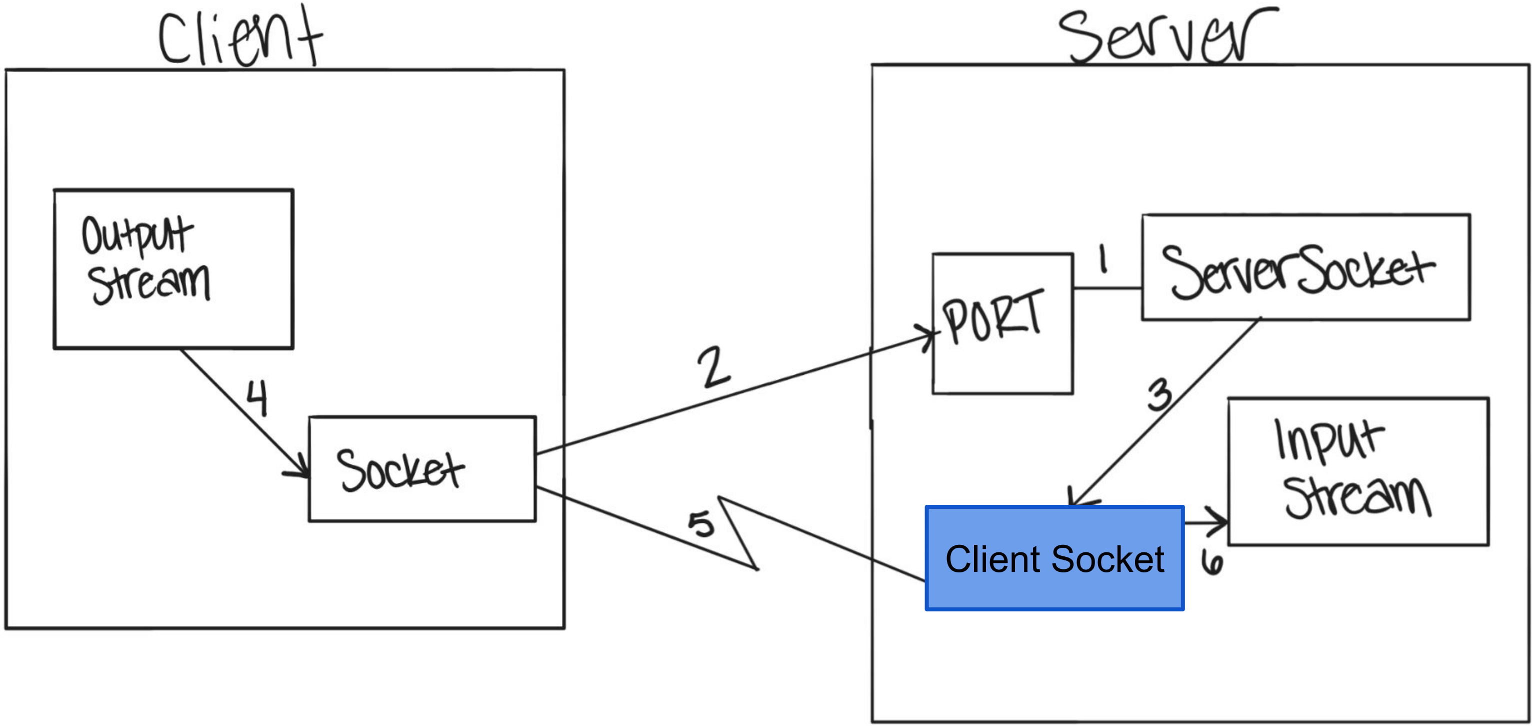
Zoom Chat

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

SER 321

Client Socket

Remember that the OS will dynamically allocate a new port for the Client Socket!

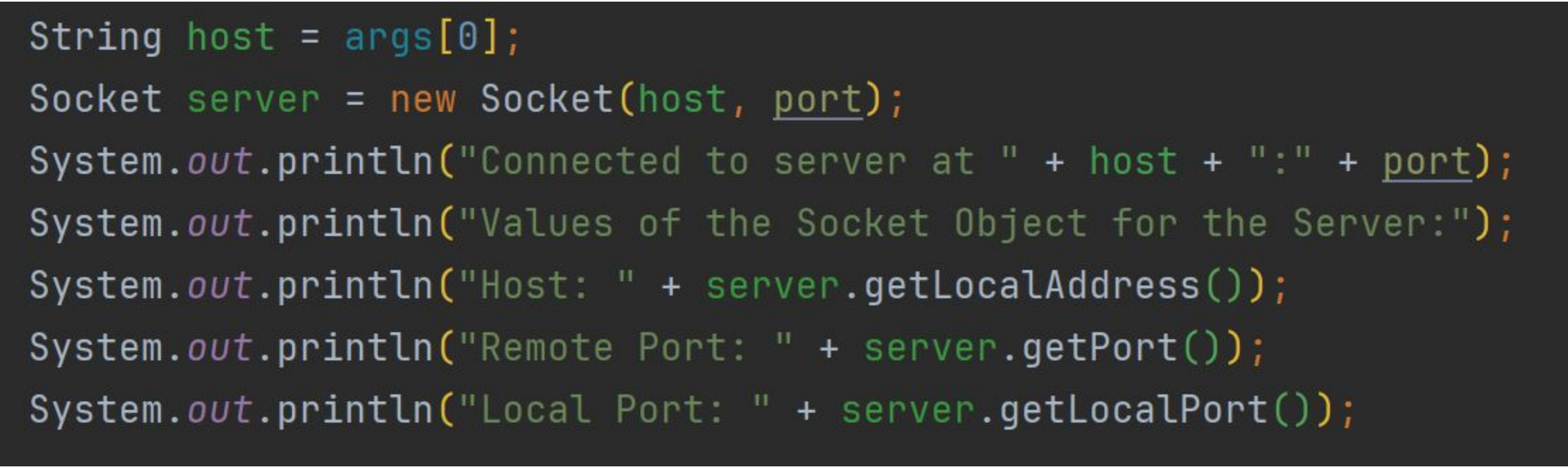


Let’s look at the print statements added to the Echo_Java code

SER 321

Client Socket

Client or Server?



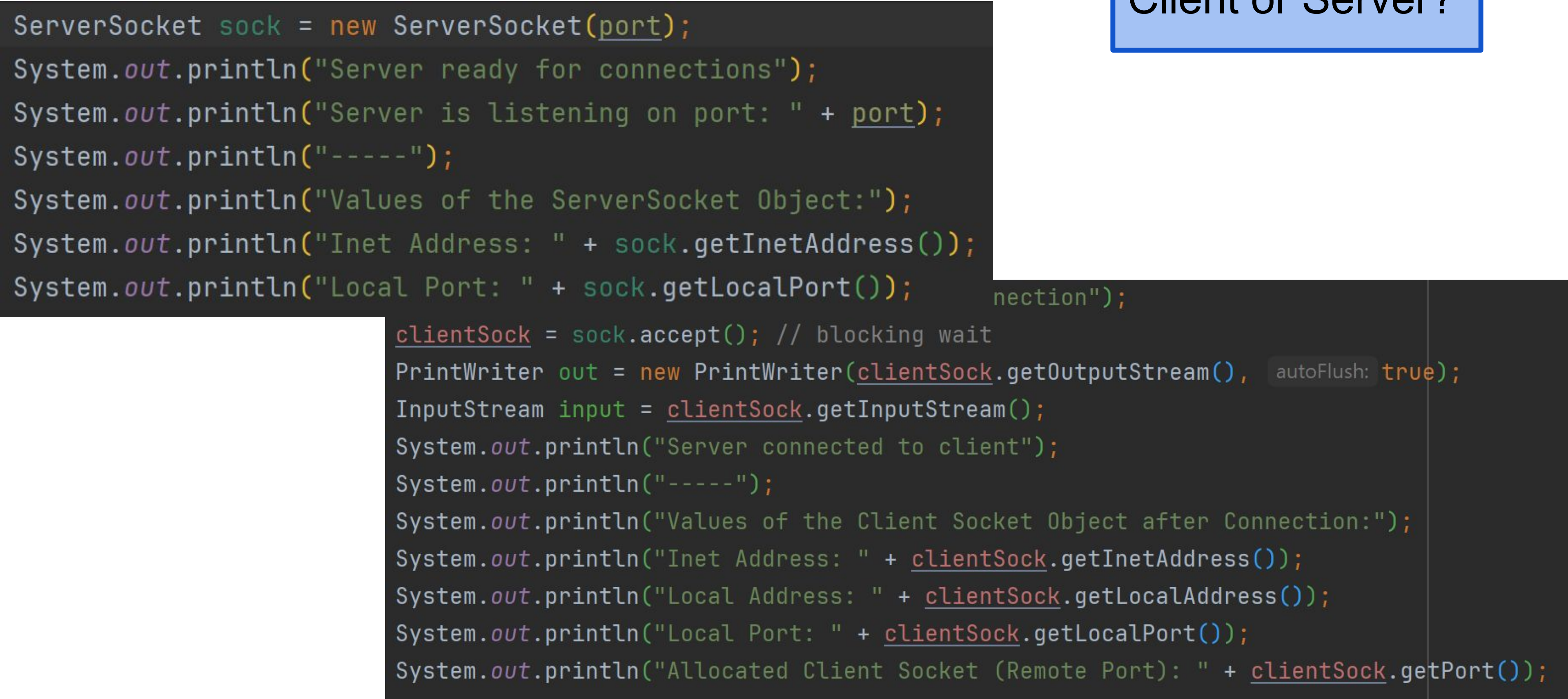
```java
String host = args[0];
Socket server = new Socket(host, port);
System.out.println("Connected to server at " + host + ":" + port);
System.out.println("Values of the Socket Object for the Server:");
System.out.println("Host: " + server.getLocalAddress());
System.out.println("Remote Port: " + server.getPort());
System.out.println("Local Port: " + server.getLocalPort());
```

