SER 321 B Session

SI Session

Monday October 23rd 2023

4:00 - 5:00 pm MST

Agenda

- Running your Server
- Running the “Real” Server
- Sockets
  - Steps for use
  - The Client Socket
- Assignment 3 GUI Intro

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

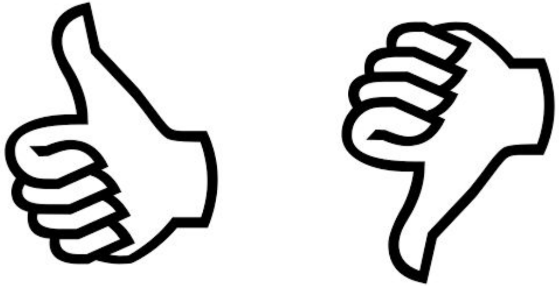
Zoom Chat

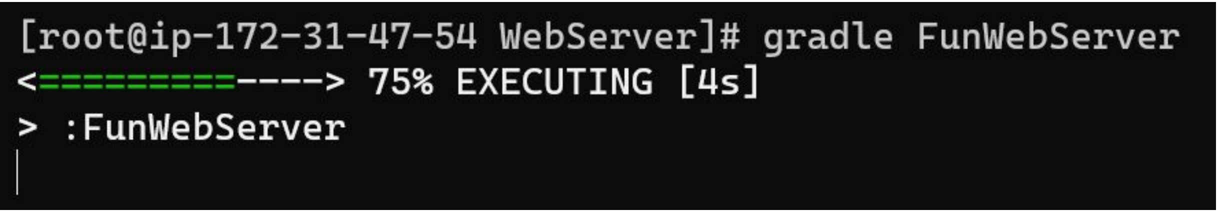
- Use the chat feature to interact with the presenter and respond to presenter's questions.
- Annotations are encouraged

SER 321

Assignment 2 WebServer

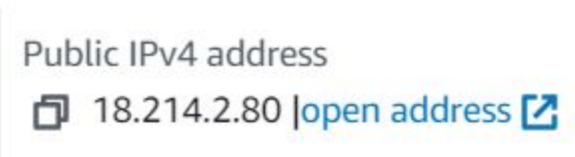
Were we able to get the WebServer in **Task 2** running?



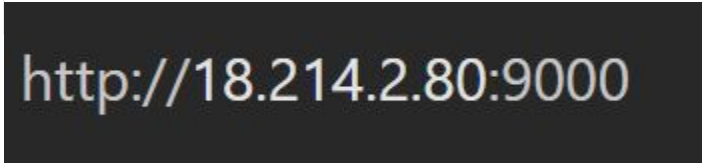


```
[root@ip-172-31-47-54 WebServer]# gradle FunWebServer
<=========----> 75% EXECUTING [4s]
> :FunWebServer
```

STEP ONE:
`gradle FunWebServer`



Public IPv4 address
18.214.2.80 |open address



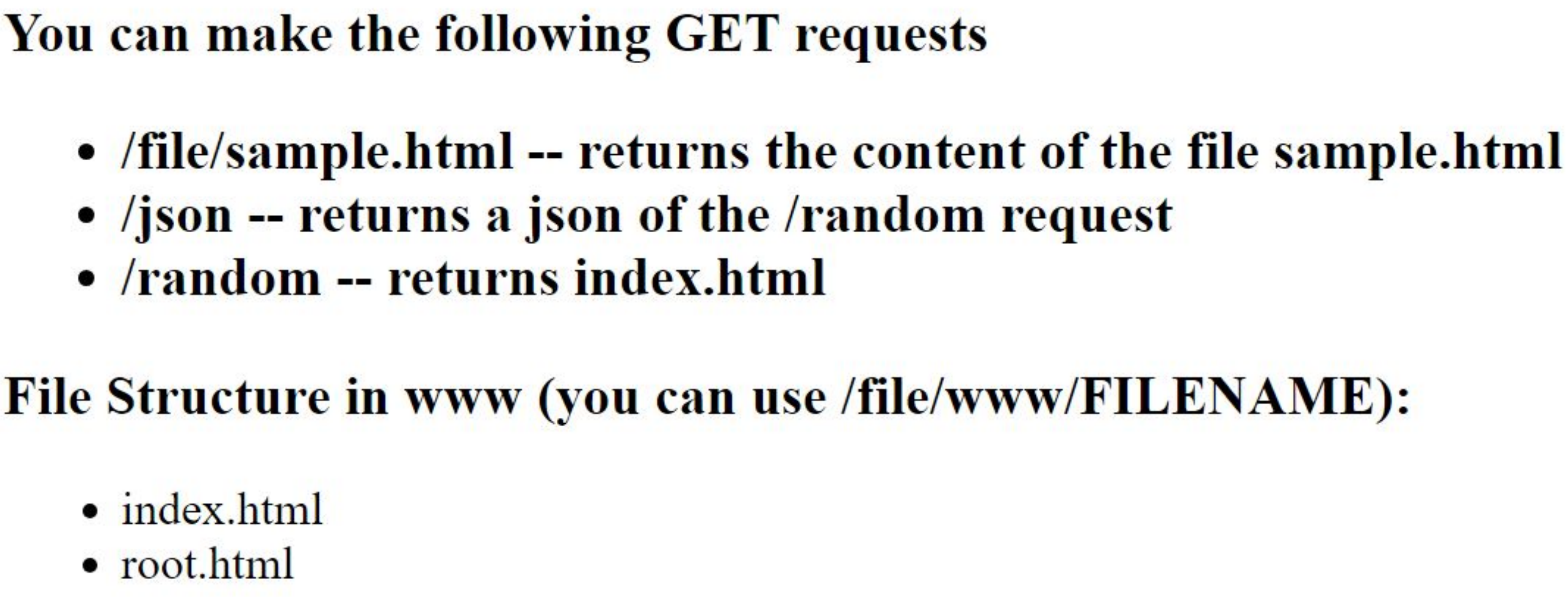
http://18.214.2.80:9000

STEP TWO:
Open in browser

SER 321

Assignment 2 WebServer

You will see this on a successful connection!