Let's look at the print statements added to the Echo_Java code

SER 321

Client Socket

Client or Server?



```
ServerSocket sock = new ServerSocket(port);
System.out.println("Server ready for connections");
System.out.println("Server is listening on port: " + port);
System.out.println("-----");
System.out.println("Values of the ServerSocket Object:");
System.out.println("Inet Address: " + sock.getInetAddress());
System.out.println("Local Port: " + sock.getLocalPort());
```

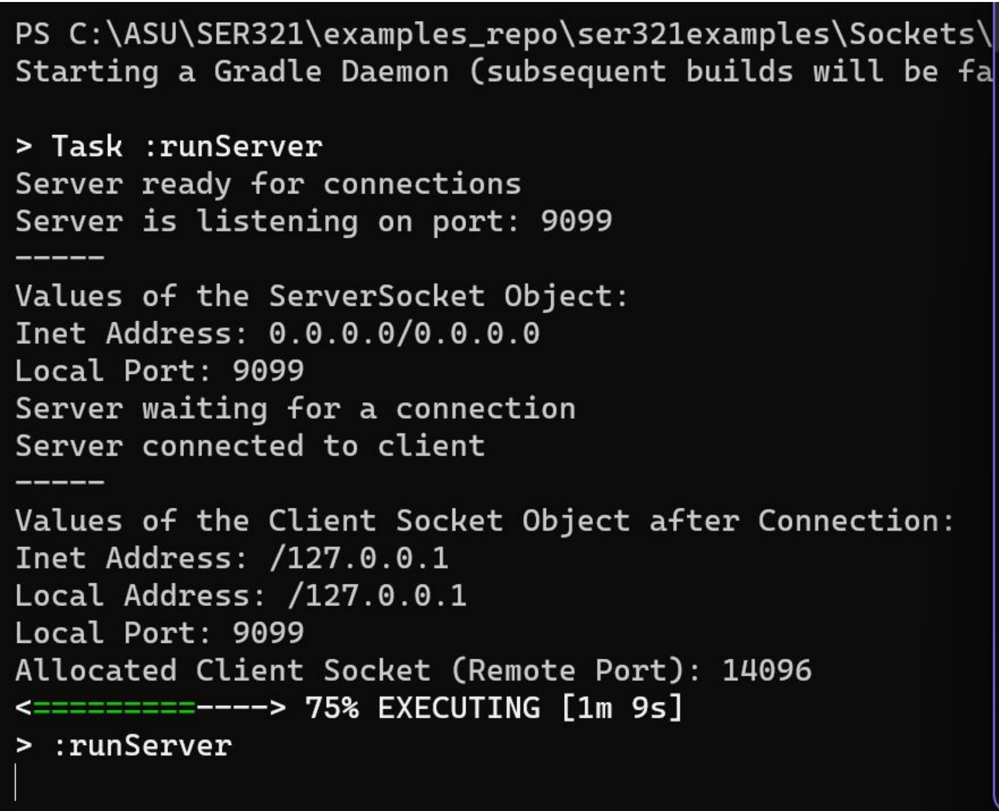
```
nection");
clientSock = sock.accept(); // blocking wait
PrintWriter out = new PrintWriter(clientSock.getOutputStream(), autoFlush: true);
InputStream input = clientSock.getInputStream();
System.out.println("Server connected to client");
System.out.println("-----");
System.out.println("Values of the Client Socket Object after Connection:");
System.out.println("Inet Address: " + clientSock.getInetAddress());
System.out.println("Local Address: " + clientSock.getLocalAddress());
System.out.println("Local Port: " + clientSock.getLocalPort());
System.out.println("Allocated Client Socket (Remote Port): " + clientSock.getPort());
```

Let's look at the print statements added to the Echo_Java code

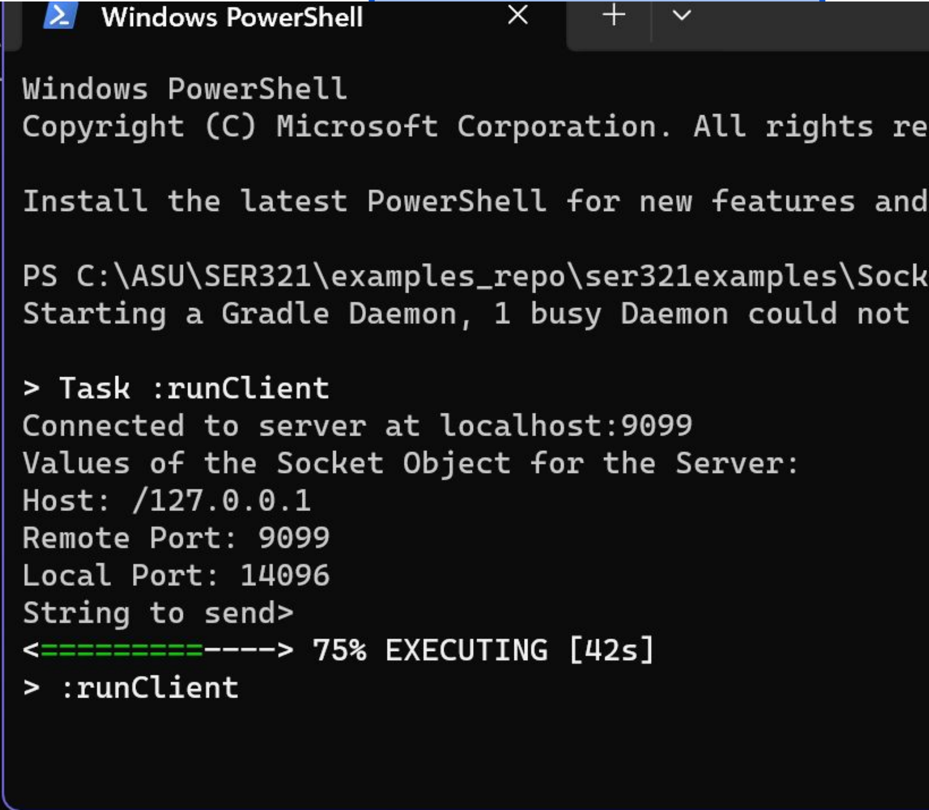
SER 321

Client Socket

Client or Server?

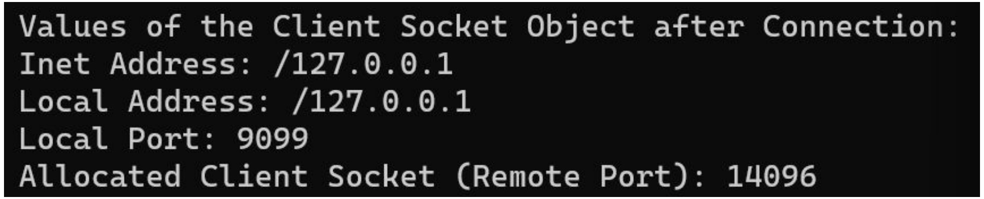


```
PS C:\ASU\SER321\examples_repo\ser321examples\Sockets\
Starting a Gradle Daemon (subsequent builds will be fa

> Task :runServer
Server ready for connections
Server is listening on port: 9099
-----
Values of the ServerSocket Object:
Inet Address: 0.0.0.0/0.0.0.0
Local Port: 9099
Server waiting for a connection
Server connected to client
-----
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): 14096
<=========----> 75% EXECUTING [1m 9s]
> :runServer
```

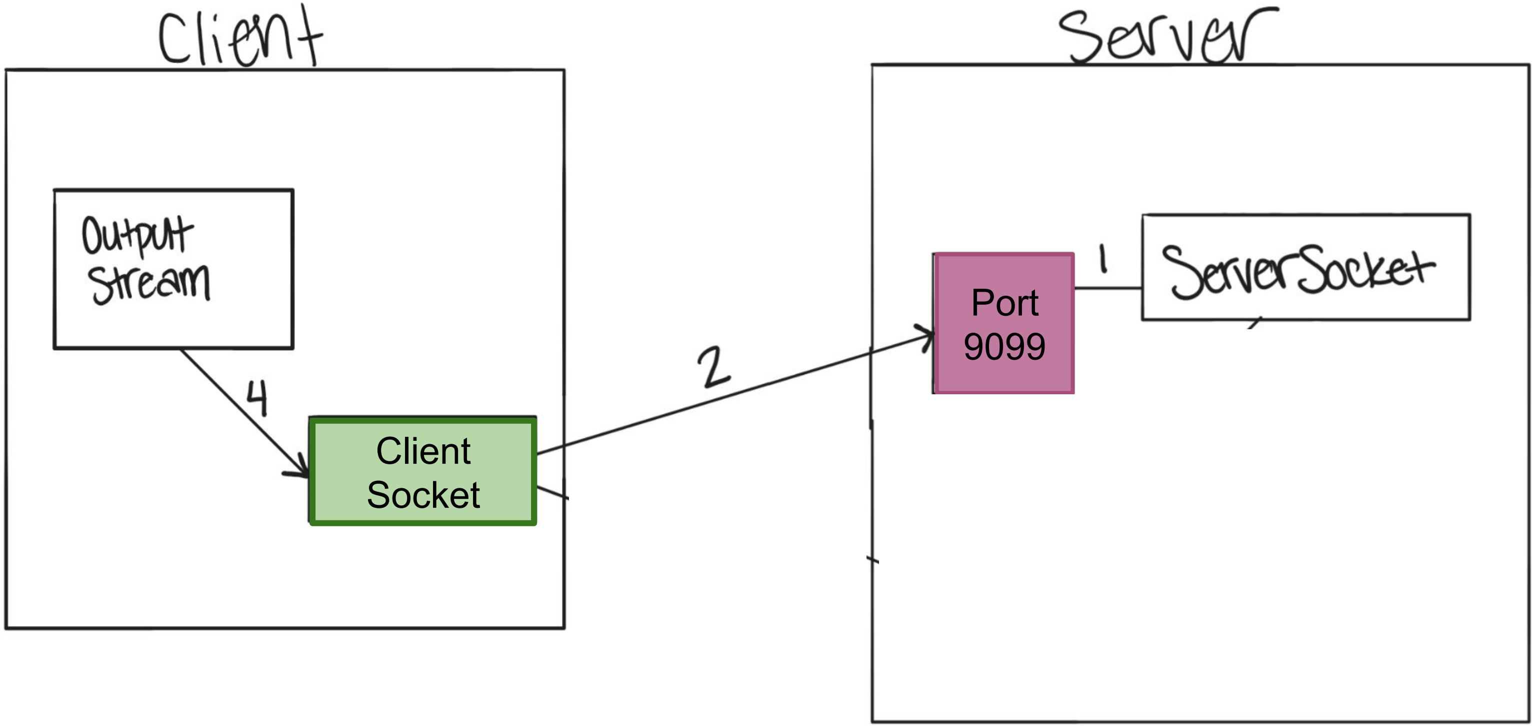


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights re

Install the latest PowerShell for new features and

PS C:\ASU\SER321\examples_repo\ser321examples\Sock
Starting a Gradle Daemon, 1 busy Daemon could not

> 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: 14096
String to send>
<=========----> 75% EXECUTING [42s]
> :runClient
```

SER 321
Client Socket

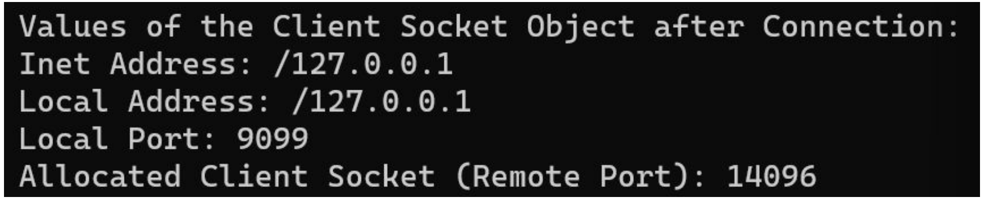


```
Values of the Client Socket Object after Connection:
Inet Address: /127.0.0.1
Local Address: /127.0.0.1
Local Port: 9099
Allocated Client Socket (Remote Port): 14096
```

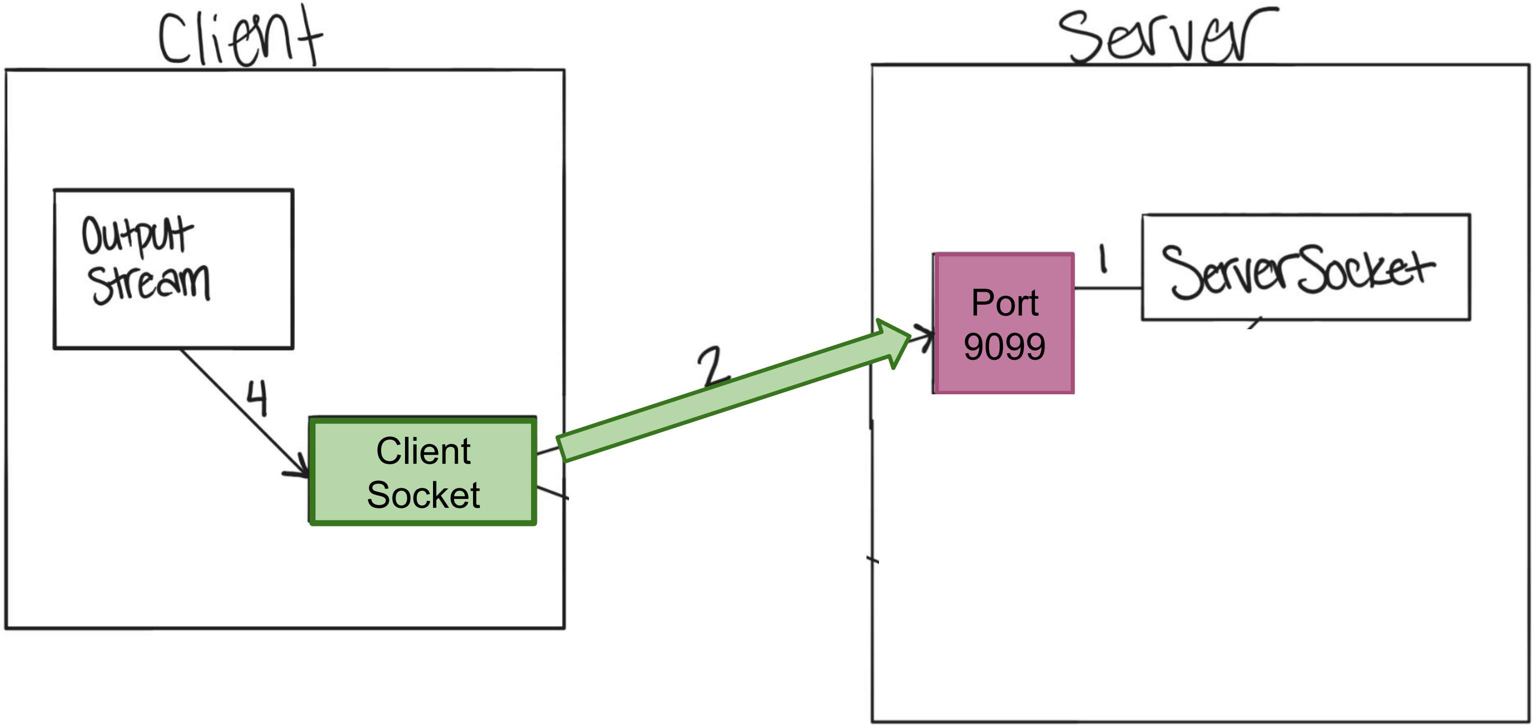


SER 321

Client Socket

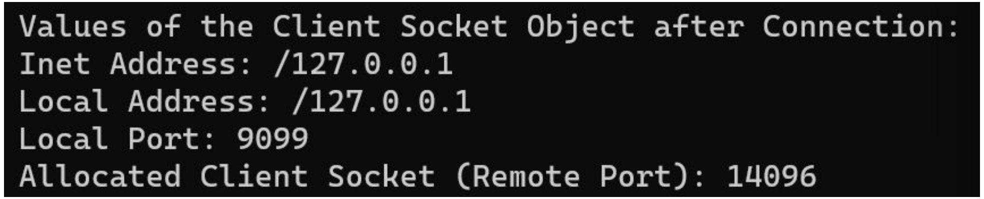