You can make the following GET requests

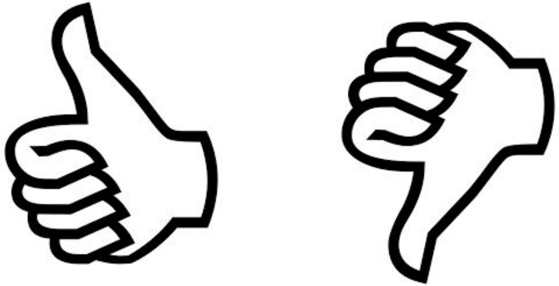
- /file/sample.html -- returns the content of the file sample.html
- /json -- returns a json of the /random request
- /random -- returns index.html

File Structure in www (you can use /file/www/FILENAME):

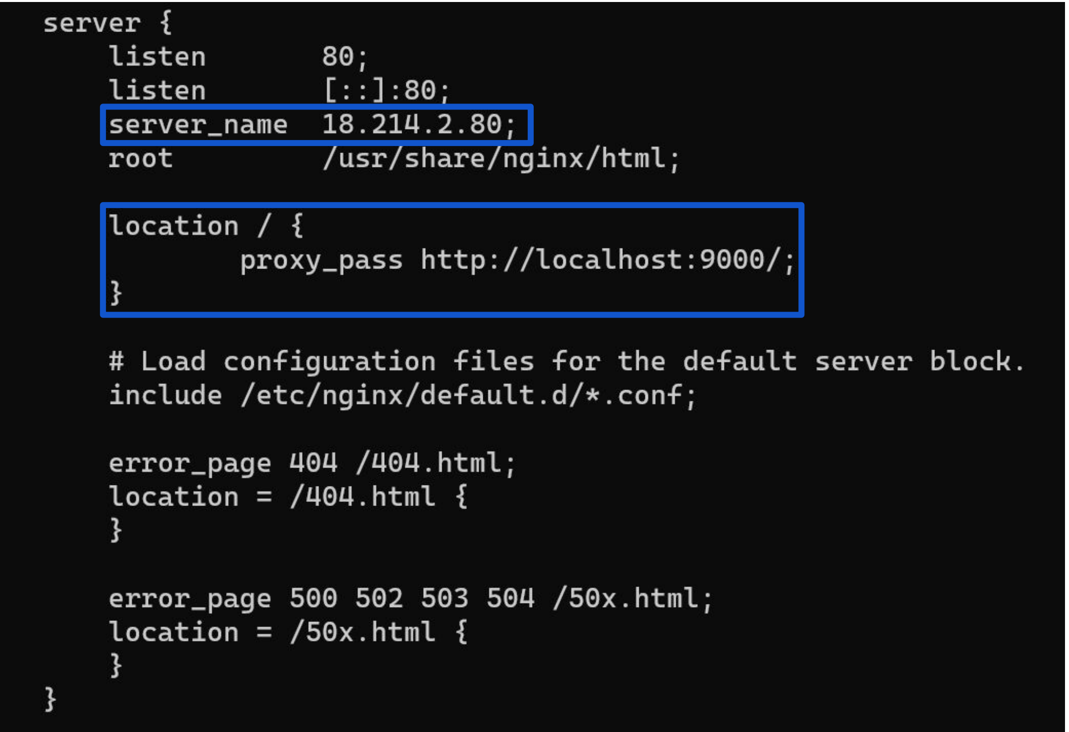
- index.html
- root.html

SER 321

Assignment 2 WebServer



What about the “real” WebServer in **Task 2.4**?



```
server {
    listen       80;
    listen       [::]:80;
    server_name  18.214.2.80;
    root         /usr/share/nginx/html;

    location / {
            proxy_pass http://localhost:9000/;
    }

    # Load configuration files for the default server block.
    include /etc/nginx/default.d/*.conf;

    error_page 404 /404.html;
    location = /404.html {
    }

    error_page 500 502 503 504 /50x.html;
    location = /50x.html {
    }
}
```

STEP ONE:
Setup and start nginx

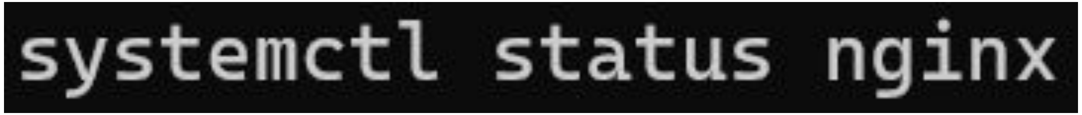
SER 321

Assignment 2 WebServer

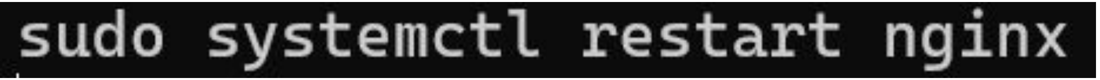
What about the “real” WebServer in **Task 2.4**?