```
Values of the Client Socket Object after Connection:
Inet Address: /127.0.0.1
Local Address: /127.0.0.1
Local Port: 9099
Allocated Client Socket (Remote Port): 14096
```

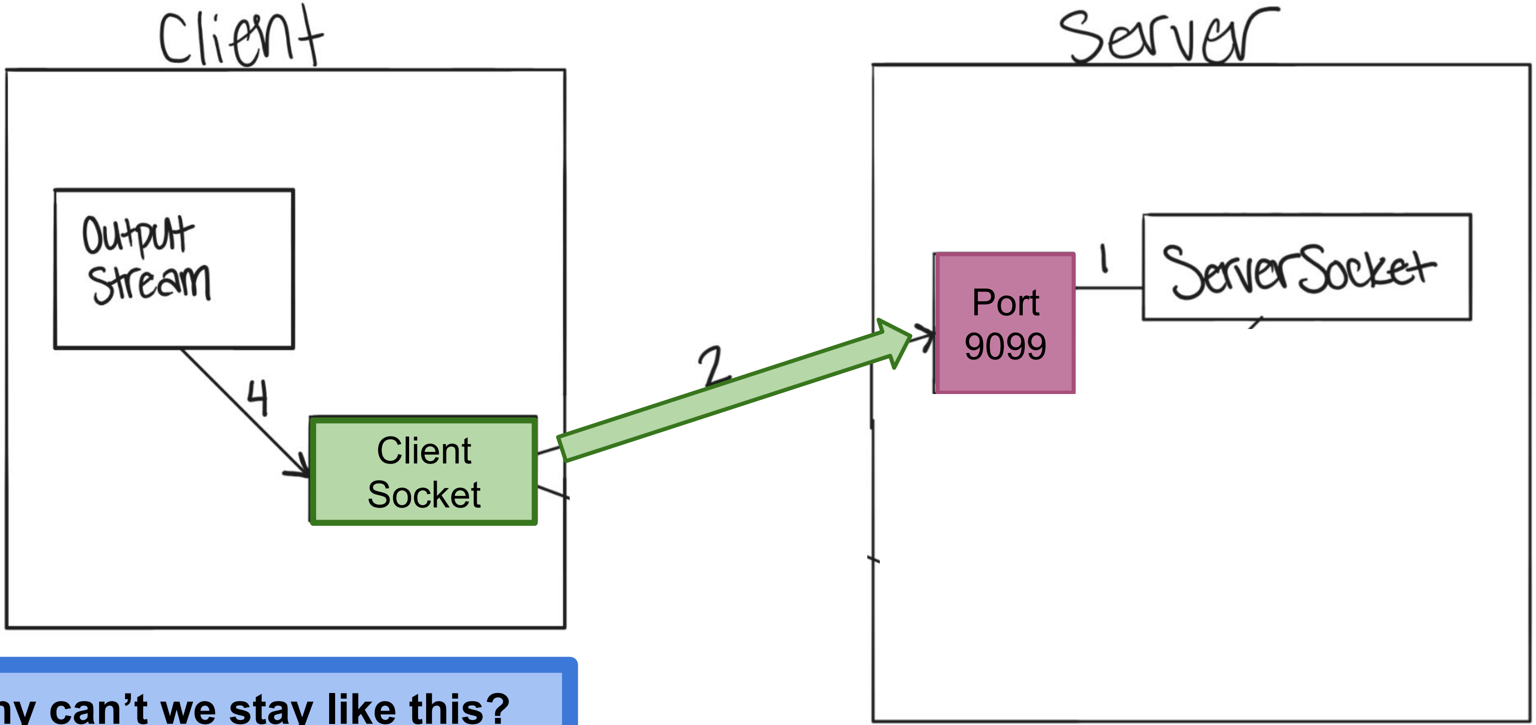


SER 321

Client Socket



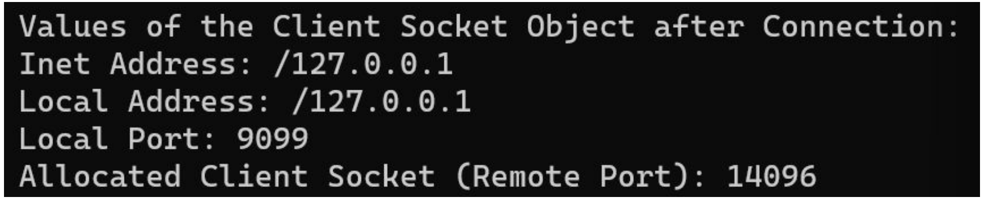
```
Values of the Client Socket Object after Connection:
Inet Address: /127.0.0.1
Local Address: /127.0.0.1
Local Port: 9099
Allocated Client Socket (Remote Port): 14096
```



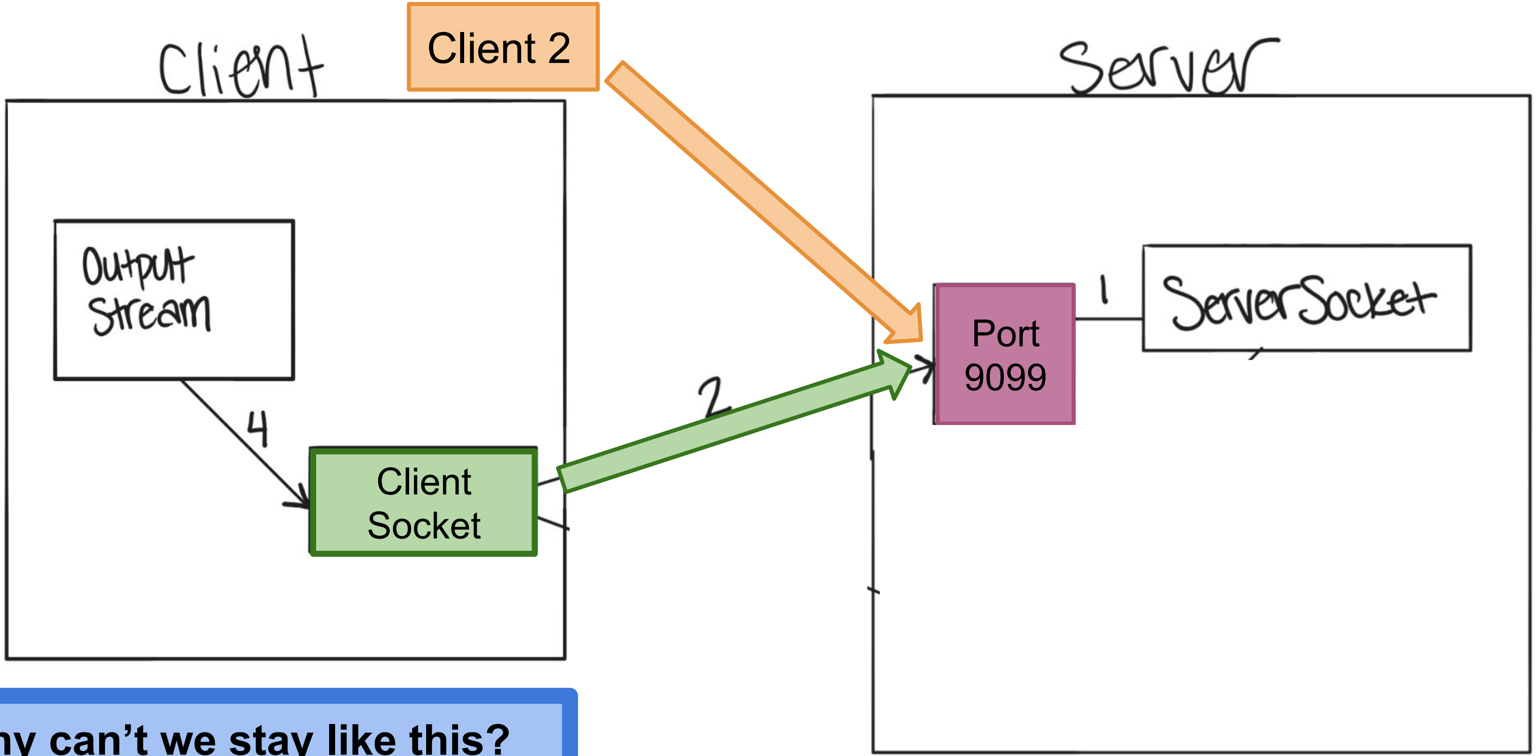
Why can't we stay like this?

SER 321

Client Socket



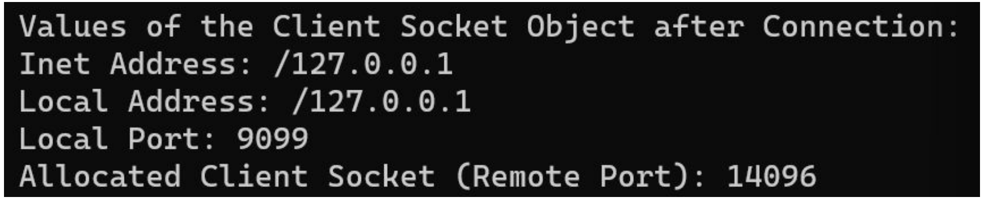
```
Values of the Client Socket Object after Connection:
Inet Address: /127.0.0.1
Local Address: /127.0.0.1
Local Port: 9099
Allocated Client Socket (Remote Port): 14096
```



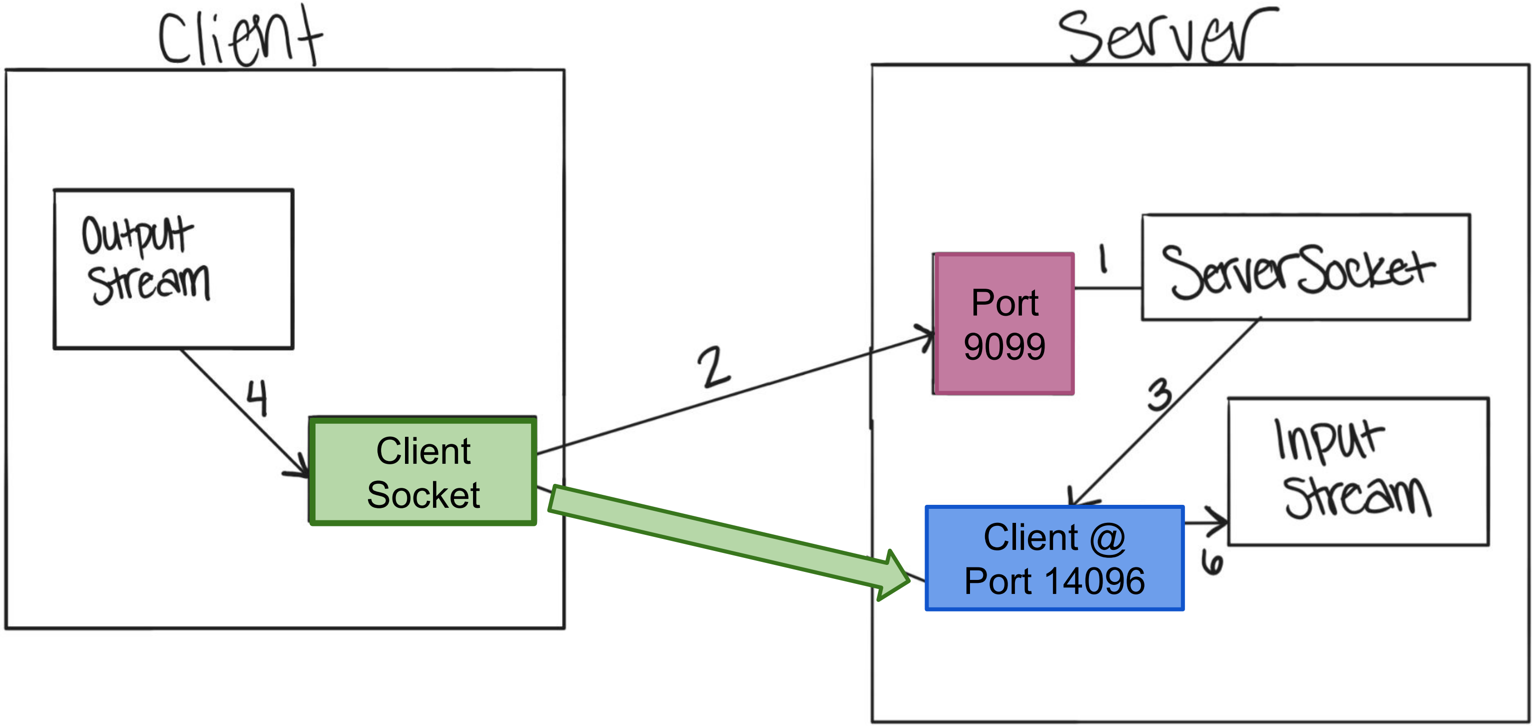
Why can’t we stay like this?

SER 321

Client Socket

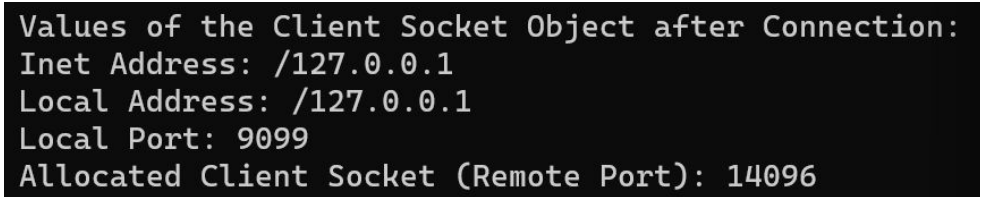


```
Values of the Client Socket Object after Connection:
Inet Address: /127.0.0.1
Local Address: /127.0.0.1
Local Port: 9099
Allocated Client Socket (Remote Port): 14096
```

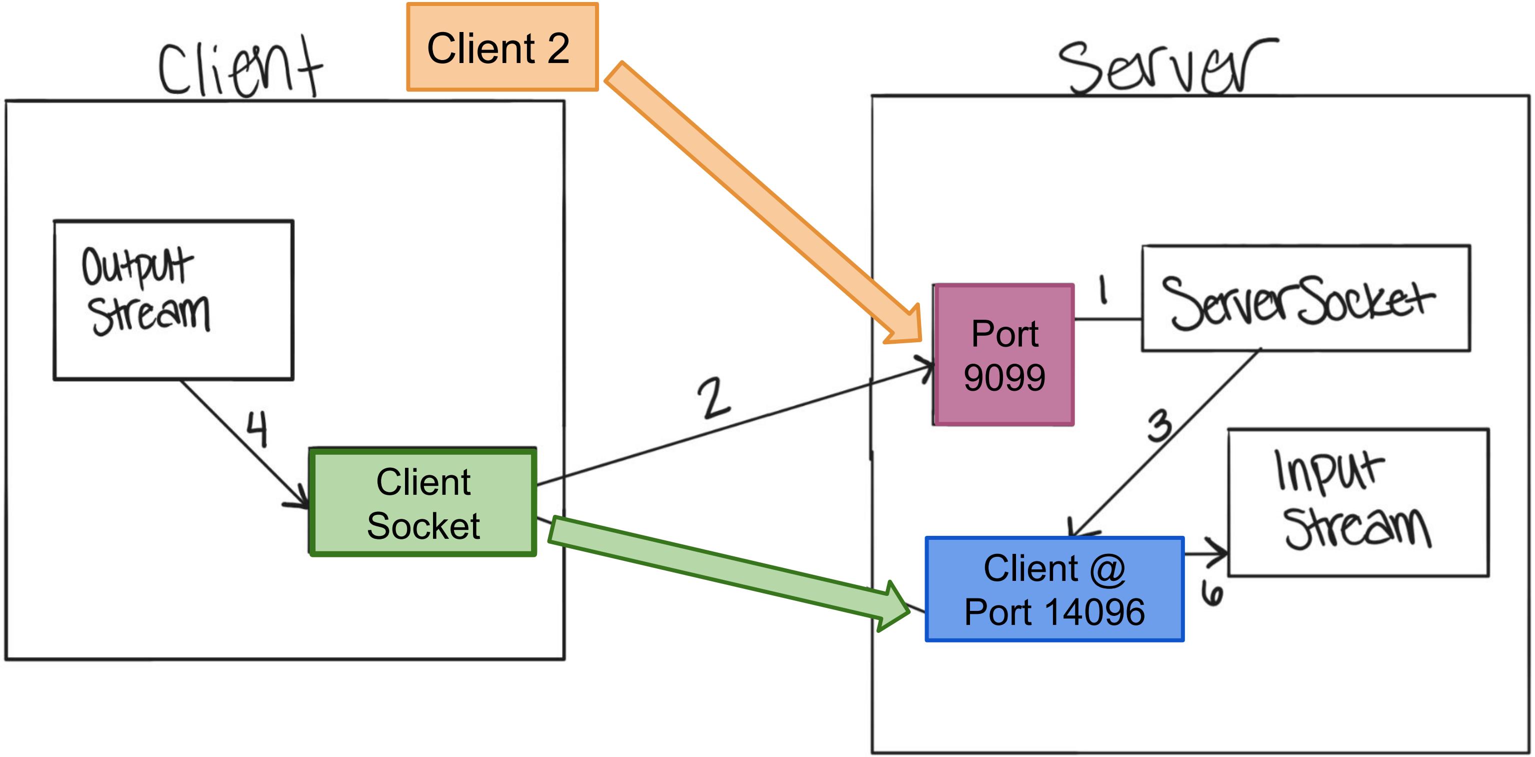


SER 321

Client Socket



```
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): 14096
```



SER 321

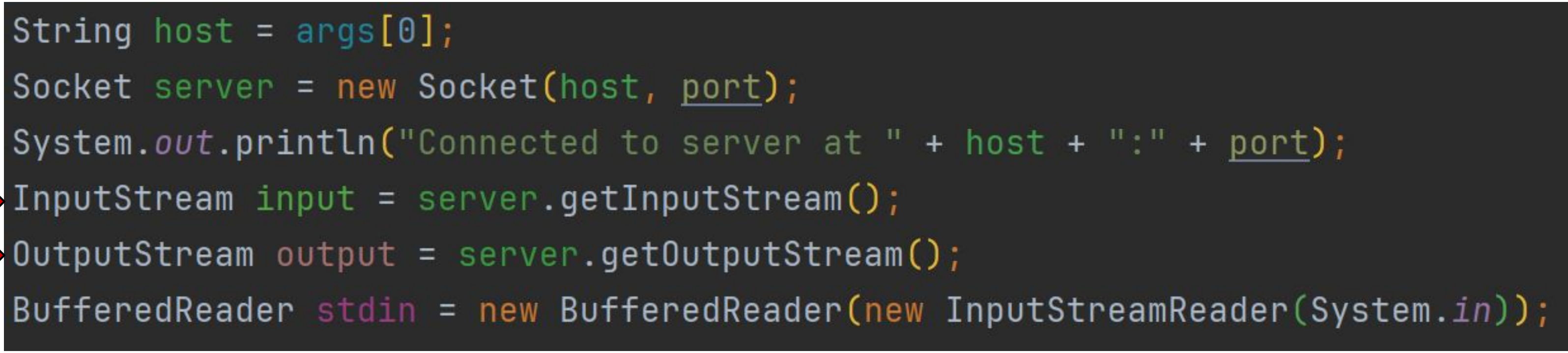
Working with Sockets

So the connection is established, now what?

Need to get the Input and Output streams!

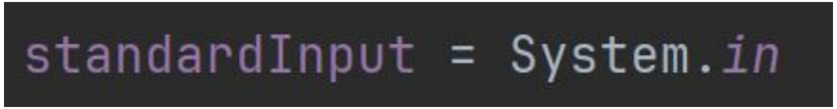
Read **from** server

Write **to** server



```java
String host = args[0];
Socket server = new Socket(host, port);
System.out.println("Connected to server at " + host + ":" + port);
InputStream input = server.getInputStream();
OutputStream output = server.getOutputStream();
BufferedReader stdin = new BufferedReader(new InputStreamReader(System.in));
```

Read from console