STEP ONE:
Setup and start nginx

Do we remember how to work with nginx?



```
systemctl status nginx
```



```
sudo systemctl restart nginx
```

SER 321

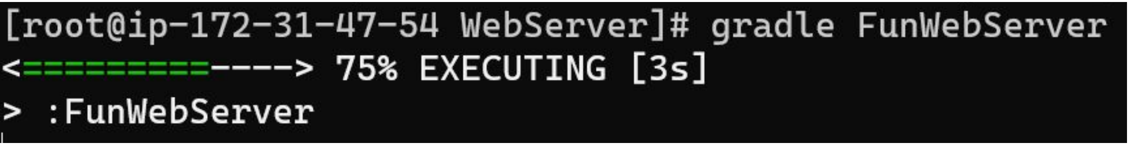
Assignment 2 WebServer

What about the “real” WebServer in **Task 2.4**?

STEP ONE:
Setup and start nginx

Do we remember how to work with nginx?

STEP TWO:
`gradle FunWebServer`

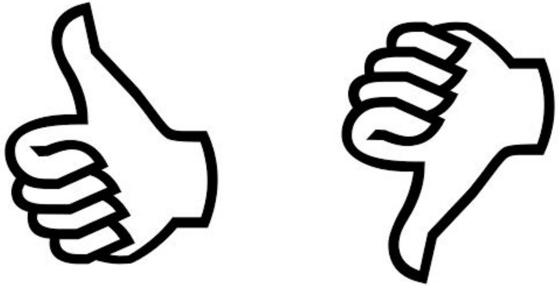


```
[root@ip-172-31-47-54 WebServer]# gradle FunWebServer
<=========----> 75% EXECUTING [3s]
> :FunWebServer
```

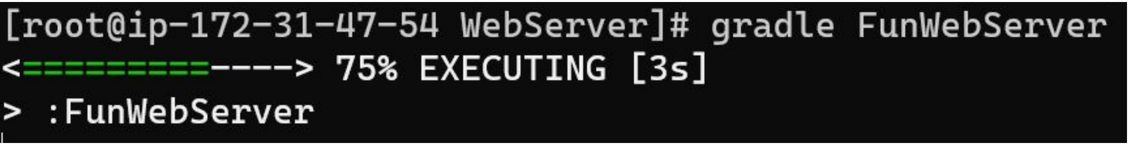
SER 321

Assignment 2 WebServer

What about the “real” WebServer in **Task 2.4**?



STEP ONE:
Setup and start nginx



```
[root@ip-172-31-47-54 WebServer]# gradle FunWebServer
<=========----> 75% EXECUTING [3s]
> :FunWebServer
```

Do we remember how to work with nginx?

STEP TWO:
`gradle FunWebServer`

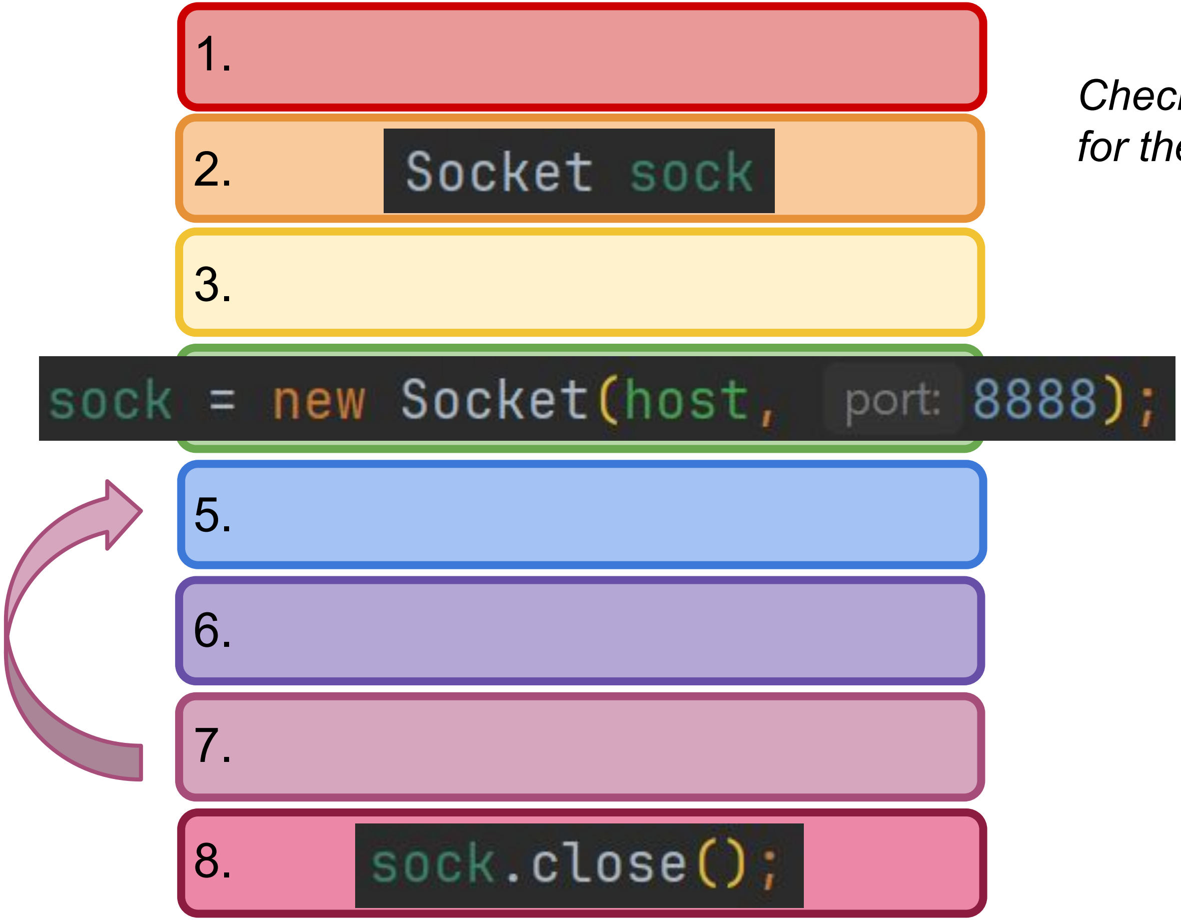
STEP THREE:
Open in browser

SER 321

Sockets!

Generic Steps for working with sockets:

Client



1.

2.

```
Socket sock
```

3.

```
sock = new Socket(host, port: 8888);
```

5.

6.

7.

8.

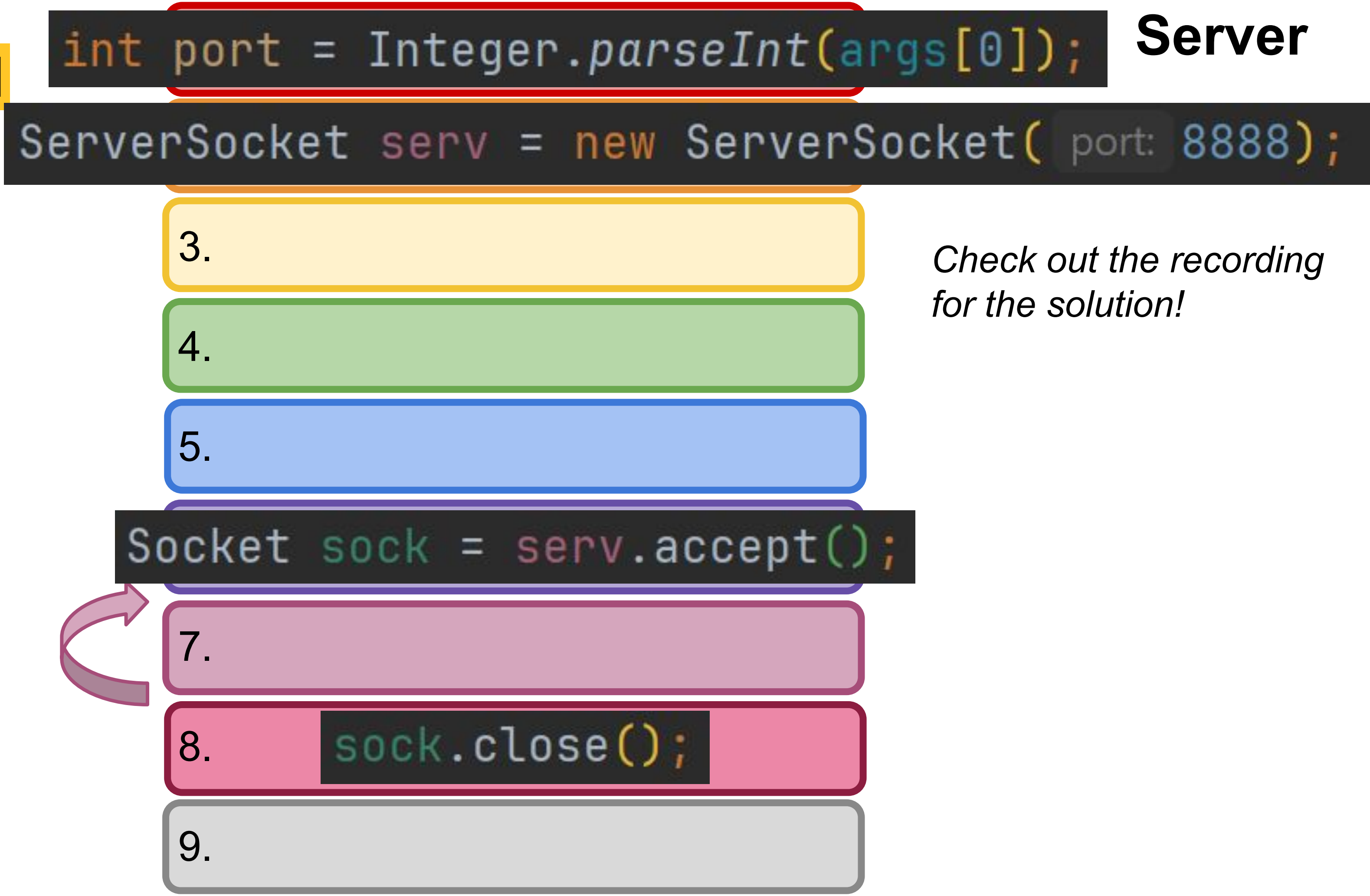
```
sock.close();
```

Check out the recording for the solution!

SER 321

Sockets!

Server



```
int port = Integer.parseInt(args[0]);
```

```
ServerSocket serv = new ServerSocket( port: 8888);
```

3.

4.

5.

```
Socket sock = serv.accept();
```

7.

8.