```java
standardInput = System.in
```

Need this in your gradle task if you want to read from the console!

SER 321

Assignment 3 Part 1

Play around with the code first! Want to be familiar with it before you start!

Make sure your server is **robust**!

Does not crash!

Handles bad input

Handles client disconnect

Do we have questions on how to go about making the server robust?

Keep the protocol handy at all times!

If you don’t follow the protocol, the system won’t know what you are talking about!

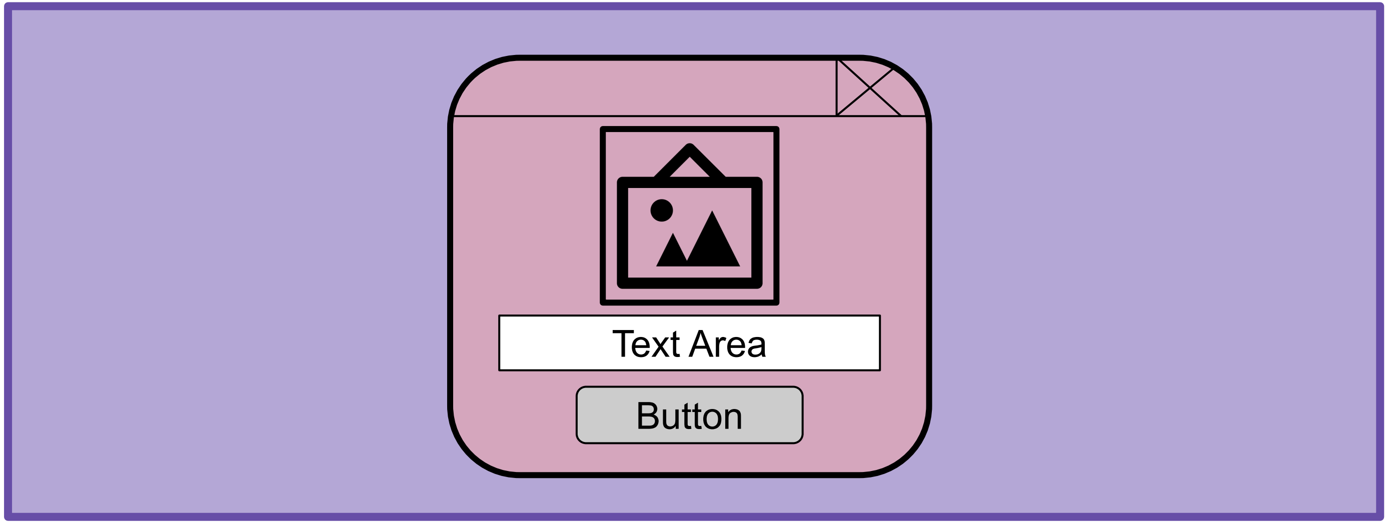
SER 321

Assignment 3 Part 2

Let’s review swing a bit before we dive in…

Panel

Panels hold the content that is displayed! You will put objects like buttons, pictures, and other panels, inside an external panel.



SER 321

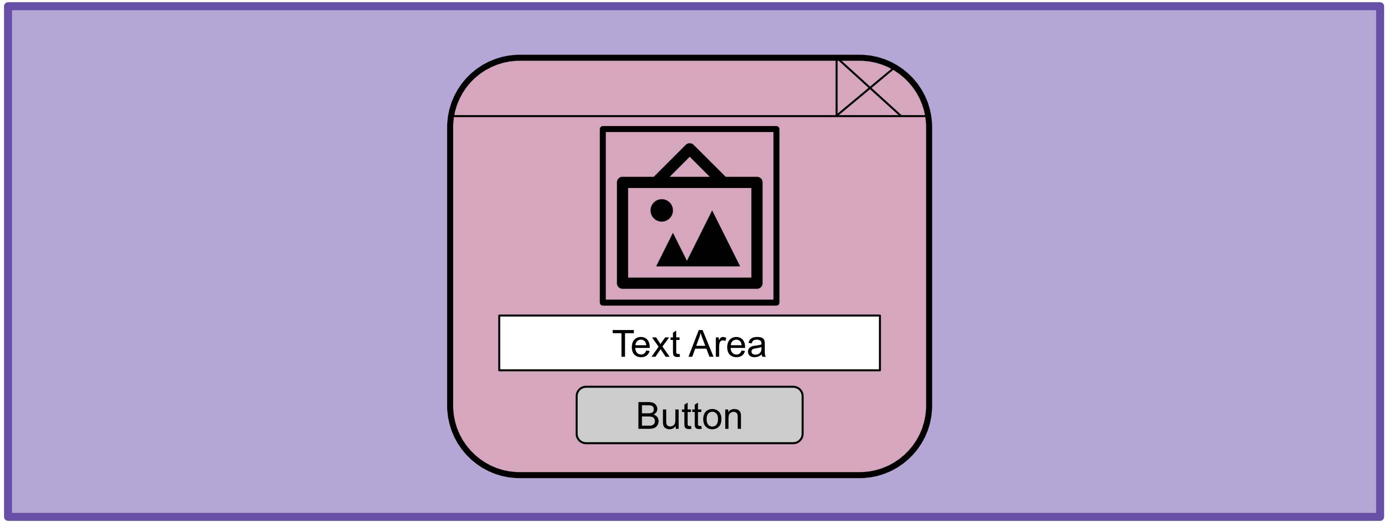
Assignment 3 Part 2

Let’s review swing a bit before we dive in…

Objects

Buttons, Text, Labels, Pictures

For Images we will use an *ImageIcon*. The others prepend a *J* to the object name - *JButton, JTextArea, JLabel, etc.*



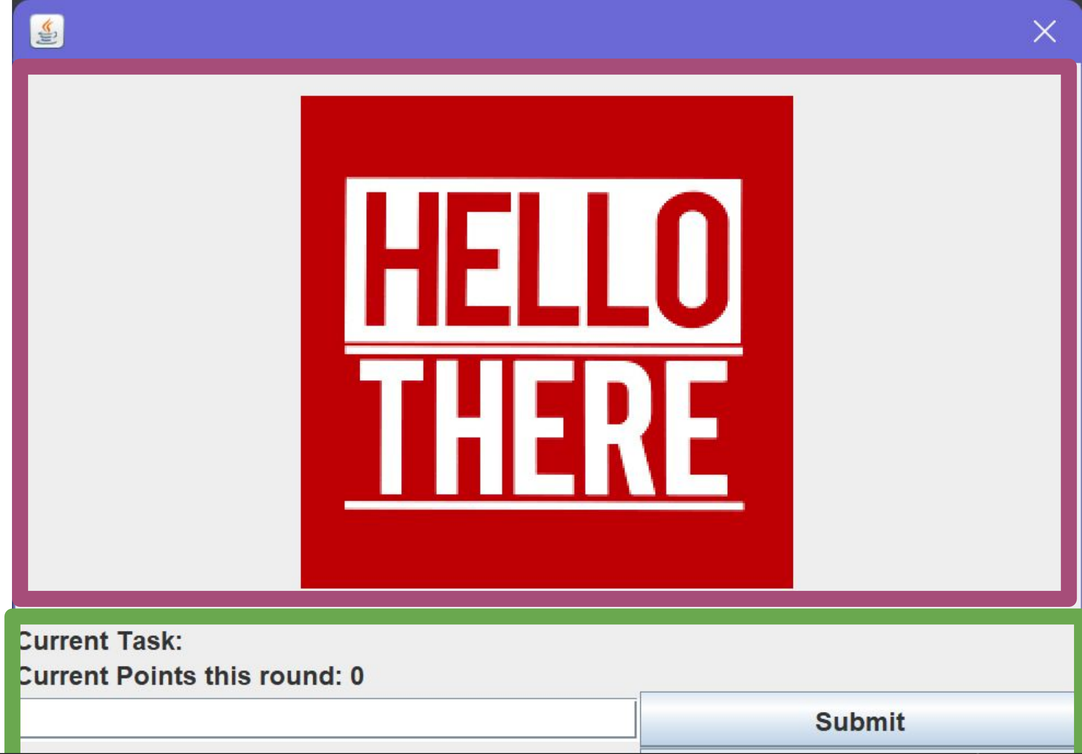
SER 321

Assignment 3 Part 2

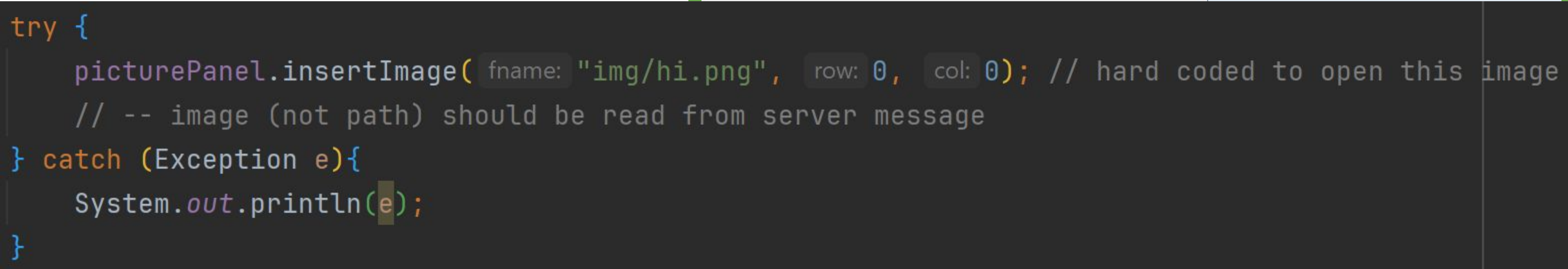
Let’s take a quick peek at the UI in Assignment 3 Part 2

PicturePanel

OutputPanel



Current Task:
Current Points this round: 0