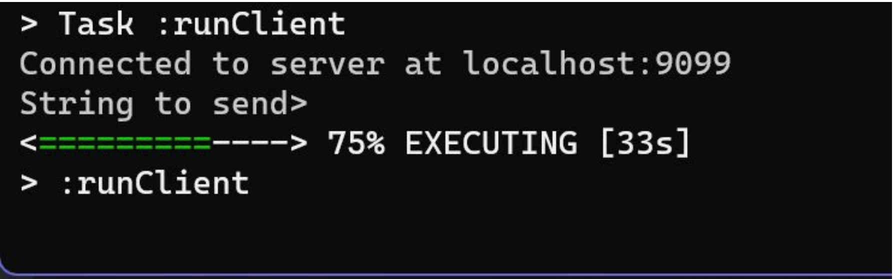
```
sock.close();
```

9.

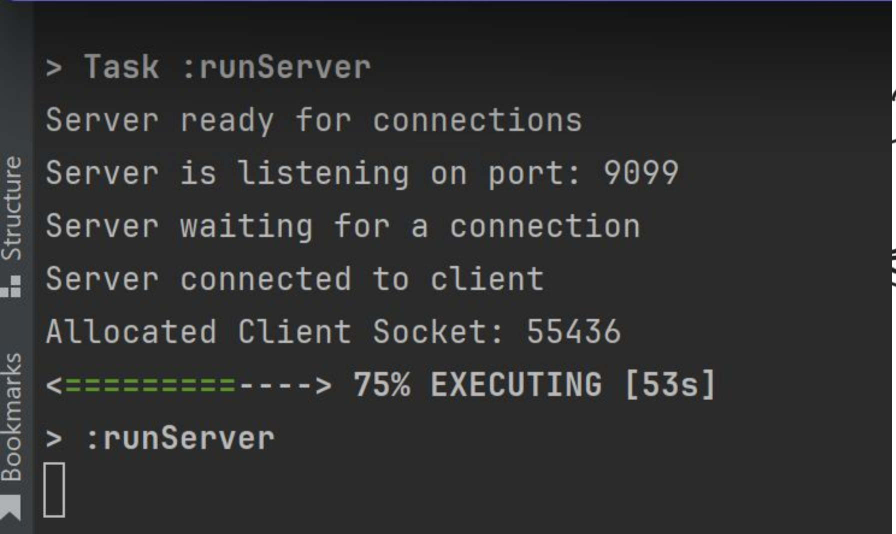
Check out the recording for the solution!

SER 321
Client Socket

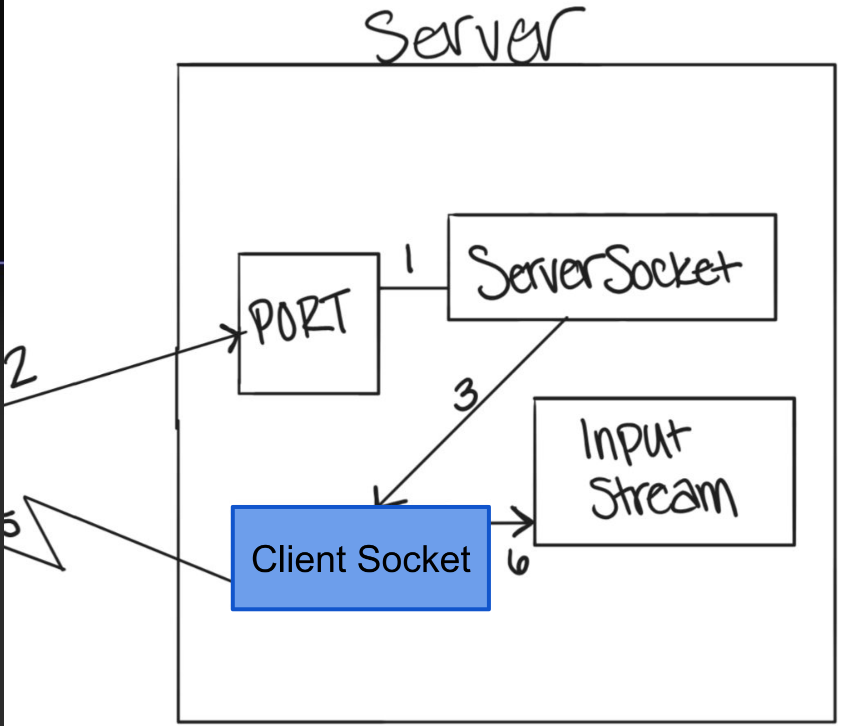
Remember that the OS will dynamically allocate a new port for the Client Socket!



```
> Task :runClient
Connected to server at localhost:9099
String to send>
<=========----> 75% EXECUTING [33s]
> :runClient
```



```
> Task :runServer
Server ready for connections
Server is listening on port: 9099
Server waiting for a connection
Server connected to client
Allocated Client Socket: 55436
<=========----> 75% EXECUTING [53s]
> :runServer
```