```
try {
    picturePanel.insertImage( fname: "img/hi.png",  row: 0,  col: 0); // hard coded to open this image
    // -- image (not path) should be read from server message
} catch (Exception e){
    System.out.println(e);
}
```

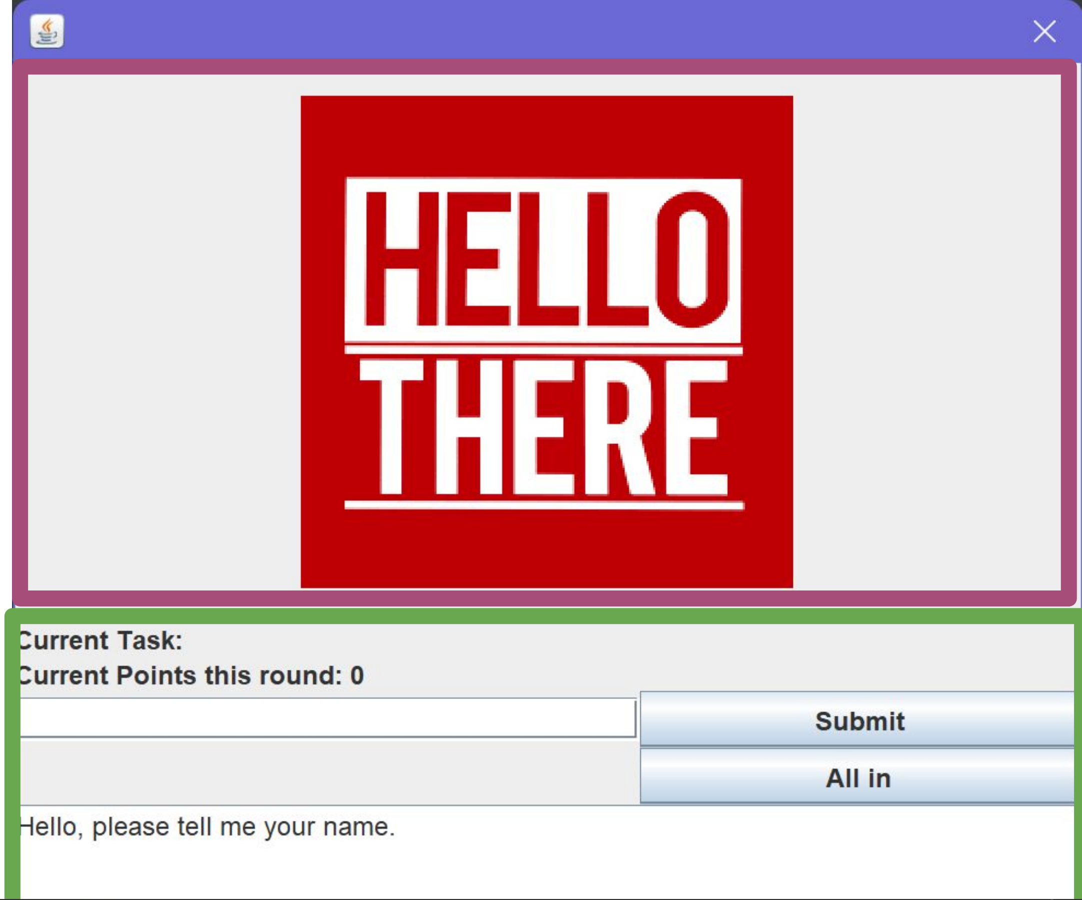
SER 321

Assignment 3 Part 2

Let’s take a quick peek at the UI in Assignment 3 Part 2

PicturePanel

OutputPanel



Current Task:

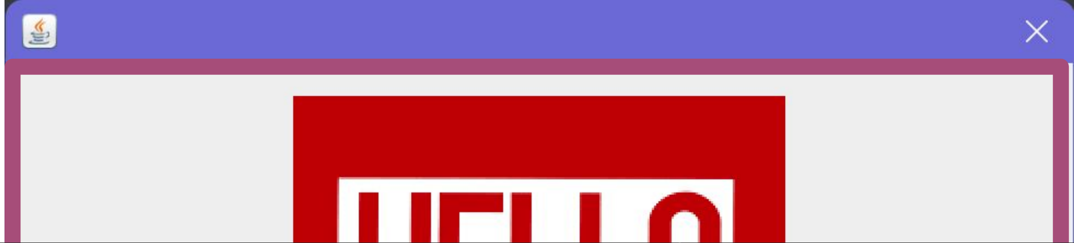
Current Points this round: 0

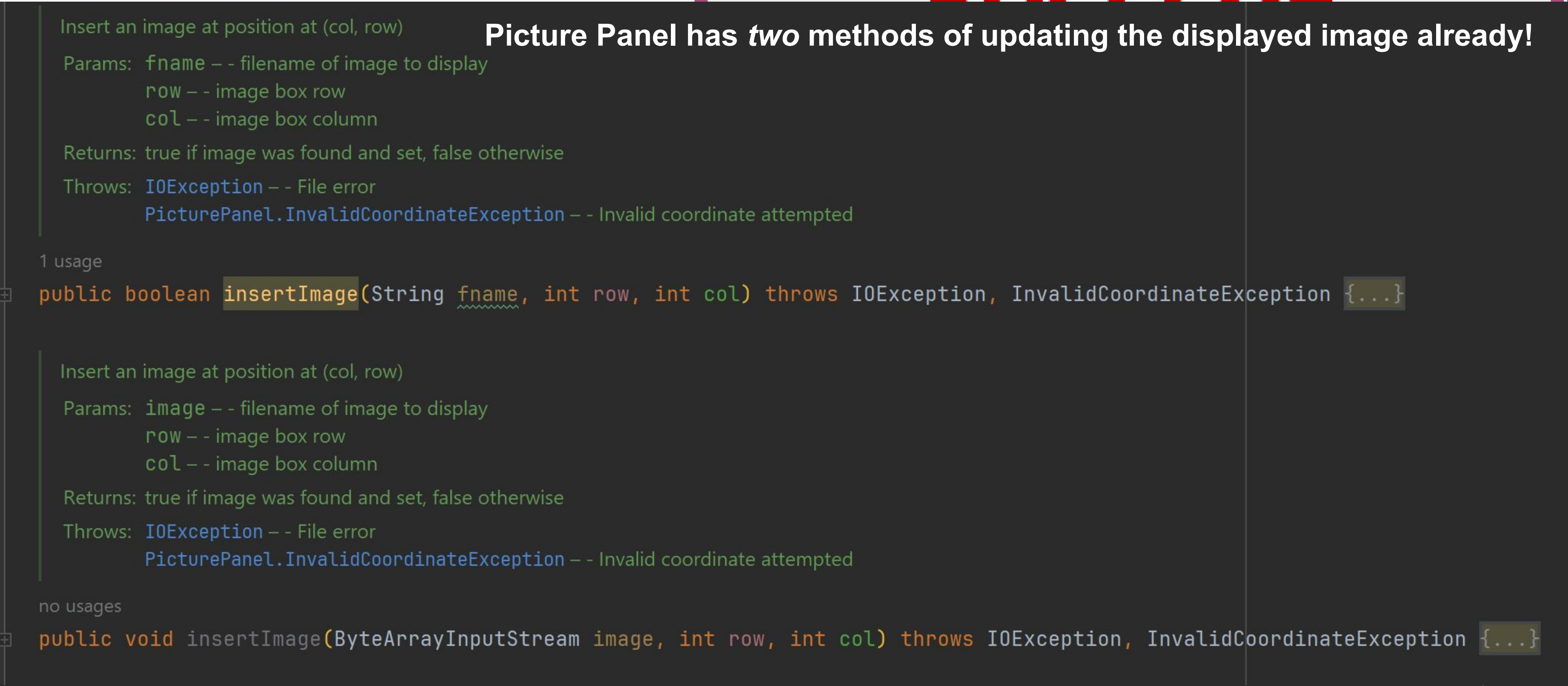
Hello, please tell me your name.

```
String string = this.bufferedReader.readLine(); // wait for answer
JSONObject json = new JSONObject(string); // assumes answer is a JSON
outputPanel.appendOutput(json.getString( key: "value")); // write output value to output panel
```

SER 321

Assignment 3 Part 2





Picture Panel has *two* methods of updating the displayed image already!

```
Insert an image at position at (col, row)
Params:  fname – - filename of image to display
         row – - image box row
         col – - image box column
Returns: true if image was found and set, false otherwise
Throws:  IOException – - File error
         PicturePanel.InvalidCoordinateException – - Invalid coordinate attempted

1 usage
public boolean insertImage(String fname, int row, int col) throws IOException, InvalidCoordinateException {...}

Insert an image at position at (col, row)
Params:  image – - filename of image to display
         row – - image box row
         col – - image box column
Returns: true if image was found and set, false otherwise
Throws:  IOException – - File error
         PicturePanel.InvalidCoordinateException – - Invalid coordinate attempted