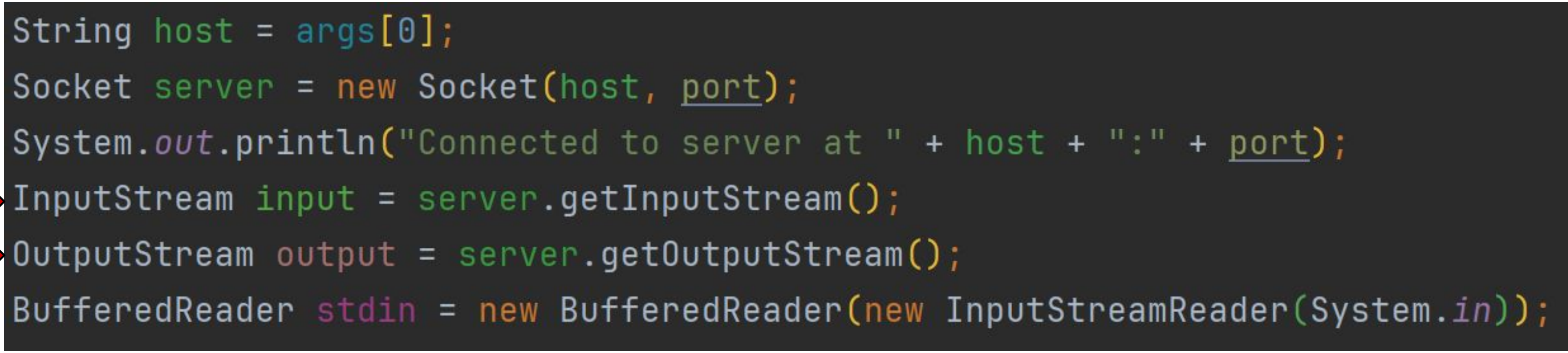


SER 321

Working with Sockets

So the connection is established, now what?

Need to get the Input and Output streams!

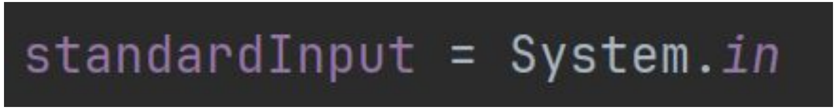


```
String host = args[0];
Socket server = new Socket(host, port);
System.out.println("Connected to server at " + host + ":" + port);
InputStream input = server.getInputStream();
OutputStream output = server.getOutputStream();
BufferedReader stdin = new BufferedReader(new InputStreamReader(System.in));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

Read **from** server

Write **to** server

Read from console



```
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

Keep the protocol handy at all times!

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

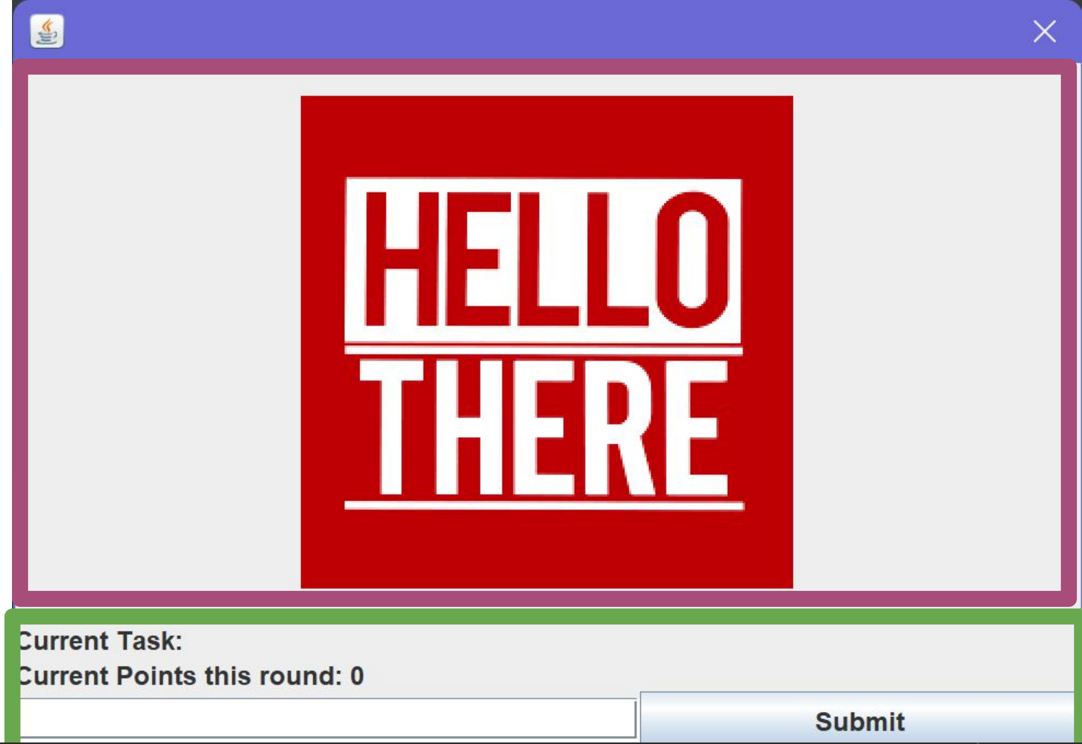
SER 321

Assignment 3 Part 2

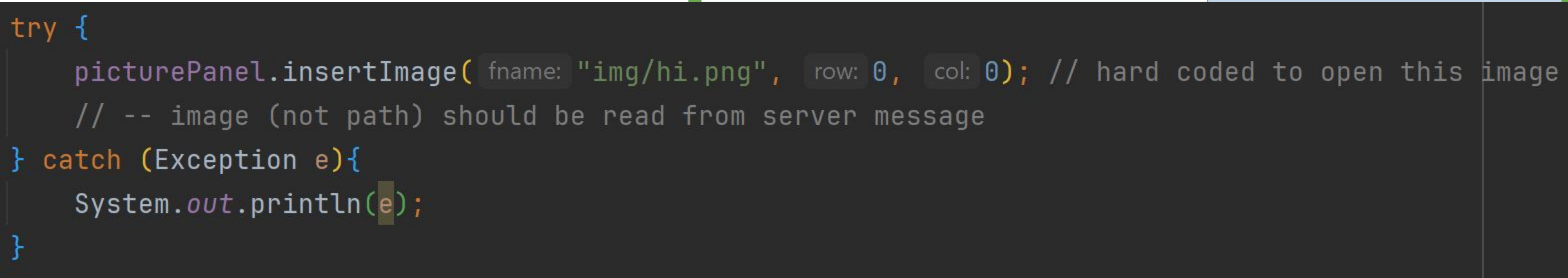
Let’s take a quick peek at the UI in Assignment 3 Part 2