no usages
public void insertImage(ByteArrayInputStream image, int row, int col) throws IOException, InvalidCoordinateException {...}
```

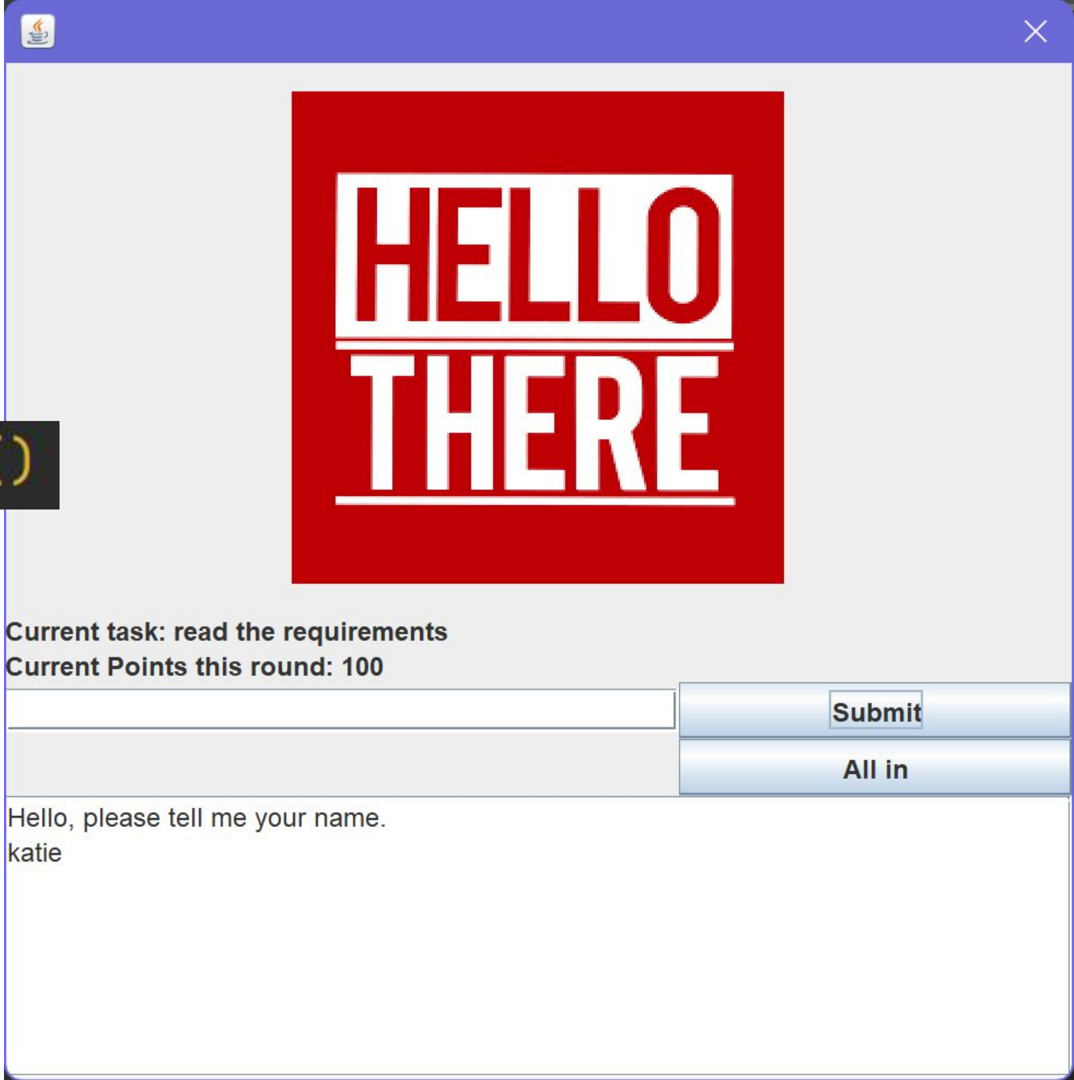
SER 321

Assignment 3 Part 2

Let's take a quick peek at the UI in Assignment 3 Part 2

```
public void submitClicked()
```

Let's take a closer look together!



Current task: read the requirements

Current Points this round: 100

Hello, please tell me your name.
katie

Questions?

Survey:

https://bit.ly/asn_survey

Upcoming Events

SI Sessions:

- Sunday, October 29th 2023 at 7:00 pm MST
- Monday, October 30th 2023 at 4:00 pm MST
- Thursday, November 2nd, 2023 at 7:00 pm MST

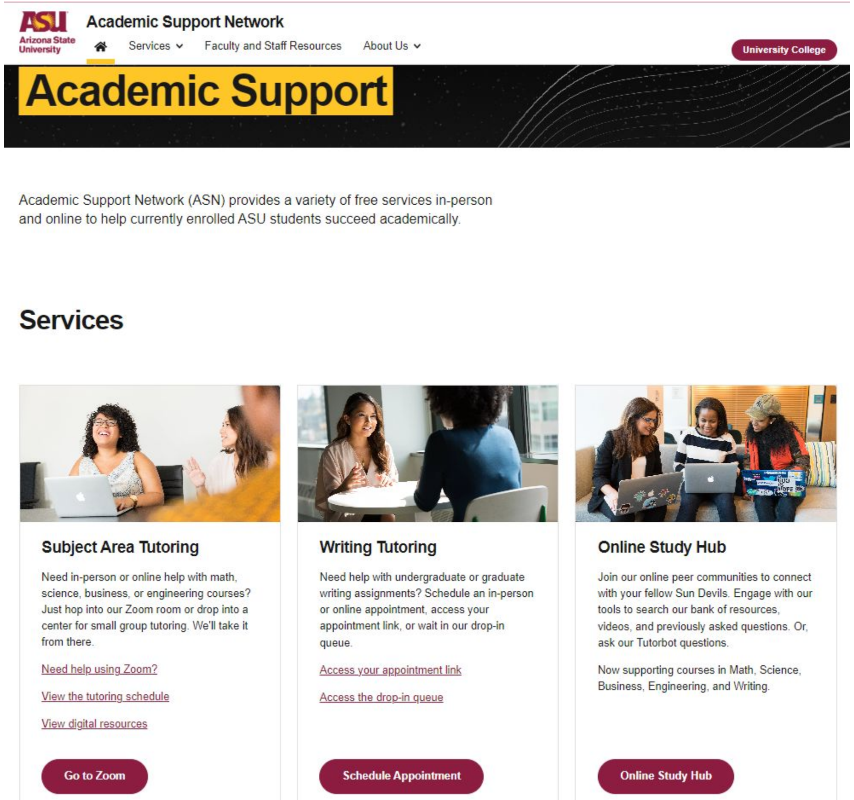
Review Sessions:

- TBD

More Questions?

Check out our other resources!

tutoring.asu.edu



ASU Arizona State University

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

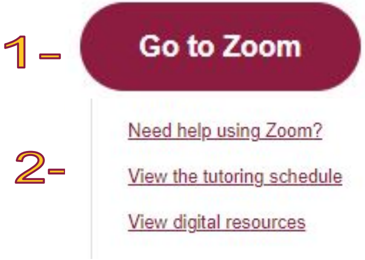
Schedule Appointment

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 –

Go to Zoom

2–

Need help using Zoom?

View the tutoring schedule

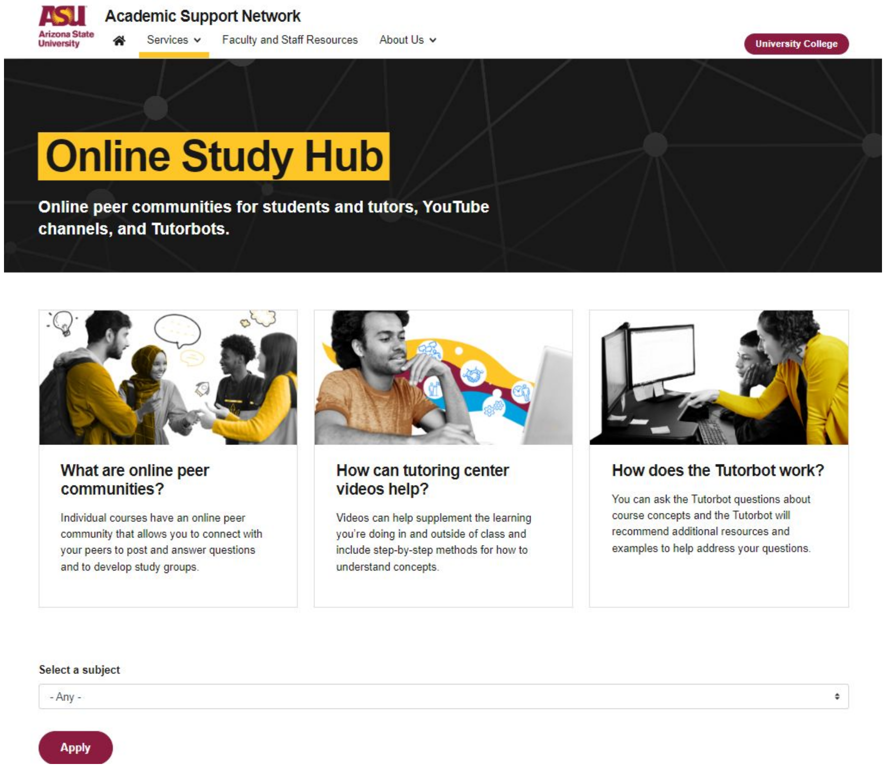
View digital resources

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.

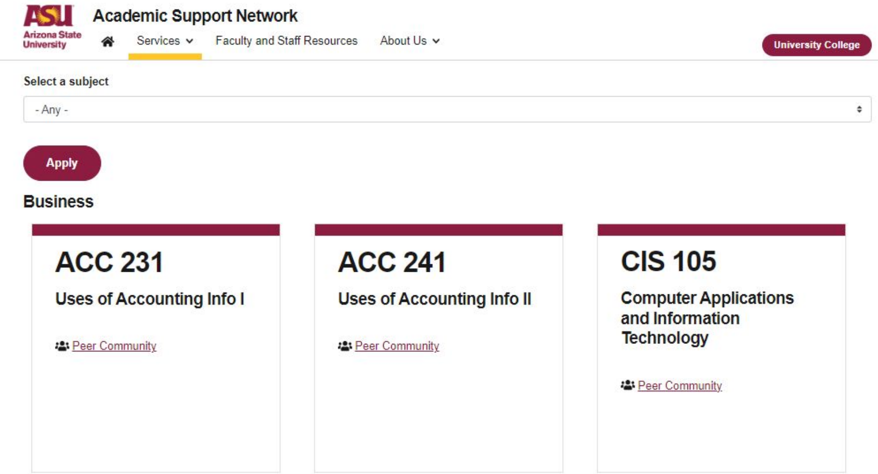
How can tutoring center videos help?

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.

Select a subject: - Any -



Select a subject: - Any -

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