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);
}
```

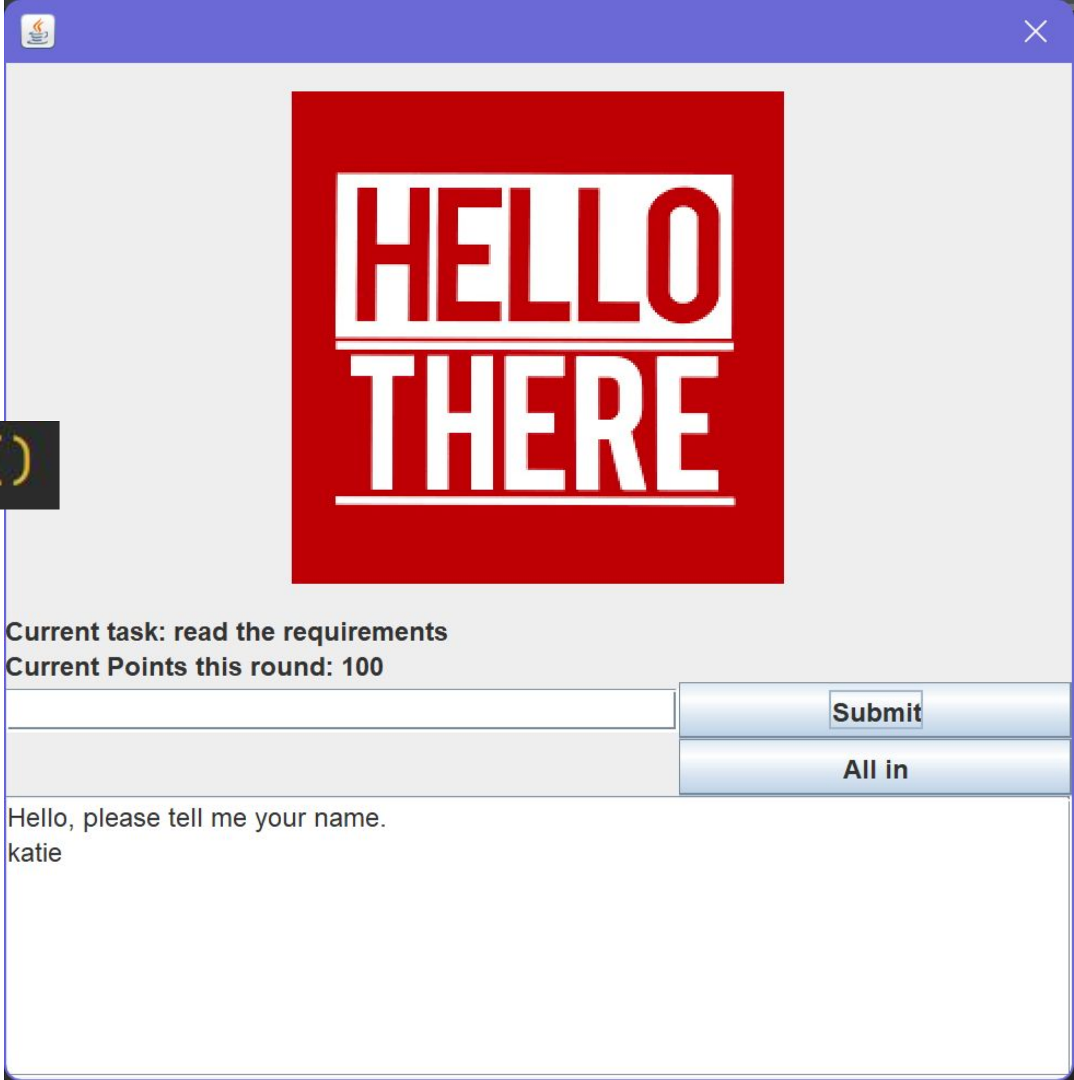
SER 321

Assignment 3 Part 2

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

```
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:

- Thursday, October 26th 2023 at 7:00 pm MST

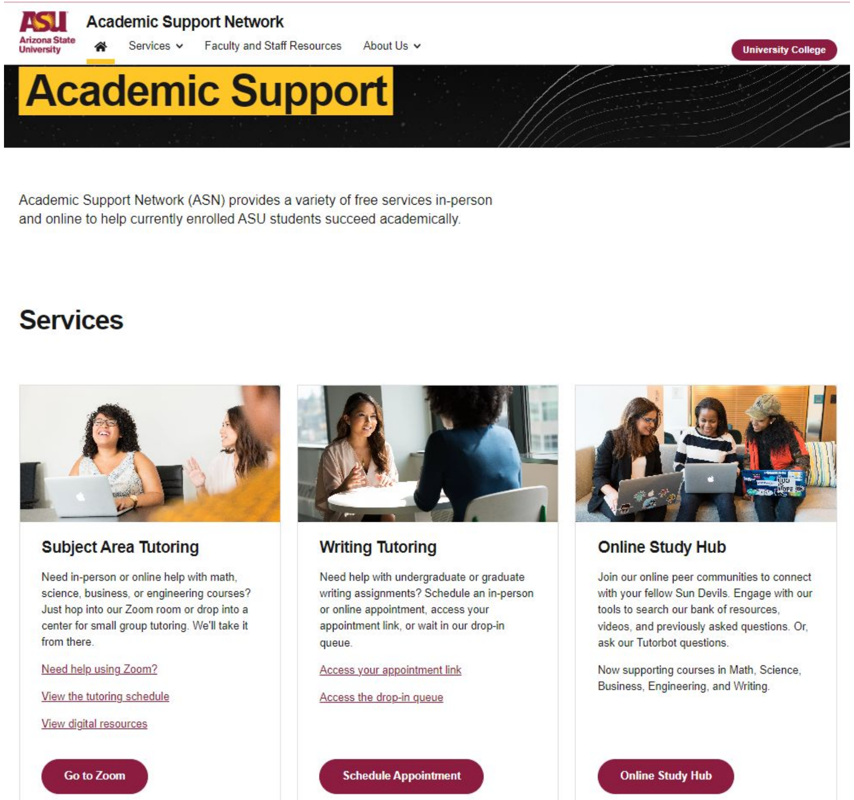
Review Sessions:

- 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

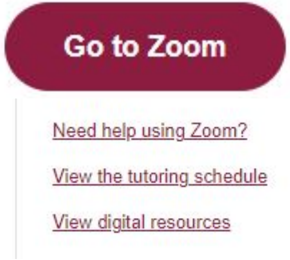
Online Study Hub

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

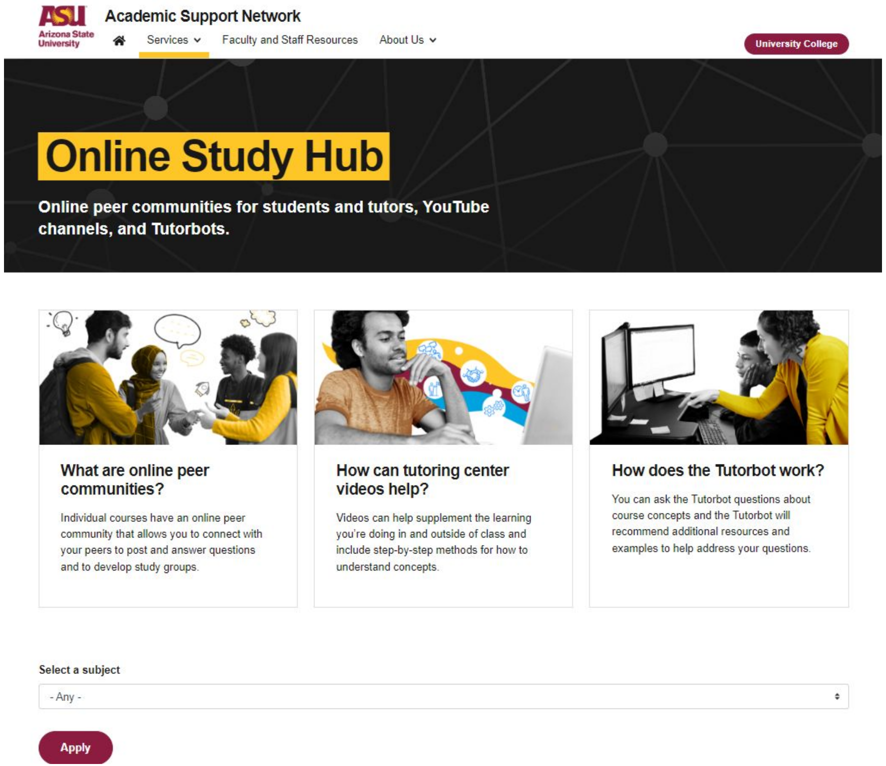
View digital resources

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.

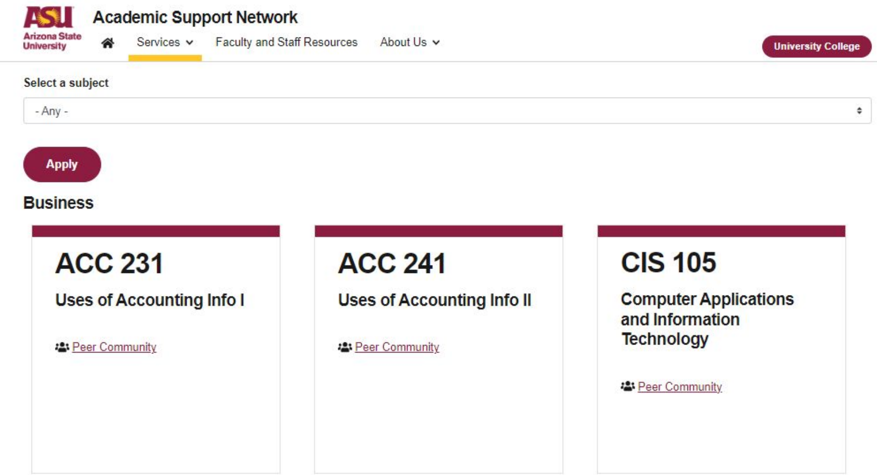
How can tutoring center videos help?

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	ACC 241	CIS 105
Uses of Accounting Info I	Uses of Accounting Info II	Computer Applications and Information Technology
Peer Community	Peer Community	Peer Community

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

Additional Resources

CoureRepo